IJNS
International Journal of Network Security

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

Vol. 22, No. 5 (September 2020)

INTERNATIONAL JOURNAL OF NETWORK SECURITY

PUBLISHING OFFICE

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

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

PUBLISHER: Candy C. H. Lin

Volume: 22, No: 5 (September 1, 2020)

International Journal of Network Security

Research on Malware Detection and Classification Based on Artificial Intelligence

Li-Chin Huang[1], Chun-Hsien Chang[2], and Min-Shiang Hwang[3,4]
(Corresponding author: Min-Shiang Hwang)

Department of Information Management, Executive Yuan, Taipei 10058, Taiwan[1]
Department of Management Information Systems, National Chung Hsing University, Taiwan[2]
Department of Computer Science & Information Engineering, Asia University, Taichung, Taiwan[3]
500, Lioufeng Rd., Wufeng, Taichung 41354, Taichung, Taiwan, R.O.C.
Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan[4]
(Email: mshwang@asia.edu.tw)
(Received Apr. 13, 2020; Revised and Accepted July 8, 2020; First Online July 9, 2020)

Abstract

Malware remains one of the major threats to network security. As the types of network devices increase, in addition to attacking computers, the amount of malware that affects mobile phones and the Internet of Things devices has also significantly increased. Malicious software can alter the regular operation of the victim's machine, damage user files, steal private information from the user, steal user permissions, and perform unauthorized activities on the device. For users, in addition to the inconvenience caused by using the device, it also poses a threat to property and information. Therefore, in the face of malware threats, if it can accurately and quickly detect its presence and deal with it, it can help reduce the impact of malware. To improve the accuracy and efficiency of malware detection, this article will use deep learning technology in the field of artificial intelligence to study and implement high-precision classification models to improve the effectiveness of malware detection. We will use convolutional neural networks and long and short-term memory as the primary training model. When using convolutional neural networks for training, we use malware visualization techniques. By converting malware features into images for input, and adjusting the input features and input methods, models with higher classification accuracy will be found; in long-term and short-term memory models, appropriate features and preprocessing methods are used to find Model with high classification accuracy. Finally, the accuracy of small sample training is optimized by generating features for network output samples. In the above training, all of us want to use malware as a sample that affects different devices. In this article, we propose three research topics: 1). When importing images, high-precision models are used to study malware. 2). When importing non-images, a high-precision model will be used to study the malware. 3). By using this model, the generated adversarial network is optimized for small sample malware detection.

Keywords: Android Malware; Deep Learning; Machine Learning; Malware; Ransomware Detection

1 Introduction

Today, malware is still one of the significant cybersecurity threats [2, 10]. According to Symantec's 2018 Cybersecurity Threat Report [23]: In terms of traditional malware that affects computers, the total number of malware variants discovered in 2017 was as high as 669 million, an increase of 87.7% from the amount found in 2016; In terms of malware that affects mobile devices such as cell phones, the number of new variants has increased from 17,000 to 27,000, an increase of about 55%. Due to the advancement and popularization of IoT technology, the number of malware that affects IoT devices has increased significantly in recent years. The report mentions that malware variants affecting IoT devices have grown by about 600% in recent years. It is the main threat to the currently accessible IoT device environment.

Common malware currently includes Kotver Trojans, worms, ransomware, spyware, and Coinminer. Most malware infection methods are that attackers use system and software security vulnerabilities to implant malware into the victim's device or trick the victim into downloading a file containing malware, and then implant the malware into the victim's computer. Different malware can have different effects on infected devices. Usually, it may affect the regular operation of the device, damage user files, steal user's private information, steal user rights, and perform unauthorized activities on the device. Most ransomware encrypts users' files and requires a ransom to unlock files, which poses a severe threat to users' data and property. Besides, due to the prevalence of virtual

currencies such as Bitcoin, malware that secretly uses the computing power of the victim's computer for mining operations is also one of the widespread malware in recent years.

Traditionally, malware detection methods can be roughly divided into static analysis and dynamic analysis. Static analysis can be realized by byte sequence analysis, control flow chart and operation code frequency distribution to achieve malware identification; Currently, dynamic analysis is based on the use of appropriate monitoring tools for API call monitoring and program behavior monitoring in controlled environments (such as sandboxes or virtual machines). Due to the development of artificial intelligence technology, the identification of malware has excellent potential. At present, many studies aiming at this aspect aim to improve accuracy and efficiency. Currently, most artificial intelligence technologies are used to improve the detection of static analysis.

In order to achieve the purpose of detecting and distinguishing malware, many methods have been proposed in different studies. In 2011, Nataraj *et al.* proposed a method for classifying malware after visualization is proposed [17]; In 2011, Santos *et al.* A method for detecting malware using the opcode sequence of malware is proposed. In 2013, based on the 2011 proposed method and the method of tracking the behavior of malware after execution, combined static analysis and dynamic analysis to detect malware [20, 21]; In 2014, Zolotukhin *et al.* proposed to use the n-gram method to find the essential features in the opcode sequence, and then use support vector machines for training to detect malware [26]. In previous studies, only a small amount of opcode extraction can reduce the performance overhead. However, when this method is used for a small number of training sets, accuracy may be severely affected. Therefore, Zhang *et al.* proposed a method to convert Opcode sequences to images in 2017 was introduced to improve this problem [25]; In the same year, Kwon *et al.* proposed to use API call patterns to identify the type of malware [15]; In 2018, Ni *et al.* proposed to use the SimHash method for hashing and converting it into an image as a sample for training [18]; Kim *et al.* proposed a method for detecting Android malware using multi-feature input as training is proposed [14]; For malware affecting the Internet of Things, Hamed *et al.* uses LSTM as training to detect malware that affects IoT devices in 2018 [8]; Sajad *et al.* digitized ransomware running records, and then use CNN and LSTM to train a classifier to achieve ransomware recognition [9].

The following evaluation criteria are commonly used to evaluate the effectiveness of this research topic:

1) Accuracy, recall rate, precision, and F-measure: These indicators are used to analyze the results of machine learning. Table 1 shows the calculation method for each parameter.

   Accuracy: The proportion of correct samples classified for the classifier to the total number of samples:

   $$Accuracy = \frac{TP + TN}{FN + TP + FP + TN}$$

   This result indicates the classifier's ability to distinguish the entire sample set. However, in some cases, this indicator will fail. For example, if there are 10,000 A samples and 100 B samples in the data set, and the classifier will judge all samples as A, the accuracy is still 99%.

   Recall: The proportion of positive samples correctly classified by the classifier among all positive samples:

   $$Recall = \frac{TP}{FN + TP}$$

   Precision: The proportion of all classified samples classified as classified samples by the classifier:

   $$Precision = \frac{TP}{TP + FP}$$

   The result indicates the actual accuracy of predicting positive samples.

   F-measure: If the Recall is as important as Precision in the model, the F-measure can be used as an indicator. The calculation also considers two indicators: Precision and Recall:

   $$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

2) Operating cost: Compare the time spent training and distinguishing samples with the required operating resources.

Table 1: The evaluation criteria

	Actual True	Actual False
Prediction True	True Positive (TP)	False Positive (FP)
Prediction False	False Negative (FN)	True Negative (TN)

2 Research on Malware Input by Image

2.1 Motivations

Many types of machine learning models can identify malware. At present, the performance is the best, and the most popular are various neural network models. Among them, a convolutional neural network (CNN) can be said to be one of the representatives. CNN is very suitable

for the recognition of image processing. When identifying malware, if the training samples are input as images, CNN can achieve a good classification effect.

Nataraj *et al.* proposed the use of image processing technology to visualize and classify malware in 2011 [17]. This method converts binary malware files into grayscale images for classification. Kancherla *et al.* proposed in 2013 to convert malware executable files into grayscale images called byteplot. And use a Support Vector Machine (SVM) as a training tool [11]. The method proposed by Zhang *et al.* in 2017 decompiled the binary executable file and converted the Opcode sequence into an image. Then identify whether the file is malware [25]. Ni *et al.* proposed to use the SimHash method in 2018 to hash decompiled malware for sequence similarity comparison. Then use CNN for training to achieve the effect of distinguishing different types of malware [18].

From the above research, we can see that after visualizing the malware and then analyzing it, a lot of research has been done in this area. But the methods are different, so we think it is still possible to find higher accuracy and efficiency in visualizing and analyzing malware. In addition to the above research, we also need to focus on converting different data from malware and then performing machine learning, different preprocessing of image input, and the impact on training. In the first research topic, we propose to use several different feature input methods to convert to images, and then try to use different preprocessing methods to process the images. Finally, the best feature selection and preprocessing methods in the experiment are obtained.

2.2 Related Works

The malware is based on images as training samples. After the malware is visualized, it can be processed for related research to identify the malware. Nataraj *et al.* proposed a method for visualizing malware, and performing automatic classification in 2011 is proposed [17]. Figure 1 is a schematic of the study. Since it was observed that the malware of the same family would have similarities in the texture and layout of the image, the study first converted binary malware files into 8-bit vectors. Then convert the 8-bit vector into a grayscale image. Finally, the k-nearest neighbor of Euclidean distance is used for classification. The 9458 sample classifications have 98% classification accuracy among its 25 categories.

The method proposed by Zhang et al. in 2017 decompiled the binary executable file into an Opcode sequence. After converting it into an image, a convolutional neural network was used to compare the target image with the image generated by known malware. Then determine whether the file is malware [25]. The flow chart of its method is shown in Figure 2.

First, after decompressing the unknown file, find out its opcode sequence and frequency of occurrence. Next, use "Information Gain" to find out which function to use in training. After the selected function is converted to an image, training will be conducted to achieve the effect of identifying malware. In this study, the data set used included ten types of malware, with a total of 9168 samples. The classification accuracy of the research results is about 93.7% 96.7%.

In 2018, Ni *et al.* proposed an MCSC (Malware Classification using SimHash and CNN) method [18]. Use the SimHash method to hash the decompiled malware to achieve the effect that similar functions can hash similar sequences. After hashing, the results are converted into grayscale images, which are then trained using CNN to classify the malware. Figure 3 shows the research structure.

This method first decompiles the malware file into Opcode and then executes SimHash. The calculation method of SimHash is shown in Figure 4.

The results obtained by SimHash will be converted into grayscale images and used as samples for CNN training. The method proposed in this study maintains a stable classification accuracy rate for relatively few malware categories in the sample, with an average accuracy rate of 98.86%.

2.3 Malware Detection Based on the High-precision Model for Image Input

This research topic, malware implemented with the high-precision model during image input, focuses on models that can obtain the highest accuracy and performance when using malware as training input. The research architecture is shown in Figure 5.

This research topic will use different methods to deal with the steps of converting samples into images. Evaluate the impact of selecting the appropriate function or directly converting to an image on accuracy. Then, after converting to an image, explore whether different image processing methods can improve accuracy. Finally, adjust the training model to obtain a classifier that can be accurately classified.

This research topic will collect malware samples that affect different devices—for example, traditional computers used for research, mobile phones, Internet of Things devices, etc. Then, when mirroring the malware, we will use different methods to deal with malware. One is to extract features from malware and then convert the features into images. The extraction is mainly based on opcodes; the other is to image the malware directly. After processing the image, the image is used as a training sample.

Convolutional neural network (CNN) is the most widely used deep learning technique in image processing [1]. We will input the samples obtained in the previous step to CNN for training. CNN is roughly composed of a convolutional layer, pooling layer, and fully connected layer, as shown in Figure 6.

C1 and C2 in Figure 6 are convolutional layers. The convolution layer consists of convolution units. The input malware image and function detector (filter) are con-

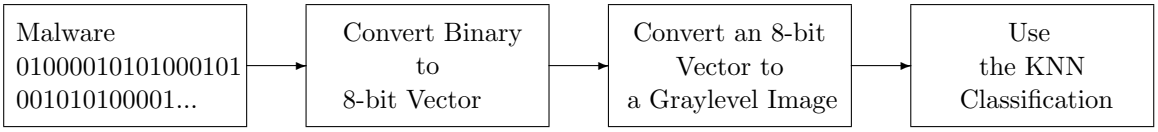


Figure 1: Method for visualizing and automatically classifying malware [17]

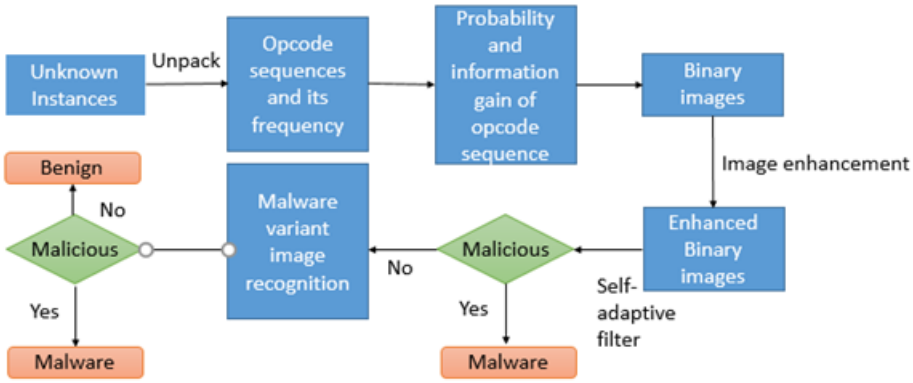


Figure 2: Flow chart of Zhang et al.'s method [25]

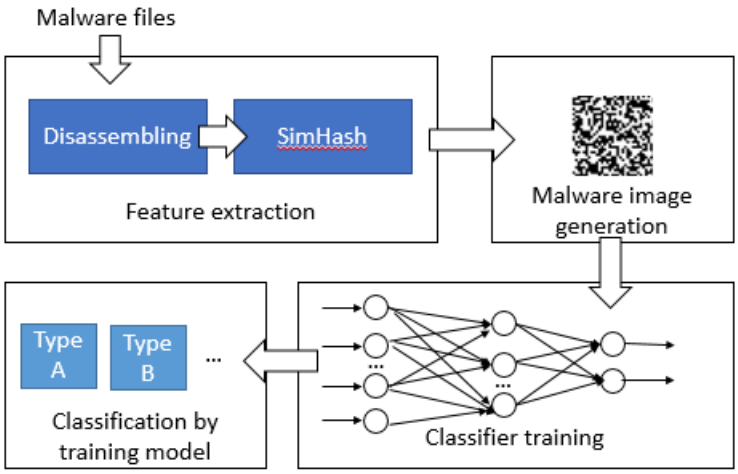


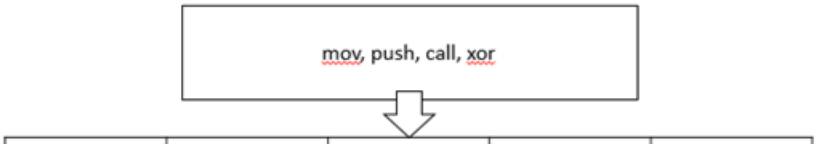
Figure 3: MCSC architecture

volved. Adding the appropriate excitation function (such as ReLU), we get the resulting output. The convolution operation can extract input features and repeatedly remove high-level, sophisticated features from low-level features such as lines in a multi-layer network.

The pooling layer will process the output of the convolutional layer. The pooling layer can be seen as a subsampling process. Taking the maximum pool as an example, if the maximum pool is 2×2, the output will be the maximum value in each 2×2 block. The data size, after processing by the pooling layer, will become smaller. Therefore, the number of parameters will also be reduced, which helps reduce the phenomenon of overfitting.

Figure 7 is an example of convolution operation and maximum pooling. The yellow box in the figure is the convolution operation of the 3×3 area and the feature detector; The red block is the largest pool in the 2×2 area.

Finally, the fully connected layer flattens the previous output and connects it to the neural network, and obtains the output after passing through the neural network. In this research topic, we first use the traditional CNN model. The training affected malware samples from different devices. After finding a better image input and processing method, adjust the CNN model to obtain better accuracy and performance.



Keyword no.	Opcode	Weight	Hashcode				Weight Vector			
W_1	mov	1	1	0	0	0	1	-1	-1	-1
W_2	push	1	1	1	0	0	1	1	-1	-1
W_3	call	1	1	1	0	1	1	1	-1	1
W_4	xor	1	0	1	1	0	-1	1	-1	1
SimHash Vector							3	2	-4	0
SimHash Code							1	1	0	0

Figure 4: Example of SimHash method

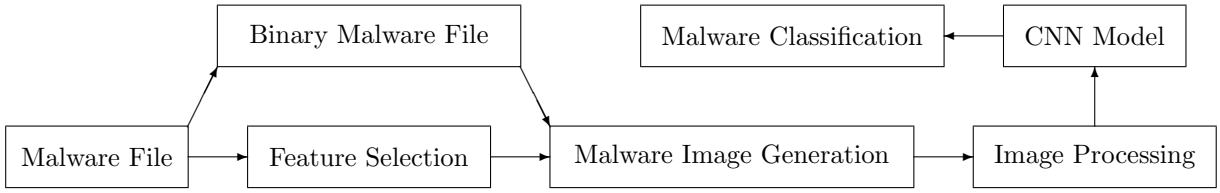


Figure 5: A research framework for the malware detection based on the high-precision model for image input

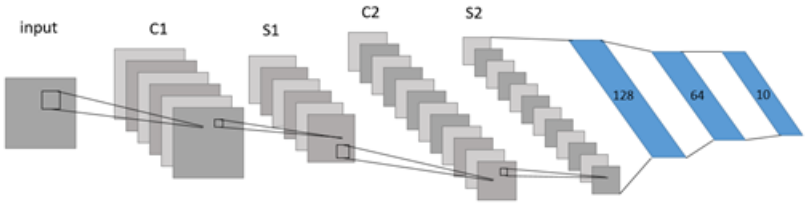


Figure 6: Example of CNN architecture

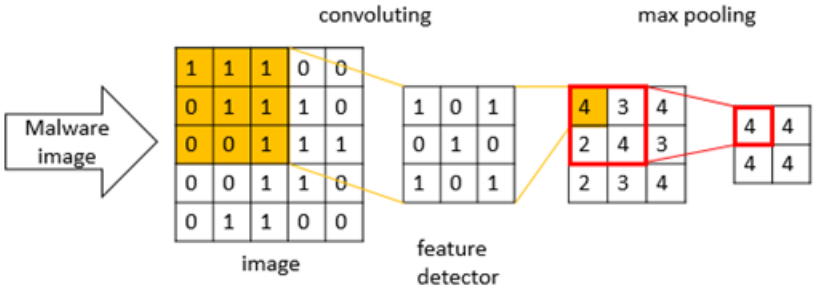


Figure 7: Convolution operation and maximum pooling

There are currently many sample sets open to the outside world. However, new malware samples that have recently appeared may be missing. There are two leading solutions: One is to cooperate with the malware collection platform to obtain fresh samples. The second is to establish a platform and encourage users to upload malware samples. Several malware data sets are currently open to the public and are expected to be used in research:

1) Microsoft Malware Classification Challenge [16];
2) The ultimate gaming malware research benchmark [4];

3) Android malware data set [24];

4) AndroZoo [3];

5) CTU-13 [22].

3 Research on Malware Input by Non-image

3.1 Motivations

In addition to using image input for machine learning of malware samples, directly using opcodes or character string sequences as training inputs is also one of the main methods. However, just as the input image is used as the training sample, the selection of the input features of the sample, the preprocessing of the data, and the adjustment of the model all affect the accuracy and efficiency of the classification.

Kim *et al.* proposed a multi-mode deep learning method using multiple types of features to detect Android malware. The method uses information obtained from decompiled apk files to extract various kinds of features—for example, strings, permissions, and calls API, and so on. Then use the Multimodal Neural Network (MNN) for training. Finally, it is used to determine whether the input file is benign software or malware [14]. Hamed et al. used RNN (Recurrent Neural Network) and LSTM (Long Short Term Memory) in 2018 to take Opcode of IoT malware as input to train a model that can be judged to be benign or malware [8]. Sajad et al. used to record and digitize the actions of the ransomware when it was running, and then used LSTM and CNN to train separately to classify benignly and ransomware and their types [9].

The subject of this research is to test different types of samples (traditional computer malware, mobile malware, etc.) based on their selectable characteristics. Find the combination of the best accuracy and efficiency, and compare it with the best performing images obtained in the previous stage.

3.2 Related Works

The subject of this study uses non-image methods as a sample input. This section will introduce research on different types of malware (such as Android, IoT, and ransomware).

In 2018, Kim *et al.* proposed an Android malware detection framework that uses multi-mode deep learning methods with multiple input features to detect Android malware [14]. Its architecture is shown in Figure 8. First, decompile the apk file to extract various types of functions. This study is divided into seven categories: String functions, method opcode functions, method API functions, shared library function opcode functions, permission functions, component functions, and environment functions. After generating the feature matrix from the features, a multimodal neural network will be used for training. Finally, input files can be distinguished as benign software or malware. The study used two sets of samples, namely 1,075 malware and 19,417 benign software, and 1,209 malware and 1,300 benign software. The model under study achieved 98% accuracy and 0.99 F measurement in the first set of samples. In the second set of samples, 99% accuracy and 0.99 F measurement were obtained.

Due to the rapid increase in malware targeting IoT devices, Hamed *et al.* used LSTM to detect IoT malware [8] in 2018. Figure 9 is its research architecture. After decompressing and decompiling the sample, the opcode is taken out, and then the opcode is selected as the feature to generate the feature vector. Then use RNN and LSTM for training. The study used 281 malware samples and 270 benign software samples. And use three different LSTM configurations for training (Layers 1 to 3). Finally, 100 samples not used in training are used to evaluate the performance of the model. Finally, compare with random forest, support vector machine, KNN, and other classification models. The two-layer LSTM configuration has the highest accuracy. The accuracy rate is 98.18%.

Sajad *et al.* proposed the DRTHIS (Deep ransomware threat hunting and intelligence system) method in 2018 [9]. The research architecture is shown in Figure 10. It is roughly divided into three parts: data conversion, threat detection (whether detection is ransomware), and what kind of ransomware is detected. This method records and digitizes the motion information within 10 seconds after the file is executed. The digitized data will be merged with the label into a training data set. Then through the two models of CNN and LSTM, the binary classifier and the ransomware classifier are trained. Finally, a system capable of judging benign and malicious software and distinguishing the types of ransomware is obtained. In this study, the LSTM model obtained relatively good results. In the experiment, three different types of ransomware samples (Locky, Cerber, TeslaCrypt) were used in the experiment. Each type of ransomware used 220 and 219 benign samples for training. The result of the F-measure is 0.96, and the true positive rate is 97.2%. Also, the study used other types of ransomware not used for training. 99% of CryptWall samples, 75% of TorrentLocker samples, and 92% of Sage samples were correctly classified.

3.3 Malware Detection Based on the High-precision Model for Non-image Input

For this research topic, we focus on research when non-images are used as a sample input. The research architecture is shown in Figure 11.

The input samples are adjusted according to different feature selection methods and combinations. At the same time, there are many variations of LSTM. For example, GRU (Gated Cycling Unit), *etc.* The accuracy and performance of the image are also worth discussing.

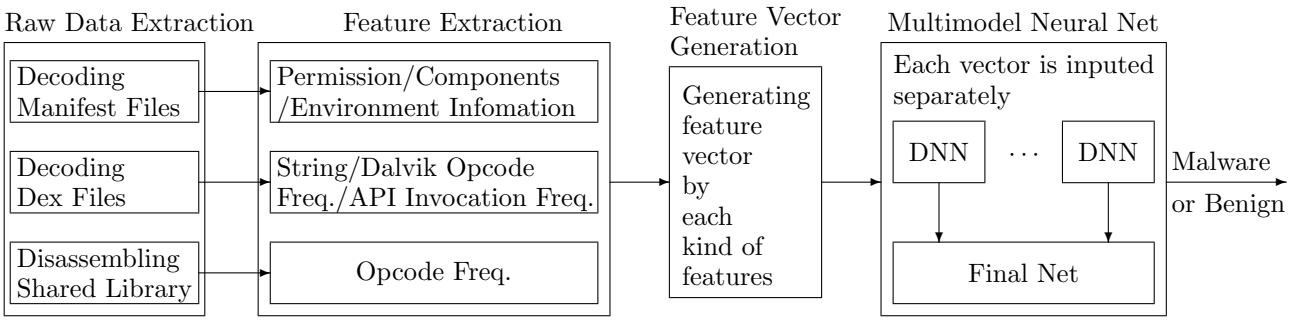


Figure 8: The framework of Kim *et al.* [14]

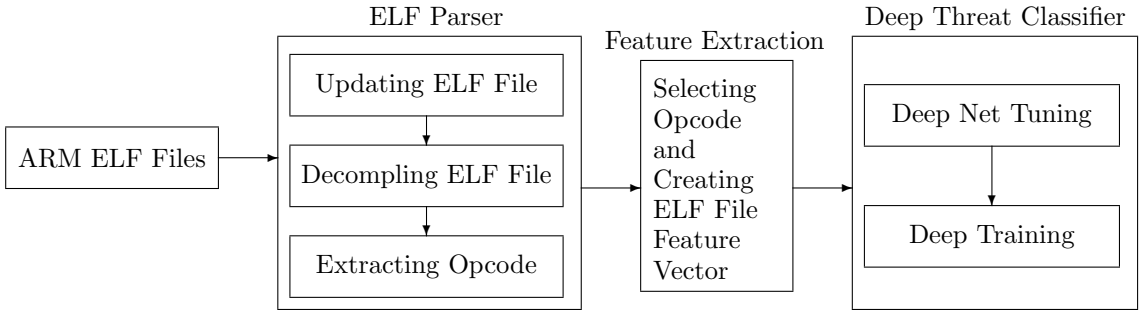


Figure 9: Using LSTM to detect IoT malware [8]

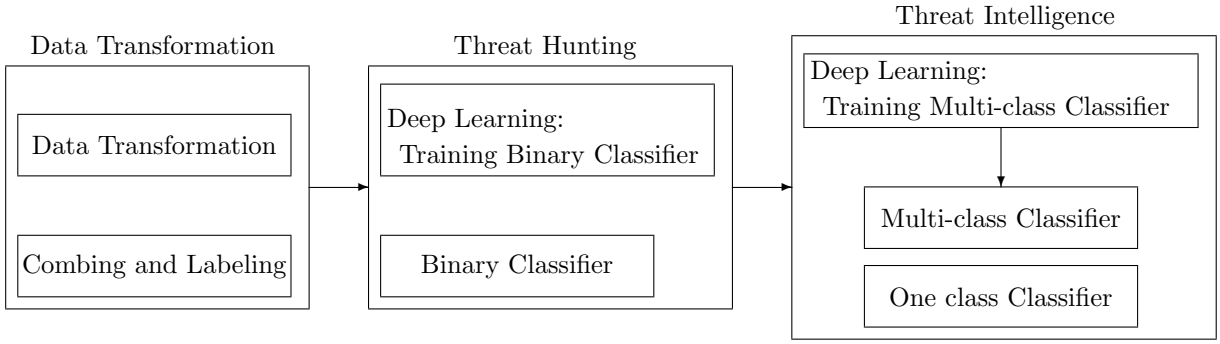


Figure 10: DRTHIS architecture [9]

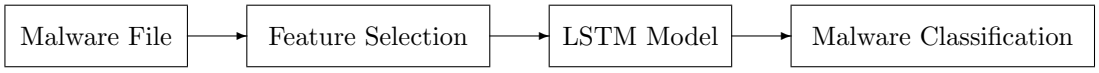


Figure 11: Research structure for the malware implemented with the high-precision model for non-image input

First, we select the features of malware samples based on their category characteristics. For example, malware that affects mobile phones will have functions such as permission requirements, which can also be used as features. At this stage, we use LSTM and its deformation as the training model.

When dealing with non-image input, especially the input of sequence data, LSTM will be a very suitable model. Since the traditional RNN will have a connection phenomenon between the next node and the previous node (The vanishing gradient problem for RNNs), the weight of the self-loop can be changed through the input gate, output gate, and forget gate:

- The function of the input gate is to determine whether to add the current input to the long-term memory.
- The function of the output gate is to determine whether to add the current input to the output.
- The function of the forget gate is to determine whether the incoming information of the upper layer should be kept in memory or forgotten.

Therefore, when the model parameters are fixed, the problem of gradient disappearance or expansion can be avoided. Figure 12 shows the architecture of the LSTM model.

There are many variations of LSTM. One of them is GRU (Gated Circulation Unit). GRU merges the input gate and forgets the gate in LSTM into one update gate. Compared with LSTM, GRU has the advantage of simplifying the calculation. If the prediction performance used is similar to LSTM, it is possible to improve performance.

4 Research on Malware and Generative Adversarial Networks

4.1 Motivations

In the current research, the results of machine learning are related to the number of samples. However, in the face of new malware threats, you must limit the number of samples in a short period. The Generative Adversarial Network (GAN) that have emerged in recent years has considerable potential in this regard [6, 7, 12]. GAN is composed of the Generative model and the Discriminative model. Generative model random samples of the network as input, and the output should be as close as possible to the real samples in the training and deception network; The discriminative model identifies the real samples of the network or generates the output of the system and distinguishes the non-real samples as much as possible. These two networks face each other and adjust the parameters, which ultimately enables the generating network to generate samples, thus making the discriminating network unable to identify authenticity.

Most GANs are used to process image samples. When this method is applied to the classification of malware, we convert the samples into image input methods and study the accuracy of classifying a small number of samples using GANs. In addition, there are currently several variants of GANs. Improving the accuracy or efficiency of malware classification is also one of the research directions.

4.2 Related Works

In 2017, Kim et al. proposed the use of generative adversarial networks for malware classification [13]. In 2018, the research continued and discussed the protection measures for zero-day attacks [12]. Figure 13 is its research architecture.

Traditional malware detection mechanisms usually rely on existing functions. The defense effect against zero-day attacks is limited. Therefore, the method proposed in this study first visualizes the malware and then uses the generative adversarial network (GAN) for training. Pseudo samples, the GAN generates pseudo samples and adjusts the parameters through continuous confrontation with the discriminant network. Finally, the generated fake samples are similar to the actual samples, but not the same. It can mimic a variant of the virus.

Part of the detector uses an autoencoder to learn the features of the malware and feed it back to the generation network to train the generator stably. The classification accuracy rate is 95.74%. At the same time, in the experiments of this study, noise-added sample images were used to simulate virus variants. The SSIM (Structural Similarity Index) of the sample after adding noise and the original sample is $0.6 \sim 0.69$. The accuracy rate is between 98.16% and 98.99%, which indicates that it also has good detection ability for the new variant virus.

4.3 Malware Detection Based on GAN for Generating Small-sample Malware

For this research topic, we used the best performing image input and processing methods in the first research topic to process the samples to be used at this stage. Then use GAN for training to improve the training effect of a small number of samples. It can be expected that when new types of malware appear, they can be effectively detected even with a small number of samples.

The research architecture at this stage is shown in Figure 14. In this research stage, we used a malware data set with a small number of samples and extracted a certain amount of samples from the previous samples as samples at this stage. Then use GAN as a model for this training phase.

The concept of GAN was proposed by Goodfello *et al.* in 2014 [7]. Figure 15 shows the underlying architecture of the GAN.

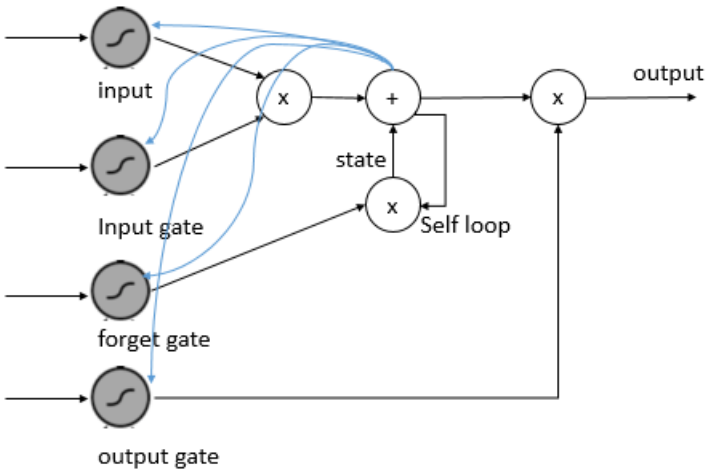


Figure 12: LSTM model

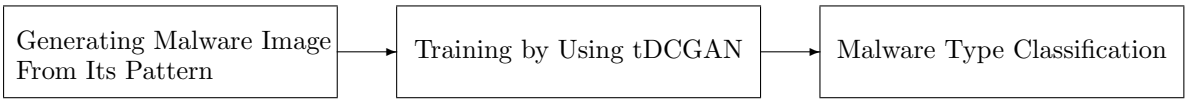


Figure 13: The research framework of Kim *et al.* [12]

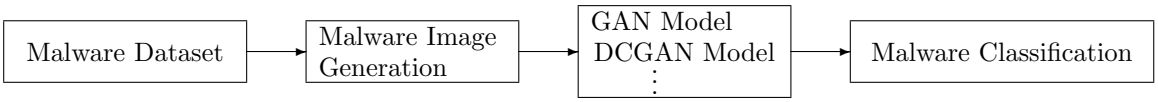


Figure 14: Research structure of malware detection based on GAN for generating small-sample malware

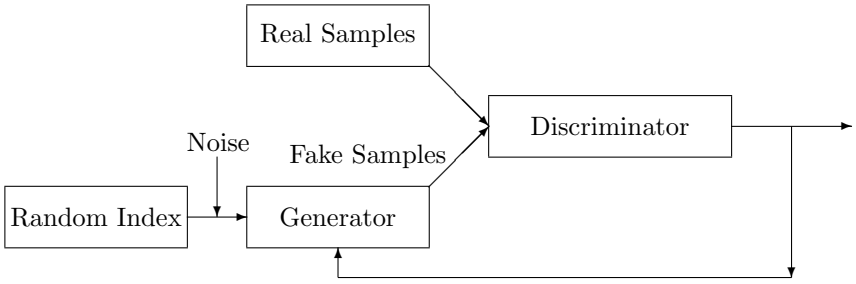


Figure 15: Generating an adversarial network model

GAN consists of a generating network and a discriminative network. The generating network generates random samples of the network as input and produce an output that can deceive and distinguish the network; The discriminative network inputs real samples or generates network output and distinguishes non-real samples as much as possible. The two networks struggle with each other and are adjusting parameters. The final sample generated by the network is almost real. Using this technology, you can amplify the number of samples analyzed by malware and a small amount of samples, and combine the results of the previous two research topics to optimize the accu-

racy of malware analysis for a small number of samples. GAN is not suitable for learning discrete data, such as text. Therefore, in this part, the image will still be used as the sample input form.

In addition, GAN has conducted many studies to propose its deformation. For example, DCGAN proposed in [19], WGAN proposed in [5], and so on.

5 Conclusions

In this article, we have proposed three research topics: 1) Research on malware input by image: Malware detection based on the high-precision model for image input; 2) Research on malware input by non-image: Malware detection based on the high-precision model for non-image input; 3) Research on malware and generative adversarial networks: Malware detection based on GAN for generating small-sample malware.

This article collected and used malware samples that affected different devices for training. It contains malware that affects computers, malware, and ransomware that affects mobile phones. Malware that affects different methods has various features. For example, due to the apk file structure of mobile phones, the malware that affects mobile phones has many different characteristics compared to the malware of cellular phones; Another example, the permission functions and component functions mentioned in the related research. The goal of this article is to propose a high-precision model for different types of samples.

Acknowledgments

This research was partially supported by the Ministry of Science and Technology, Taiwan (ROC), under contract no.: MOST 108-2410-H-468-023 and MOST 108-2622-8-468-001-TM1.

References

[1] D. S. Abdul Minaam and E. Amer, "Survey on machine learning techniques: Concepts and algorithms," *International Journal of Electronics and Information Engineering*, vol. 10, no. 1, pp. 34–44, 2019.

[2] A. A. Al-khatib, W. A. Hammood, "Mobile malware and defending systems: Comparison study," *International Journal of Electronics and Information Engineering*, vol. 6, no. 2, pp. 116–123, 2017.

[3] K. Allix, T. F. Bissyandé, J. Klein, and Y. Le Traon, "AndroZoo: Collecting millions of android apps for the research community," in *IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR'16)*, 2016.

[4] H. S. Anderson, P. Roth, "EMBER: An open dataset for training static PE malware machine learning models," *arXiv preprint*, arXiv: 1804.04637, 2018.

[5] M. Arjovsky, S. Chintala, L. Bottou, "Wasserstein GAN," *arXiv preprint*, arXiv: 1701.07875, 2017.

[6] J. Bruner, A. Deshpande, *Generative Adversarial Networks for Beginners*, July 8, 2020. (`https://github.com/jonbruner/generative-adversarial-networks/blob/master/gan-notebook.ipynb`)

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirze, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14)*, vol. 2, pp. 2672–2680, 2014.

[8] H. HaddadPajouh, A. Dehghantanha, R. Khayami, K. K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," *Future Generation Computer Systems*, vol. 85, pp. 88-96, 2018.

[9] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, R. Khayami, K. K. R. Choo, D. E. Newton, "DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer," *Future Generation Computer Systems*, vol. 90, pp. 94-104, 2019.

[10] S. Islam, H. Ali, A. Habib, N. Nobi, M. Alam, and D. Hossain, "Threat minimization by design and deployment of secured networking model," *International Journal of Electronics and Information Engineering*, vol. 8, no. 2, pp. 135–144, 2018.

[11] K. Kancherla, S. Mukkamala, "Image visualization based malware detection," in *IEEE Symposium on Computational Intelligence in Cyber Security (CICS'13)*, pp. 40-44, 2013.

[12] J. Y. Kim, S. J. Bu, S. B. Cho, "Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders," *Information Sciences*, vol. 460–461, pp. 83-102, 2018.

[13] J. Y. Kim, S. J. Bu, S. B. Cho, "Malware detection using deep transferred generative adversarial networks," in *International Conference on Neural Information Processing*, pp. 556-564, 2017.

[14] T. Kim, B. Kang, M. Rho, S. Sezerm, E. G. Im, "A multimodal deep learning method for Android malware detection using various features," *IEEE Transcations on Information Foresics and Security*, vol. 14, no. 3, pp. 773-788, 2019.

[15] I. Kwon, E. G. Im, "Extracting the representative API call patterns of malware families using recurrent neural network," in *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, ACM, pp. 202-207, 2017.

[16] Microsoft, *Microsoft Malware Classification Challenge (BIG 2015)*, July 8, 2020. (`https://www.kaggle.com/c/malware-classification`)

[17] L. Nataraj, S. Karthikeyan, G. Jacob, B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proceedings of the Eighth International Symposium on Visualization for Cyber Security*, pp. 311–320, 2011.

[18] S. Ni, Q. Qian, R. Zhang, "Malware identification using visualization images and deep learning," *Computers & Security*, vol. 77, pp. 871-885, 2018.
[19] A. Radford, L. Metz, S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *Under review as a conference paper at ICLR 2016*, 2016. (`https://arxiv.org/pdf/1511.06434.pdf`)
[20] I. Santos, F. Brezo, X. Ugarte-Pedrero, P. G. Bringas, "Opcode sequences as representation of executables for data mining based malware variant detection," *Information Sciences*, vol. 231, no. 9, pp. 64-82, 2011.
[21] I. Santos, J. Devesa, F. Brezo, J. Nieves, "OPEM: A static-dynamic approach for machine learning based malware detection," in *Proceedings of International Conference (CISIS'12)*, pp. 271-280, 2013.
[22] Stratosphereips Lab., *The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background Traffic*, July 8, 2020. (`https://www.stratosphereips.org/datasets-ctu13/`)
[23] Symantec, *Internet Security Threat Report*, vol.23, 2018. (`https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf`)
[24] F. Wei, Y. Li, S. Roy, X. Ou, W. Zhou, "Deep ground truth analysis of current android malware," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, pp. 252-276, 2017.
[25] . J. Zhang, Z. Qin, H. Yin, L. Ou, Y. Hu, "IRMD: Malware variant detection using opcode image recognition," in *IEEE International Conference on Parallel and Distributed Systems (ICPADS'17)*, pp. 1175–1180, 2017.
[26] M. Zolotukhin, T. Hamalainen, "Detection of zero-day malware based on the analysis of opcode sequences," in *Proceedings of The 11th Annual IEEE CCNC Security, Privacy and Content Protection*, pp. 386-391, 2014.

Biography

Li-Chin Huang received the B.S. in computer science from Providence University, Taiwan, in 1993 and M.S. in information management from Chaoyang University of Technology (CYUT), Taichung, Taiwan, in 2001; and the Ph.D. degree in computer and information science from National Chung Hsing University (NCHU), Taiwan in 2001. Her current research interests include information security, cryptography, medical image, data hiding, network, security, big data, and mobile communications.

Chun-Hsien Chang received B.S. in Department of Mechanical Engineering from National Cheng Kung University, Taiwan in 2017; M.S. in Department of Management Information Systems, National Chung Hsing University, Taiwan, in 2019. His current research interests include artificial intelligence and information security.

Min-Shiang Hwang received M.S. in industrial engineering from National Tsing Hua University, Taiwan in 1988, and a Ph.D. degree in computer and information science from National Chiao Tung University, Taiwan, in 1995. He was a professor and Chairman of the Department of Management Information Systems, NCHU, during 2003-2009. He was also a visiting professor at the University of California (UC), Riverside, and UC. Davis (USA) during 2009-2010. He was a distinguished professor of the Department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Excellent Research Award of National Science Council (Taiwan). Dr. Hwang was a dean of the College of Computer Science, Asia University (AU), Taichung, Taiwan. He is currently a chair professor with the Department of Computer Science and Information Engineering, AU. His current research interests include information security, electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.

A BLS Signature Scheme from Multilinear Maps

Fei Tang[1] and Dong Huang[2,3]
(Corresponding author: Fei Tang)
School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications[1]
Chongqing, China
Chongqing University of Science and Technology [2]
Chongqing Vocational and Technical University of Mechatronics[3]
(Email: tangfei@cqupt.edu.cn)
(Received Mar. 12, 2019; Revised and Accepted Dec. 10, 2019; First Online Apr. 6, 2020)

Abstract

The security of the BLS signature scheme is based on the random oracle model. Hence, there is a question that how to instantiate BLS scheme without random oracles. In this work, by using a powerful tool multilinear map, we answer this question. The main contributions of this work are as follows. First of all, we describe the BLS scheme in the setting of multilinear group and prove its security in the standard model. Then, we design a ring signature scheme based on the multilinear BLS scheme. In the proposed scheme, ring signatures consist of a single multilinear group element.

Keywords: BLS Signatures; Multilinear Map; Ring Signatures; Standard Model

1 Introduction

Digital signatures are one of the most fundamental and well studied cryptographic primitives. The research of digital signatures has two aspects: practicability and security. As for the aspect of practicability, we mainly consider the efficiencies of the signing and verification algorithms, the storage space of the state information, and the properties that the scheme can provide, such as aggregate signatures [17], ring signatures [23], blind signatures [16, 19, 20] and so on. As for the aspect of the security, we mainly focus on the assumptions that the scheme based on, such as one-way function, and whether in the random oracle model or standard model.

At ASIACRYPT 2001, Boneh, Lynn, and Shacham [6] designed a short signature scheme based on bilinear group, the so called BLS scheme. The BLS scheme has shown to be very useful to construct other cryptographic primitives, such as threshold signature scheme [7], blind signature scheme [7], signcryption scheme [10], key-generation algorithm of identity-based encryption (IBE) scheme [4]. The main reason of the BLS scheme is so useful is that it has a relatively simple structure. The security of the BLS scheme is rely on the random oracle model. However, it is well known that random oracle is an ideal model. After BLS scheme, there has several works that presented some signature schemes which secure in the standard model, such as [3]. However, all of these schemes do not preserve the BLS scheme's simple structure. This leads us to a question: *Can we instantiate the BLS signature scheme without random oracles?* In this work, by using a powerful tool multilinear map, we answer this question.

The notion of multilinear maps was introduced (but without concrete instantiation) by Boneh and Silverberg [8]. Until 2013, Garg, Gentry, and Helevi [12] gave the first approximate candidate. Then there has many multilinear map schemes been proposed or analyzed, *e.g.*, [9, 15, 21] *et al.*

There are several relevant works answered the above question. Hohenberger *et al.* [18] instantiated the random oracle with an actual family of hash functions for the BLS scheme by using a more powerful tool indistinguishability obfuscation [13]. Freire *et al.* [11] took advantage of multilinear maps to realize programmable hash functions and construct IBE, BLS signature, and SOK non-interactive key exchange schemes. In addition, Hohenberger *et al.* [17] made use of the multilinear BLS scheme to construct an (identity-based) aggregate signature scheme which admits unrestricted aggregation. In [17], Hohenberger *et al.* followed in the Waters [25] framework and proved the adaptive security of the multilinear BLS signature scheme.[1] However, their proof is based on a *strong* new assumption, (n, k)-modified multilinear computational Diffie-Hellman exponent where n is a polynomial of the number of queries made by the adversary.

In this work, we also give an adaptive proof for the multilinear BLS scheme. However, our proof which takes advantage of the technique of admissible hash function [2] is based on a *weaker* assumption, multilinear computa-

[1]Their adaptive proof of the aggregate signature scheme implies this result. (Please refer to Appendix D.2 of [17] for details.)

tional Diffie-Hellman assumption [12]. In addition, we consider the applications of the multilinear BLS signature scheme. It can be served as the key-generation algorithm of the multilinear IBE scheme [11]. It also can be used to construct aggregate signature [17], threshold signature scheme [7] and so on. In this work, we take advantage of the structure of the multilinear BLS scheme to construct a ring signature scheme in the standard model. In a ring signature scheme [23], a signer can generate signatures on behalf of a group of users (*i.e.*, ring) if and only if he is a member of the ring. Then, any verifier can confirm that the message has been signed by one of the members in the ring, but he cannot know who is the real signer. Our ring signature scheme has an attractive feature that for n members of a ring the signatures consist of just a single group element.

2 Preliminaries

2.1 Notations

The following notations will be used in this paper. Let $\mathbb{Z}$ be the set of integers and $\mathbb{Z}_p$ be the ring modulo p. 1^λ denotes the string of λ ones for $\lambda \in \mathbb{N}$. $|x|$ denotes the length of the bit string x. $[k]$ is a shorthand for the set $\{1, 2, \ldots, k\}$. Finally, we write PPT for the probabilistic polynomial time.

2.2 Multilinear Maps

Let $(\mathbb{G}_1, \ldots, \mathbb{G}_k)$ be a sequence of groups each of large prime order p, and g_i be a generator of group $\mathbb{G}_i$, where we let $g = g_1$. There exists a set of bilinear maps $\{\mathbf{e}_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j} | i, j \geq 1 \wedge i + j \leq k\}$, which satisfy:

$$\mathbf{e}_{i,j}(g_i^a, g_j^b) = g_{i+j}^{ab} : \forall a, b \in \mathbb{Z}_p.$$

When the context is obvious, we omit the indexes i and j, *i.e.*, $\mathbf{e}(g_i^a, g_j^b) = g_{i+j}^{ab}$. It also will be convenient to abbreviate $\mathbf{e}(h_1, h_2, \ldots, h_j) = \mathbf{e}(h_1, \mathbf{e}(h_2, \ldots, \mathbf{e}(h_{j-1}, h_j)\ldots)) \in \mathbb{G}_i$ for $h_j \in \mathbb{G}_{i_j}$ and $i_1 + i_2 + \ldots + i_j \leq k$.

Let $\mathsf{MulGen}(1^\lambda, k)$ be a PPT multilinear group generator algorithm which takes as input a security parameter λ and an integer k, where k is the number of allowed pairing operations, then it outputs the multilinear parameters $\mathbb{MP} = (\mathbb{G}_1, \ldots, \mathbb{G}_k, p, g = g_1, g_2, \ldots, g_k, \mathbf{e}_{i,j})$ to satisfy the above properties.

In recent years, there has many multilinear maps been proposed, *e.g.*, [9,12,14,22]. However, some of them have been shown to be insecure, *e.g.*, [15, 21]. Fortunately, there still has several multilinear maps are beyond the existing cryptanalysis, *e.g.*, [1,14]. Therefore, we also can use this tool to design cryptographic schemes. For example, [26, 28, 29] take advantage of the multilinear maps to design different cryptographic schemes.

2.3 Complexity Assumption

We assume that the following assumption holds in the setting described above: Multilinear Computational Diffie-Hellman (MCDH) assumption.

Definition 1. For any PPT algorithm $\mathcal{B}$, any polynomial $p(\cdot)$, any integer k, and all sufficiently large $\lambda \in \mathbb{N}$,

$$\Pr\left[\begin{array}{ll} \mathbb{MP} \leftarrow \mathsf{MulGen}(1^\lambda, k); & \\ c_1, \ldots, c_k \xleftarrow{R} \mathbb{Z}_p; & : v = g_{k-1}^{\prod_{i\in[k]} c_i} \\ v \leftarrow \mathcal{B}(\mathbb{MP}, g^{c_1}, \ldots, g^{c_k}) & \end{array}\right] < \frac{1}{p(\lambda)}.$$

This assumption can be viewed as an adaptation of the Bilinear Computational Diffie-Hellman (BCDH) assumption [4] in the setting of multilinear groups.

3 Digital Signatures

3.1 Definitions

For ease of description, we define digital signature schemes with four algorithms: Setup, KeyGen, Sign, and Vrfy. Formally, given a security parameter λ, the PPT algorithm Setup, run by a trusted authority, generates public parameters $\mathbb{PP}$. The public parameters will be used in all of the following three algorithms, for simplicity, we omit this fact. The PPT algorithm KeyGen outputs a signing/verification key pair (SK, VK) for the signer. The PPT algorithm Sign takes as input a signing key SK and a message M, then outputs a signature σ. Finally, the deterministic algorithm Vrfy processes a purported signature σ with respect to a message M and verification key VK, accordingly, it outputs 1 to indicate a successful verification and 0 otherwise.

3.2 Existential Unforgeability

The security model for signature schemes is Existential Unforgeability against adaptive Chosen-Message Attacks (EU-CMA) which is defined by the following game.

1) **Setup:** Challenger runs Setup and KeyGen algorithms to generate the public parameters and challenge keys (SK^*, VK^*). Adversary $\mathcal{A}$ is given the public parameters and VK^*.

2) **Signing queries:** Adversary $\mathcal{A}$ is allowed to adaptively queries the signing oracle at most q times on messages $M_1, \ldots, M_q$. In its i-th query, it receives back a signature $\sigma_i \leftarrow \mathsf{Sign}(SK^*, M_i)$.

3) **Output:** Finally, adversary $\mathcal{A}$ outputs a tuple of (M^*, σ^*), where $M^* \neq M_i$ for all $i \in [q]$. $\mathcal{A}$ wins the game if $\mathsf{Vrfy}(VK^*, M^*, \sigma^*) = 1$.

We denote the success probability of a PPT adversary $\mathcal{A}$ (taken over the random choices of the challenger and adversary) to win the game as $\mathbf{Adv}_{\mathcal{A}}^{eu-cma}$.

Definition 2. We say that a signature scheme is EU-CMA secure, if for any PPT adversary $\mathcal{A}$, it cannot win the above game with non-negligible advantage.

Selective security. The model of selective security is a weaker notion of the model of EU-CMA. In such model, we require that the adversary gives its challenge message M^* before the setup phase, then it cannot make signing query for M^*.

Definition 3. We say that a signature scheme is selectively secure, if for any PPT adversary, it cannot win the selective game with non-negligible advantage.

4 Multilinear BLS Scheme

In this section, we describe the multilinear BLS signature scheme and its security.

4.1 Construction

We specify the message space $\mathcal{M} := \{0,1\}^\ell$, more generally, a collision resistant hash function can be used to hash messages to this size. The construction of the multilinear BLS signature scheme is as follows:

- $\mathsf{Setup}(1^\lambda, \ell)$: Trusted authority takes as input a security parameter λ and the length ℓ of messages to runs this algorithm to generate the public parameters. It first runs $\mathbb{MP} = (\mathbb{G}_1, \ldots, \mathbb{G}_k, p, g, \ldots, g_k, \mathbf{e}) \leftarrow \mathsf{MulGen}(1^\lambda, k = \ell+1)$. Next, it chooses 2ℓ random integers $(a_{1,0}, a_{1,1}), \ldots, (a_{\ell,0}, a_{\ell,1}) \in \mathbb{Z}_p^2$ and computes $A_{i,\beta} = g^{a_{i,\beta}} \in \mathbb{G}_1$, for $i \in [\ell]$ and $\beta \in \{0,1\}$. The public parameters $\mathbb{PP}$ contain the group descriptions $\mathbb{MP}$ and $(A_{1,0}, A_{1,1}), \ldots, (A_{\ell,0}, A_{\ell,1})$.

- $\mathsf{KeyGen}(\mathbb{PP})$: Each user chooses a random element $x \in \mathbb{Z}_p$ as his signing key SK. The corresponding verification key is $VK = g^x \in \mathbb{G}_1$.

- $\mathsf{Sign}(SK, M)$: Given a message, M, of length ℓ, let $m_1, \ldots, m_\ell$ be the bits of this message, the signer computes the signature as:

$$\sigma = \mathbf{e}(A_{1,m_1}, \ldots, A_{\ell,m_\ell})^x = (g_{k-1}^{\prod_{i=1}^{\ell} a_{i,m_i}})^x \in \mathbb{G}_{k-1}.^2$$

- $\mathsf{Vrfy}(VK, M, \sigma)$: Given a verification key VK and a purported signature σ on message M, verify the following equation:

$$\mathbf{e}(\sigma, g) \stackrel{?}{=} \mathbf{e}(A_{1,m_1}, \ldots, A_{\ell,m_\ell}, VK).$$

[2]In the bilinear BLS signature scheme [6], signer computes a signature as $\sigma = H(M)^x$, where $H(\cdot)$ is a collision-resistant hash function that will be treated as a random oracle in the proof. In the multilinear BLS signature scheme, $H(M)$ is defined as $\mathbf{e}(A_{1,m_1}, \ldots, A_{\ell,m_\ell})$ which can be computed from the public parameters of the (leveled) multilinear maps.

Correctness. To see the correctness, a signature σ on message M is $(g_{k-1}^{\prod_{i=1}^{\ell} a_{i,m_i}})^x$, and thus we have $\mathbf{e}(\sigma, g) = \mathbf{e}((g_{k-1}^{\prod_{i=1}^{\ell} a_{i,m_i}})^x, g) = \mathbf{e}((g_{k-1}^{\prod_{i=1}^{\ell} a_{i,m_i}}), g^x) = \mathbf{e}(A_{1,m_1}, \ldots, A_{\ell,m_\ell}, VK)$.

4.2 Security of Multilinear BLS Scheme in the Standard Model

We now prove the security of the multilinear BLS scheme in the standard model based on the MCDH assumption. Our proof needs an admissible hash function h which can be used to partition the message space to two subsets with probability $1/\theta(q)$ (where q is the upper bound of the adversary's queries) so that: the adversary's query messages M_i fall in one subset where we know a trapdoor that allows us to answer its queries, and the adversary's challenge message M^* falls in the other set where we do not know any trapdoor but hope to embed a challenge element. We show that we can leverage the structure of the multilinear BLS signature scheme to prove adaptive security. For simplicity of exposition, we assume that there is a polynomial $s(\lambda)$ which denotes the length of messages space to be signed. We use a function $h : \{0,1\}^{s(\lambda)} \rightarrow \{0,1\}^{\ell(\lambda)}$ maps the messages to ℓ bits, and an efficient randomized algorithm Sample that is θ-admissible. The following definition of admissible hash functions is from [17] which is a slight variant of the simplified definition in [11].

Definition 4. Let s, ℓ and θ be efficiently computable univariate polynomials. We say that a function $h : \{0,1\}^{s(\lambda)} \rightarrow \{0,1\}^{\ell(\lambda)}$, and an efficient randomized algorithm Sample, are θ-*admissible* if the following properties hold:

- For any $u \in \{0,1,\perp\}^\ell$, define $P_u : \{0,1\}^s \rightarrow \{0,1\}$ as follows: $P_u(X) = 0$ iff $\forall i : h(X)_i \neq u_i$, and otherwise (if $\exists i : h(X)_i = u_i$) we have $P_u(X) = 1$.

- We require that for any efficiently computable polynomial $q(\lambda)$, for all $X_1, \ldots, X_q, Z \in \{0,1\}^s$, where $Z \notin \{X_i\}$, we have $\Pr[P_u(X_1) = \ldots = P_u(X_q) = 1 \wedge P_u(Z) = 0] \geq 1/\theta(q)$, where the probability is taken only over $u \leftarrow \mathsf{Sample}(1^\lambda, q)$.

Theorem 1. For any efficiently computable polynomials s, ℓ, there exists an efficiently computable polynomial θ such that there exists θ-admissible function families mapping s bits to ℓ bits.

The construction is identical to the multilinear BLS scheme with the exception of the Setup algorithm creates the admissible hash functions. Then the signing and verification algorithms take as input $h(M)$ instead of M.

Theorem 2. If h is a θ-admissible function and the k-MCDH problem is hard in the multilinear groups, then the multilinear BLS signature scheme with an admissible hash is adaptively secure.

Proof. If there exists a PPT adversary $\mathcal{A}$ who can break the security of the multilinear BLS signature scheme with an admissible hash in the EU-CMA game with advantage ϵ for message length s, level k of the multilinear maps, and security parameter λ, then we can construct a PPT challenger $\mathcal{B}$ to break the k-MCDH assumption with probability $\epsilon' \geq \epsilon/\theta(q)$. The challenger $\mathcal{B}$ takes as input a k-MCDH instance $(g^{c_1}, \ldots, g^{c_k})$ together with the group descriptions $\mathbb{MP}$ to interactive with the adversary. The challenger's goal is to compute $g_{k-1}^{\prod_{i\in[k]} c_i}$.

We describe the proof as a sequence of hybrid games where the first hybrid corresponds to the original EU-CMA game. Then in the first hybrid step we do a "partitioning" of the space of the messages. After the first proof step, we prove that any PPT adversary's advantage must be close with negligible gap at most between each successive hybrid games. We finally show that any PPT adversary in the final game that succeeds with non-negligible advantage can be used to break the k-MCDH assumption.

- Game_0 is the original EU-CMA game.

  1) **Setup:** Challenger $\mathcal{B}$ runs $\mathsf{MulGen}(1^\lambda, k)$ to produce group parameter $\mathbb{MP}$. It then chooses a random exponent $x \in \mathbb{Z}_p$ for the secret key and sets the challenge verification key as $VK^* = g^x$. It also randomly chooses $(a_{1,0}, a_{1,1}), \ldots, (a_{\ell,0}, a_{\ell,1})$ from $\mathbb{Z}_p$ and computes $A_{i,\beta} = g^{a_{i,\beta}}$ for $i \in [\ell], \beta \in \{0,1\}$. Finally, it sets $\mathbb{PP} = (\mathbb{MP}, \{A_{i,\beta} | i \in [\ell], \beta \in \{0,1\}\})$ and gives $\mathbb{PP}$ and VK^* to the adversary $\mathcal{A}$.
  2) **Signing queries:** Adversary $\mathcal{A}$ adaptively queries the signing oracle at most q times on messages $M_1, \ldots, M_q$. In its i-th query, it receives back $\mathbf{e}(A_{1,m_{i_1}}, \ldots, A_{\ell,m_{i_\ell}})^x$ from challenger $\mathcal{B}$.
  3) **Output:** At some point $\mathcal{A}$ outputs a forgery σ^* with respect to the challenge key VK^* and message M^*, it wins the game if $\mathsf{Vrfy}(VK^*, M^*, \sigma^*) = 1$ and $M^* \neq M_i$ for $i \in [q]$.

- Game_1 is the same as Game_0 except that the challenger begins by sampling a string $u \in \{0,1,\perp\}^\ell$ by revoking $u \leftarrow \mathsf{Sample}(1^\lambda, q)$. At the end of the game, the adversary is only considered to be successful if both its output satisfies the winning conditions and for the challenge message M^* we have $P_u(M^*) = 0$ and for all messages M_i queried $P_u(M_i) = 1$.

- Game_2 is the same as Game_1 except that the following modification. The challenger sets the parameters $(A_{1,0}, A_{1,1}), \ldots, (A_{\ell,0}, A_{\ell,1})$ in the following way: for $i \in [\ell]$ and $\beta \in \{0,1\}$ it chooses random $b_{i,\beta} \in \mathbb{Z}_p$ and sets

$$A_{i,\beta} = \begin{cases} g^{b_{i,\beta}}, & \text{if } \beta = u_i \\ (g^{c_i})^{b_{i,\beta}}, & \text{if } \beta \neq u_i. \end{cases}$$

Lemma 1. Assume an adversary that makes at most a polynomial of signing queries $q = q(\lambda)$ in Game_0. If the advantage of an adversary in Game_0 is ϵ, then the advantage of the adversary in Game_1 will be at least $\epsilon/\theta(q)$. In particular, any PPT adversary with non-negligible advantage in Game_0 will also have non-negligible advantage in Game_1.

Proof. The lemma follows immediately from the property of function h satisfies the definition of a θ-admissibility, since the only independent choice of $u \leftarrow \mathsf{Sample}(1^\lambda, q)$ determines whether or not the game aborts. □

Lemma 2. The advantage of any PPT adversary in Game_2 is the same as its advantage in Game_1.

Proof. The two games are equivalent as all $A_{i,\beta} \in \mathbb{G}_1$ are still set to uniformly at random in both games. □

Lemma 3. If the k-MCDH assumption holds, then the advantage of any PPT adversary in Game_2 is negligible.

Proof. We prove this lemma by giving a reduction to the k-MCDH assumption. To do so, we construct an algorithm $\mathcal{B}$.

$\mathcal{B}$ takes as input a k-MCDH problem instance $(\mathbb{MP}, g^{c_1}, \ldots, g^{c_k})$. Next, $\mathcal{B}$ runs $u \leftarrow \mathsf{Sample}(1^\lambda, q)$. It sets the challenge key as $VK^* = g^{c_k}$. All these steps together simulate the Setup phase of the Game_2. Now, it plays the game with the adversary $\mathcal{A}$ by using public parameters $\mathbb{PP} = (\mathbb{MP}, \{A_{i,\beta} | i \in [\ell], \beta \in \{0,1\}\})$ and challenge key VK^*.

The adversary $\mathcal{A}$ will then adaptively make at most q signing queries each for message M_i. If $P_u(M_i) = 0$, $\mathcal{B}$ aborts and quits. Otherwise, $P_u(M_i) = 1$ and there exists an γ we have $h(M_i)_\gamma = u_\gamma$. Thus, $\mathcal{B}$ can compute the signature as $\sigma = \mathbf{e}(A_{1,h(M_i)_1}, \ldots, A_{\gamma-1,h(M_i)_{\gamma-1}}, A_{\gamma+1,h(M_i)_{\gamma+1}}, \ldots, A_{\ell,h(M_i)_\ell}, VK^*)^{b_{\gamma,h(M_i)_\gamma}}$ by knowing the exponent $b_{\gamma,h(M_i)_\gamma}$ of the parameter $A_{\gamma,h(M_i)_\gamma}$.

Finally, the adversary $\mathcal{A}$ outputs an attempted forgery σ^* with respect to the challenge verification VK^* on some message M^*. $\mathcal{B}$ first checks the signature verification $\mathsf{Vrfy}(VK^*, M^*, \sigma^*)$ and aborts if it returns 0. Next, it checks if $P_u(M^*) = 1$ and aborts if that is the case. Otherwise, $P_u(M^*) = 0$ and for all i we have $h(M^*)_i \neq u_i$. This means that the hash of M^* will be $g_{k-1}^{\prod_{i\in[k-1]} c_i}$ raised to some known product of $b_{i,\beta}$ values. The signature therefore contains $g_{k-1}^{\prod_{i\in[k]} c_i}$ raised to some known product of $b_{i,\beta}$ values. This value can be recovered by taking the proper root of the signature, *i.e.*, $(\sigma^*)^{1/\prod_{i\in[\ell]} b_{i,h(M^*)_i}} = g_{k-1}^{\prod_{i\in[k]} c_i}$, and thus if σ^* is a successful forgery, then this root of the signature is a solution to the challenge instance of the k-MCDH problem.

By construction of the algorithm $\mathcal{B}$, the probability of $\mathcal{B}$ succeeds is exactly the advantage that the adversary $\mathcal{A}$ succeeds in Game_2. Whenever $\mathcal{B}$ aborted, the adversary by the rules of Game_2 was not considered to be successful

since its queries or forgery violated the partition. The lemma follows. □

These three lemmas together yield the main theorem that the multilinear BLS signature scheme with an admissible hash is adaptively secure. □

5 Ring Signatures from Multilinear BLS Scheme

In this section, we show the applications of the multilinear BLS signatures. It can be served as the key-generation algorithm of the multilinear IBE scheme [11], it also can be used to construct aggregate signature [17]. In addition, based on Boldyreva's [7] work, we can easily obtain a threshold signature, a multi-signature, and a blind signature scheme, respectively, based on the multilinear BLS scheme in the standard model. Here, we take advantage of the multilinear BLS scheme to construct a ring signature scheme in the standard model. The resultant scheme has an attractive feature that for n members of a ring our signatures consist of just a single group element.

5.1 Definition of Ring Signatures

For convenience, we define an algorithm Setup, run by trusted authority, to generate public parameters and build the system of the ring signature scheme. The public parameters will be used in all of the following three algorithms. In addition, we refer to an ordered set $R = \{VK_1, \ldots, VK_n\}$ of verification keys as a ring, and let $R[i] = VK_i$. We will also freely use set notation, *e.g.*, $VK \in R$ if there exists an index i such that $R[i] = VK$.

Definition 5. A ring signature scheme contains the following four algorithms:

- $\mathsf{Setup}(1^\lambda) \rightarrow \mathbb{PP}$: The system setup algorithm takes as input a security parameter λ to produce the system public parameters $\mathbb{PP}$.
- $\mathsf{KeyGen}() \rightarrow (SK, VK)$: The key generation algorithm generates users' signing and verification keys (SK, VK).
- $\mathsf{Sign}(SK_s, R, M) \rightarrow \sigma$: The signing algorithm takes as input a message M to be signed, a set of verification keys R (*i.e.*, the ring), and an user's signing key SK_s. It is required that $VK_s \in R$ meanings that the signer is a member of the signing ring. The algorithm outputs a signature σ.
- $\mathsf{Vrfy}(R, M, \sigma) \rightarrow 0/1$: The verification algorithm takes as input a purported signature σ on a ring R and a message M. It outputs 1 if σ is valid. Otherwise, it outputs 0.

5.2 Security Models of Ring Signatures

Security models of the ring signature scheme contains two parts: unforgeability and anonymity.

5.2.1 Ring Unforgeability

This security guarantees that an adversary can compute a valid signature on behalf of a ring only if he knows a secret key corresponding to one of them. In this work, we use Bender *et al.*'s [5] model: unforgeability with respect to insider corruption[3] which is defined by the following game:

1) **Setup:** The challenger runs Setup and KeyGen algorithms to generate public parameters and users' keys $\{(SK_i, VK_i)\}_{i=1}^{n(\lambda)}$. Then it gives the adversary $\mathcal{A}$ the system parameters and verification keys $S = \{VK_i\}_{i=1}^{n(\lambda)}$. In addition, the challenger maintains a set C to record the corrupted users, initially, $C \leftarrow \emptyset$.
2) **Signing queries:** The adversary $\mathcal{A}$ can adaptively make singing queries on inputs (M, R, s), where M is the message to be signed, $R \subseteq S$ is a ring of verification keys and s is an index such that $VK_s \in R$. The challenger returns back a ring signature $\sigma \leftarrow \mathsf{Sign}(SK_s, R, M)$ to $\mathcal{A}$.
3) **Corruption queries:** The adversary $\mathcal{A}$ also can adaptively make some corruption queries on input $s \in [n(\lambda)]$. The challenger returns back SK_s to $\mathcal{A}$ and adds VK_s into the set C.
4) **Output:** Finally, the adversary $\mathcal{A}$ outputs a tuple of (M^*, σ^*, R^*). We say that $\mathcal{A}$ wins the game if the following conditions hold: (1) $\mathsf{Vrfy}(R^*, M^*, \sigma^*) = 1$; (2) $R^* \subseteq S \backslash C$; (3) it never made a singing query (M^*, R^*, s) for any s.

We denote the success probability of a PPT adversary $\mathcal{A}$ (taken over the random choices of the challenger and adversary) to win the above game as $\mathbf{Adv}_{\mathcal{A}}^{Unf}$.

Definition 6. We say that a ring signature scheme has the property of unforgeability with respect to insider corruption, if for any PPT adversary $\mathcal{A}$, it cannot win the above game with non-negligible advantage.

Selective security. We define a weaker notion, selective security, to the above model. In the game of selective security, the adversary $\mathcal{A}$ is required that to give a forgery ring/message pair (R^*, M^*)[4] to the challenger before the setup phase, then it cannot make

[3] We make use of a weaker notion of this security model in which corruptions of honest users are allowed but adversary-chosen public keys are not allowed. This weaker notion has been used in [24, 27].

[4] In the beginning, $\mathcal{A}$ does not given the keys $S = \{VK_i\}_{i=1}^{n(\lambda)}$. In order to obtain the forgery ring R^*, we require that $\mathcal{A}$ outputs a set of index $I_{R^*} = \{i_1, \ldots, i_{|R^*|}\} \subseteq [n(\lambda)]$. Then, after the keys $S = \{VK_i\}_{i=1}^{n(\lambda)}$ be generated, the forgery ring $R^* = \{VK_{i_1}, \ldots, VK_{i_{|R^*|}}\} \subseteq S$ also be defined.

signing query on inputs (M^*, R^*, s) for any s, it also cannot make corruption query on input s for which $VK_s \in R^*$.

Definition 7. We say that a ring signature scheme is selectively unforgeable with respect to insider corruption, if for any PPT adversary $\mathcal{A}$, it cannot win the selective game with non-negligible advantage.

5.2.2 Ring Anonymity

This security guarantees that any verifier can be convinced that someone in the ring has generated a valid ring signature, but the real signer remains unknown. In this paper, we make use of the notion of perfect anonymity. We say that a ring signature scheme is perfectly anonymous, if a signature on a message M^* under a ring R^* and key VK_{i_0} looks exactly the same as a signature on the same message M^* under the same ring R^* and a different key VK_{i_1}. This means that the signer's key is hidden among all the honestly generated keys in the ring. Formally, it is defined by the following game:

1) **Setup:** The challenger runs Setup and KeyGen algorithms to generate public parameters and users' keys $\{(SK_i, VK_i)\}_{i=1}^{n(\lambda)}$. Then it returns back the public parameters and all keys $\{(SK_i, VK_i)\}_{i=1}^{n(\lambda)}$ to the adversary $\mathcal{A}$.

2) **Challenge:** The adversary $\mathcal{A}$ gives a tuple of (M^*, R^*, i_0, i_1), where M^* is the challenge message, R^* is the challenge ring, i_0 and i_1 are two indices such that $\{VK_{i_0}, VK_{i_1}\} \subseteq R^*$, to the challenger. The challenger chooses random $b \in \{0,1\}$, computes $\sigma^* \leftarrow \mathsf{Sign}(M^*, SK_{i_b}, R^*)$, and sends σ^* to the adversary.

3) **Guess:** Finally, the adversary outputs b', indicating his guess for b.

We denote the advantage of an unbounded adversary $\mathcal{A}$ (taken over the random choices of the challenger and the adversary) to win the above game as $\mathbf{Adv}_{\mathcal{A}}^{Ano} = |\Pr[b' = b] - \Pr[b' \neq b]|$.

Definition 8. A ring signature scheme has the property of perfect anonymity, if even an unbounded adversary cannot win the above game with non-negligible advantage.

5.3 Construction

We now construct a ring signature scheme based on the multilinear BLS scheme. We specify the message space $\mathcal{M} := \{0,1\}^\ell$, more generally, a collision resistant hash function can be used to hash messages to this size. Let $m_1, \ldots, m_\ell$ be the bits of the message $M \in \mathcal{M}$. The following construction is an n-user ring signature scheme, means that $|R| = n$.

- $\mathsf{Setup}(1^\lambda, n, \ell)$: Trusted authority takes as input a security parameter λ, the length ℓ of messages and ring size n to runs this algorithm to generate public parameters. It first runs $\mathbb{MP} = (\mathbb{G}_1, \ldots, \mathbb{G}_k, p, g, \ldots, g_k, \mathbf{e}_{i,j}) \leftarrow \mathsf{MulGen}(1^\lambda, k = n + \ell)$. Next, it chooses 2ℓ random values $(a_{1,0}, a_{1,1}), \ldots, (a_{\ell,0}, a_{\ell,1}) \in \mathbb{Z}_p^2$ and computes $A_{i,\beta} = g^{a_{i,\beta}} \in \mathbb{G}_1$, for $i \in [\ell]$ and $\beta \in \{0,1\}$. The public parameters $\mathbb{PP}$ contain the group descriptions $\mathbb{MP}$ and group elements $(A_{1,0}, A_{1,1}), \ldots, (A_{\ell,0}, A_{\ell,1})$.

- $\mathsf{KeyGen}(\mathbb{PP})$: Each user i chooses a random value $x_i \in \mathbb{Z}_p$ as his signing key SK_i. The corresponding verification key is $VK_i = g^{x_i} \in \mathbb{G}_1$.

- $\mathsf{Sign}(M, SK_s, R = \{VK_1, \ldots, VK_n\})$: Given a ring of n verification keys, the holder of signing key SK_s with $s \in [n]$ can sign some message $M \in \mathcal{M}$ as $\sigma = \mathbf{e}(A_{1,m_1}, \ldots, A_{\ell,m_\ell}, VK_1, \ldots, VK_{s-1}, VK_{s+1}, \ldots, VK_n)^{x_s}$. The signature consists of just a single group element. In fact, $\sigma = g_{k-1}^{(\prod_{i=1}^{\ell} a_{i,m_i}) \cdot (\prod_{j=1}^{n} x_j)} \in \mathbb{G}_{k-1}$.

- $\mathsf{Vrfy}(M, \sigma, R = \{VK_1, \ldots, VK_n\})$: Given a ring of n verification keys and a purported signature σ on a message M, check the following equation: $\mathbf{e}(\sigma, g) \stackrel{?}{=} \mathbf{e}(A_{1,m_1}, \ldots, A_{\ell,m_\ell}, VK_1, \ldots, VK_n)$.

Correctness. To see the correctness, a signature σ on message M and ring R is $g_{k-1}^{(\prod_{i=1}^{\ell} a_{i,m_i}) \cdot (\prod_{j=1}^{n} x_j)}$, and thus we have $\mathbf{e}(\sigma, g) = \mathbf{e}(g_{k-1}^{(\prod_{i=1}^{\ell} a_{i,m_i}) \cdot (\prod_{j=1}^{n} x_j)}, g) = \mathbf{e}(g_{k-1}^{\prod_{i=1}^{\ell} a_{i,m_i}}, g^{\prod_{j=1}^{n} x_j}) = \mathbf{e}(A_{1,m_1}, \ldots, A_{\ell,m_\ell}, VK_1, \ldots, VK_n)$.

In the setting of the multilinear maps, the space to represent a group element might grow with k (which is $n + \ell$), because this happens in the GGH [12] framework. To mitigate this problem, we can use the method in [17], which differs the message alphabet size in a tradeoff between computation and storage. The above construction uses a binary message alphabet. If it uses an alphabet of 2^d symbols, then the ring signature could resident in the group $\mathbb{G}_{\ell/d+n}$ with $\ell/d + n - 1$ pairings required to compute it, at the cost of the public parameters requiring $2^d \cdot \ell$ group elements in $\mathbb{G}_1$.

5.4 Security

Theorem 3. The ring signature scheme with message length ℓ and ring size n in the above is selectively unforgeable with respect to insider corruption under the $(n + \ell)$-MCDH assumption.

Proof. If there exists a PPT adversary $\mathcal{A}$ who can break the selective security of the ring signature scheme with advantage ϵ for message length ℓ, ring size n, level $k = n + \ell$ of multilinear maps, and security parameter λ, then we show that we can construct a PPT challenger $\mathcal{B}$ to break the k-MCDH assumption for security parameter λ with

probability ϵ. Initially, $\mathcal{A}$ gives $(M^* \in \{0,1\}^\ell, I_{R^*} = \{i_1, \ldots, i_n\} \subseteq [n(\lambda)])$ to $\mathcal{B}$ who is given an instance, $(\mathbb{MP}, g^{c_1}, \ldots, g^{c_k})$, of the k-MCDH assumption.

1) **Setup:** The challenger $\mathcal{B}$ first sets signing and verification keys for the challenge ring $(SK_{i_1} = c_{\ell+1}, VK_{i_1} = g^{c_{\ell+1}}), \ldots, (SK_{i_n} = c_k, VK_{i_n} = g^{c_k})$ (it does not know these c_i). For indices $i \notin I_{R^*}$, it chooses random $x_i \in \mathbb{Z}_p$ and sets $SK_i = x_i, VK_i = g^{x_i}$. It then generates parameters as follows:

   - Choose random integers $a_1, \ldots, a_\ell \in \mathbb{Z}_p$.
   - For $i \in [\ell]$, set $A_{i,m_i^*} = g^{c_i}$ and compute $A_{i,\bar{m_i^*}} = g^{a_i}$.

   Note that these parameters are distributed uniformly at random as in the real ring signature scheme. Then $\mathcal{B}$ sets the public parameters $\mathbb{PP} = (\mathbb{MP}, \{A_{i,\beta} | i \in [\ell], \beta \in \{0,1\}\})$. Finally, it gives $\mathbb{PP}$ and $\{VK_i\}_{i=1}^{n(\lambda)}$ to $\mathcal{A}$.

2) **Signing queries:** Conceptually, the challenger $\mathcal{B}$ can generate signatures for the adversary, because the adversary's requests and the challenge ring or message will be different in at least one bit. Specifically, when $\mathcal{A}$ makes a query to the signing oracle on input $(M, R = \{VK_1, \ldots, VK_n\}, s)$. If $R \neq R^*$, we assume that $VK_j \in R$ but $\notin R^*$, then $\mathcal{B}$ ignores the index s and signs M with SK_j in the usual way since $\mathcal{B}$ knows VK_j's singing key x_j. If $R = R^*$, then we know $M \neq M^*$ and assume that $m_\gamma \neq m_\gamma^*$, where m_γ and m_γ^* are the γ-th bit of the message M and M^*, respectively. Hence $\mathcal{B}$ can compute $\sigma = \mathbf{e}(A_{1,m_1}, \ldots, A_{\gamma-1,m_{\gamma-1}}, A_{\gamma+1,m_{\gamma+1}}, \ldots, A_{\ell,m_\ell}, VK_1, \ldots, VK_n)^{a_\gamma}$ by knowing the exponent a_γ of the parameter $A_{\gamma,\bar{m_\gamma^*}}$. Finally, it returns σ to the adversary $\mathcal{A}$.

3) **Corruption queries:** When $\mathcal{A}$ makes a query to the corruption oracle with input an index i for $i \notin I_{R^*}$, $\mathcal{B}$ gives SK_i to $\mathcal{A}$ and adds VK_i to the set C of the corrupted users.

4) **Output:** Finally, $\mathcal{A}$ outputs a forgery σ^* with respect to the challenge ring $R^* = \{VK_{i_1}, \ldots, VK_{i_n}\}$ and message M^*. Then $\mathcal{B}$ outputs σ^* as the solution to the given instance of the k-MCDH assumption. According to the setting of the public parameters and the verification keys of the challenge ring in the setup phase, and the assumption that σ^* is valid, we know that σ^* should be equal to $\mathbf{e}(A_{1,m_1^*}, \ldots, A_{\ell,m_\ell^*}, VK_1, \ldots, VK_{j-1}, VK_{j+1}, \ldots, VK_n)^{c_{\ell+j}} = g_{k-1}^{\prod_{i\in[1,k]} c_i}$, where $c_{\ell+j}$ is a certain signing key $\mathcal{A}$ uses. It implies that σ^* is a solution for the given instance to the k-MCDH problem, and thus $\mathcal{B}$ breaks the k-MCDH assumption.

It is clear that $\mathcal{B}$ succeeds whenever $\mathcal{A}$ does. □

Theorem 4. The ring signature scheme with message length ℓ and ring size n in the above is anonymous against any unbounded adversary.

Given a ring signature, we show that any ring member could possibly have created it. Consider a signature σ^* on ring $R^* = \{VK_1, \ldots, VK_n\}$ and message M^*, that has been created using key SK_{i_0}. We will show that with the same probability it could have been created using SK_{i_1} with $i_1 \neq i_0$. The proof is straight-forward.

Proof. For any tuple (M^*, R^*, i_0, i_1) which are chosen by an unbounded adversary $\mathcal{A}$, the signatures created by the member i_0 and i_1 are $\sigma_{i_0}^* = \mathbf{e}(A_{1,m_1^*}, \ldots, A_{\ell,m_\ell^*}, VK_1, \ldots, VK_{i_0-1}, VK_{i_0+1}, \ldots, VK_n)^{x_{i_0}}$ and $\sigma_{i_1}^* = \mathbf{e}(A_{1,m_1^*}, \ldots, A_{\ell,m_\ell^*}, VK_1, \ldots, VK_{i_1-1}, VK_{i_1+1}, \ldots, VK_n)^{x_{i_1}}$, respectively. However, $\sigma_{i_0}^* = \sigma_{i_1}^* = g_{k-1}^{(\prod_{i=1}^{\ell} a_{i,m_i^*}) \cdot (\prod_{i=1}^{n} x_i)}$ since the signing algorithm is deterministic. Therefore, any member of a ring can compute a same signature on a given message and ring. The perfect anonymity follows easily from this observation. □

6 Conclusion

In this work, we consider the BLS signature scheme in the setting of multilinear groups. First of all, we present a proof of adaptive security for the multilinear BLS scheme based on MCDH assumption. Then, we construct a ring signature scheme that, based on the multilinear BLS scheme, has an attractive feature that for n members of a ring the signatures consist of just a single group element.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under Grant 61702067, the Natural Science Foundation of Chongqing under Grants cstc2017jcyjAX0201 and cstc2019jcyj-msxmX0551, and the key project of science and technology research program of Chongqing Education Commission of China under Grants KJZD-K201803701 and KJQN201903701. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. R. Albrecht, P. Farshim, D. Hofheinz, E. Larraia, K. G. Paterson, "Multilinear maps from obfuscation," in *Proceedings of Part I of the 13th International Conference on Theory of Cryptography (TCC'16)*, vol. 9562, pp. 446-473, 2016.

[2] D. Boneh, X. Boyen, "Secure identity based encryption without random oracles," in *Annual International Cryptology Conference*, pp. 443-459, 2004.

[3] D. Boneh, X. Boyen, "Short signatures without random oracles," in *International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 3027, pp. 56-73, 2004.

[4] D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing," in *Annual International Cryptology Conference*, vol. 2139, pp. 213-229, 2001.
[5] A. Bender, J. Katz, R. Morselli, "Ring signatures: Stronger definitions, and constructions without random oracles," *Journal of Cryptolog*, vol. 22, no. 1, pp. 114-138, 2009.
[6] D. Boneh, B. Lynn, H. Shacham, "Short signatures from the Weil pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297-319, 2004.
[7] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme," in *International Workshop on Public Key Cryptography*, pp. 31-46, 2002.
[8] D. Boneh, A. Silverberg, "Applications of multilinear forms to cryptography," *Contemporary Mathematics*, vol. 324, pp. 71-90, 2002.
[9] J. S. Coron, T. Lepoint, M. Tibouchi, "New multilinear maps over the integers," in *Annual Cryptology Conference*, vol. 9215, pp. 267-286, 2015.
[10] L. Chen, J. Malone-Lee, "Improved identity-based signcryption," *International Workshop on Public Key Cryptography*, vol. 3386, pp. 362-379, 2005.
[11] E. S. V. Freire, D. Hofheinz, K. G. Paterson, C. Striecks, "Programmable hash functions in the multilinear setting," in *Annual Cryptology Conference*, vol. 8042, pp. 513-530, 2013.
[12] S. Garg, C. Gentry, S. Halevi, "Candidate multilinear maps from ideal lattices," *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 7881, pp. 1-17, 2013.
[13] S. Garg, C. Gentry, S. Helevi, M. Raykova, A. Sahai, B. Waters, "Canditate indistinguishability obfuscation and functional encryption for all circuits," *IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 40-49, 2013. ISBN: 978-0-7695-5135-7.
[14] C. Gentry, S. Gorbunov, S. Halevi, "Graph-induced multilinear maps from lattices," in *Theory of Cryptography Conference*, vol. 9015, pp. 498-527, 2015.
[15] Y. Hu, H. Jia, "Cryptanalysis of GGH map," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 9665, pp. 537-565, 2016.
[16] M. S. Hwang, C. C. Lee, Y. C. Lai, "An untraceable blind signature scheme," *IEICE Transactions on Foundations*, vol. E86-A, no. 7, pp. 1902-1906, 2003.
[17] S. Hohenberger, A. Sahai, B. Waters, "Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures," in *Annual Cryptology Conference*, vol. 8024, pp. 494-512, 2013.
[18] S. Hohenberger, A. Sahai, B. Waters, "Replacing a random oracle: full domain hash from indistinguishability obfuscation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 8441, pp. 201-220, 2004.
[19] M. S. Hwang, C. C. Lee, Y. C. Lai, "An untraceable blind signature scheme", *IEICE Transactions on Foundations*, vol. E86-A, no. 7, pp. 1902–1906, July 2003.
[20] C. C. Lee, M. S. Hwang, and W. P. Yang, "A new blind signature based on the discrete logarithm problem for untraceability", *Applied Mathematics and Computation*, vol. 164, no. 3, pp. 837–841, May 2005.
[21] H. T. Lee, H. J. Seo, "Security analysis of multilinear maps over the integers," in *Annual Cryptology Conference*, vol. 8616, pp. 224-240, 2014.
[22] A. Langlois, D. Stehlé, R. Steinfeld, "GGHLite: More efficient multilinear maps from ideal lattices," *Advances in Cryptology*, vol. 8441, pp. 239-256, 2014.
[23] R. Rivest, A. Shamir, Y. Tauman, "How to leak a secret," in *International Conference on the Theory and Application of Cryptology and Information Security*, vol. 2248, pp. 552-565, 2001.
[24] S. SchageS, J. Schwenk, "A CDH-based ring signature scheme with short signatures and public keys," in *International Conference on Financial Cryptography and Data Security*, vol. 6052, pp. 129-142, 2010.
[25] B. Waters, "Efficient identity-based encryption without random oracles," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 3494, pp. 114-127, 2005.
[26] H. Wang, D. He, J. Shen, Z. Zheng, X. Yang, M. H. Au, "Fuzzy matching and direct revocation: A new CP-ABE scheme from multilinear maps," *Soft Computing*, vol. 22, no. 7, pp. 2267-2274, 2017.
[27] F. Tang, H. Li, "Ring signatures of constant size without random oracles," in *International Conference on Information Security and Cryptology*, vol. 8957, pp. 93-108, 2015.
[28] F. Tang, H. Li, B. Liang, "Attribute-based signatures for circuits from multilinear maps," in *International Conference on Information Security*, vol. 8783, pp. 54-71, 2014.
[29] F. Tang, Y. Zhou, "Policy-based signatures for predicates," *International Journal of Network Security*, vol. 19, no. 5, pp. 811-822, 2017.

Biography

Fei Tang received his Ph.D from the Institute of Information Engineering of Chinese Academy of Sciences in 2015. He is currently an associate professor of the College of Cyberspace Security and Law, Chongqing University of Posts and Telecommunications. His research interests are public key cryptography and blockchain.

Dong Huang received his Ph.D from the Chongqing University in 2012. He is currently a professor of the Chongqing University of Science and Technology and Chongqing Vocational and Technical University of Mechatronics. His research interest is public key cryptography.

Eighth Power Residue Double Circulant Self-Dual Codes

Changsong Jiang[1,3], Yuhua Sun[1,2], and Xueting Liang[1]
(Corresponding author: Yuhua Sun)
College of Science, China University of Petroleum[1]
Qingdao, Shandong 266580, China
Provincial Key Laboratory of Applied Mathematics, Putian University, Putian, Fujian 351100, China[2]
School of Computer Science and Engineering, University of Electronic Science and Technology of China[3]
(Email: sunyuhua_1@163.com)
(Received Mar. 1, 2019; Revised and Accepted Sept. 16, 2019; First Online Jan. 23, 2020)

Abstract

Self-dual codes are one of the most important classes of linear codes. Power residue classes are widely used in the constructions of linear codes and pseudo-random sequences. In this paper, we give new constructions of self-dual codes over GF(2) and GF(4) by eighth power residues. We get multiple pure double circulant codes and bordered double circulant codes. Some of these new self-dual codes have large minimum distances.

Keywords: Cyclotomic Number; Double Circulant Code; Eighth Power Residues; Self-Dual Code

1 Introduction

The famous paper "A mathematical theory of communication" [26] by Shannon marked the beginning of coding theory. Codes with good properties have many applications in cryptography and communication systems. Most of the codes constructed in the initial stage were binary codes. Now, codes over finite fields and over finite rings are very common in both mathematical and engineering literatures. Thanks to having neat mathematical structure and being easy to code and decode, linear codes play a decisive role in coding theory. It is worth noting that, among linear codes, there is one class of special codes, *i.e.*, self-dual codes which are widely used in data transmission and have become important tools to construct quantum error-correcting codes. Therefore various methods of construction and analysis of self-dual codes have been presented by coding researchers and various classes of linear codes with self-dual property appeared successively in many literatures. For example, readers can refer to [1–6, 9, 11, 17–19, 22, 24, 25, 28, 29] or can also refer to the survey paper [13] for the advances of early research in this field. It is well known that power residue classes have become an important tool to construct stream cipher sequences with good pseudo-random properties (for example, see [7, 23, 27]). In fact, they have also been used to construct error-correcting codes, and a very interesting method of constructing linear codes or self-dual codes is combining double circulant matrices and residue classes to give the generator matrix of codes(for example, see [8, 10, 12, 16, 21])). But, in most of the relevant literatures at present, the residue being used to construct codes are mainly quadratic residue.

Recently, Zhang and Ge introduced fourth power residue double circulant and obtained several new infinite families of classes of self-dual codes over GF(2), GF(3), GF(4), GF(8), GF(9) [30]. Some of these codes have better minimum weight than previously known codes. In this paper, inspired by their methods, we construct double circulant self-dual codes by by higher power residues, especially eighth power residues. We give new constructions of self-dual codes over GF(2) and GF(4) by prime p of the form $16f + 9$, and some of these codes have good parameters. Examples of such codes are binary self-dual [82, 41, 14] code, quaternary self-dual [82, 41, 14] code and quaternary self-dual [84, 42, 12] code. All computation have been done by MATLAB R2017b and MAGMA V2.12 on a 2.50 GHz CPU.

The paper is organized as follows. In Section 2, we give the relevant knowledge of double circulant codes and self-dual codes. In Section 3, we describe the detailed process of constructing linear codes by eighth power residues and discuss the parameter conditions satisfying self-dual property. Section 4 considers the constructions over GF(2) and GF(4) respectively. A conclusion is given in Section 5.

2 Preliminaries

Self-Dual Codes. A linear $[n, k]$ code C of length n and dimension k over the Galois field with q elements GF(q) is a linear subspace of dimension k of

$\mathrm{GF}(q)^n$, where q is a prime power. An element of the code C is called a codeword of C. A generator matrix of C is a matrix whose rows generate C. Let $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ be two codewords of $\mathrm{GF}(q)^n$. The Euclidean inner product is defined by $(x, y) = \sum_{i=1}^{n} x_i y_i$. For a linear code C, the code $C^{\perp} = \{x \in \mathrm{GF}(q)^n | (x, c) = 0 \text{ for all } c \in C\}$ is called its Euclidean dual code. And we say C is self-orthogonal if $C \subseteq C^{\perp}$ and C is self-dual if $C = C^{\perp}$.

Definition 1. *[30]: Let $P_n(R)$ and $B_n(R)$ be codes with generator matrices of the form*

$$(I_n \quad R) \tag{1}$$

and

$$\begin{pmatrix} & \alpha & 1 & \cdots & 1 \\ & -1 & & & \\ I_{n+1} & \vdots & & R & \\ & -1 & & & \end{pmatrix} \tag{2}$$

respectively, where $\alpha \in \mathrm{GF}(q)$, I is the identity matrix and R is an $n \times n$ circulant matrix. An n by n circulant matrix has the form

$$\begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{n-1} \\ r_{n-1} & r_0 & r_1 & \cdots & r_{n-2} \\ \vdots & \vdots & \vdots & & \vdots \\ r_1 & r_2 & r_3 & \cdots & r_0 \end{pmatrix} \tag{3}$$

so that each successive row is a cyclic shift of the previous one. The codes $P_n(R)$ and $B_n(R)$ are called pure double circulant and bordered double circulant, respectively.

The (Hamming) distance between two codewords x and y denoted by $d(x, y)$, is defined to be the number of places at which x and y differ. The Hamming weight of a codeword is the number of non-zero components. And the minimum distance $d(C)$ of C is defined by $d(C) = \min\{d(x, y) | x \neq y \in C\}$, and it also equals to the minimum weight of the codewords of C except for 0.

Let C be a self-dual code over $\mathrm{GF}(q)$ of length n and minimum distance d(C). Then the following bounds are known in [14, 22, 24, 25]. For binary self-dual codes:

$$d(C) \leq \begin{cases} 4[\frac{n}{24}] + 4, & if\ n \neq 22 \pmod{24}, \\ 4[\frac{n}{24}] + 6, & if\ n = 22 \pmod{24}. \end{cases}$$

The minimum distance of a self-dual ternary code C satisfies: $d(C) \leq 3[\frac{n}{12}] + 3$ and for quaternary Euclidean self-dual codes: $d(C) \leq 4[\frac{n}{12}] + 4$. The code C is called extremal if the equality holds. If a code has the highest possible minimum weight for its length and dimension, we call it optimal.

In this paper, we construct a circulant matrix R by eighth power residue and get a necessary condition such that the corresponding codes are self-dual. Further, under this condition, we get two kinds of codes called pure eighth power residue double circulant code and bordered eighth power residue double circulant code, respectively. Some codes have large minimum distances, and almost reach the bounds of the minimum distance.

3 Generator Matrices of Eighth Power Residue Double Circulant Self-Dual Codes

Let $p = Nf + 1$ be a prime with a fixed primitive root g over $\mathrm{GF}(q)$. We define the Nth cyclotomic classes $C_0, C_1, ..., C_{N-1}$ of $\mathrm{GF}(p)$ by

$$C_i = \left\{ g^{jN+i} | 0 \leq j \leq f - 1 \right\},$$

where $0 \leq i \leq N - 1$. Then we call C_0 is the Nth power residues modulo p, and $C_i = g^i C_0$ where $0 \leq i \leq N - 1$. Define the cyclotomic number (i, j) of order N to be the number of integers $n \pmod p$ which satisfy

$$n \equiv g^{16s+i}, \quad 1 + n \equiv g^{16t+j} (\mathrm{mod}\ p),$$

where s, t in $\{0, 1, 2, ..., f - 1\}$.

In order to give the necessary conditions, we give the eighth power residue cyclotomic numbers and derive the relationships between them when p is an odd prime of the form $16l + 9$.

Lemma 1. *[15]: Let $p = ef + 1$ be an odd prime. Then*

1) $(i, j)_e = (i', j')_e$, when $i \equiv i' (\mathrm{mod}\ e)$ and $j \equiv j' (\mathrm{mod}\ e)$.

2) $$(i, j)_e = (e - i, j - i)_e = \begin{cases} (j, i)_e; & if\ f\ even. \\ (j + \frac{e}{2}, i + \frac{e}{2})_e; & if\ f\ odd. \end{cases}$$

3) $\sum_{i=0}^{e-1} (i, j)_e = f - \delta_j$, where $\delta_j = 1$ if $j \equiv 0 \pmod e$; otherwise $\delta_j = 0$.

Let p be a prime of the form $p = 16l + 9$, where l is a positive integer. From Lemma 1, the relationships of cyclotomic numbers of order 8 are

$$\begin{cases} (0,0)_8 = (4,0)_8 = (4,4)_8, & (0,1)_8 = (3,7)_8 = (5,4)_8, \\ (0,2)_8 = (2,6)_8 = (6,4)_8, & (0,3)_8 = (1,5)_8 = (7,4)_8, \\ (0,4)_8, & (0,5)_8 = (1,4)_8 = (7,3)_8, \\ (0,6)_8 = (2,4)_8 = (6,2)_8, & (0,7)_8 = (3,4)_8 = (5,1)_8, \\ (1,0)_8 = (3,3)_8 = (4,1)_8 = (4,5)_8 = (5,0)_8 = (7,7)_8, \\ (1,1)_8 = (3,0)_8 = (4,3)_8 = (4,7)_8 = (5,5)_8 = (7,0)_8, \\ (1,2)_8 = (2,7)_8 = (3,6)_8 = (5,3)_8 = (6,5)_8 = (7,1)_8, \\ (1,3)_8 = (1,6)_8 = (2,5)_8 = (6,3)_8 = (7,2)_8 = (7,5)_8, \\ (1,7)_8 = (2,3)_8 = (3,5)_8 = (5,2)_8 = (6,1)_8 = (7,6)_8, \\ (2,0)_8 = (2,2)_8 = (4,2)_8 = (4,6)_8 = (6,0)_8 = (6,6)_8, \\ (2,1)_8 = (3,1)_8 = (3,2)_8 = (5,6)_8 = (5,7)_8 = (6,7)_8. \end{cases} \tag{4}$$

Remark 1. *For simplicity, in the next we denote*

$$\begin{cases} A := (0,0)_8, & B := (0,1)_8, & C := (0,2)_8, \\ D := (0,3)_8, & E := (0,4)_8, & F := (0,5)_8, \\ G := (0,6)_8, & H := (0,7)_8, & I := (1,0)_8, \\ J := (1,1)_8, & K := (1,2)_8, & L := (1,3)_8, \\ M := (1,7)_8, & N := (2,0)_8, & O := (2,1)_8. \end{cases} \tag{5}$$

Let $p \equiv 1 (\text{mod } 8)$ be a prime. Its 8th cyclotomic classes are $C_0, C_1, C_2, C_3, C_4, C_5, C_6$ and C_7. Suppose $m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8$ are the elements of GF(q). Then we construct the matrix $C_p(m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8)$ which is a $p \times p$ matrix on GF(q). The component $c_{ij}, 1 \le i, j \le p$, defines

$$\begin{cases} m_0; & \text{if} \quad j = i, \\ m_1; & \text{if} \quad j - i \in C_0, \\ m_2; & \text{if} \quad j - i \in C_1, \\ m_3; & \text{if} \quad j - i \in C_2, \\ m_4; & \text{if} \quad j - i \in C_3, \\ m_5; & \text{if} \quad j - i \in C_4, \\ m_6; & \text{if} \quad j - i \in C_5, \\ m_7; & \text{if} \quad j - i \in C_6, \\ m_8; & \text{if} \quad j - i \in C_7. \end{cases} \tag{6}$$

Let I_n be the identity matrix and J_n be the all-one square matrix, so that $C_p(1,0,0,0,0,0,0,0,0) = I_p$ and $C_p(1,1,1,1,1,1,1,1,1) = J_p$. Denote

$$\begin{array}{ll} A_1 := C_p(0,1,0,0,0,0,0,0,0), & A_2 := C_p(0,0,1,0,0,0,0,0,0), \\ A_3 := C_p(0,0,0,1,0,0,0,0,0), & A_4 := C_p(0,0,0,0,1,0,0,0,0), \\ A_5 := C_p(0,0,0,0,0,1,0,0,0), & A_6 := C_p(0,0,0,0,0,0,1,0,0), \\ A_7 := C_p(0,0,0,0,0,0,0,1,0), & A_8 := C_p(0,0,0,0,0,0,0,0,1). \end{array} \tag{7}$$

And the construction of the $n \times n$ circulant matrix R is given as follows:

$$R = m_0 I_p + m_1 A_1 + m_2 A_2 + m_3 A_3 + m_4 A_4 + m_5 A_5 + m_6 A_6 + m_7 A_7 + m_8 A_8 \tag{8}$$

Lemma 2. *Let $p = 16l + 9$ be a prime, then the matrices in equation (7) have the following relationships.*

$$\begin{array}{l} A_1 = A_5^t, A_2 = A_6^t, A_3 = A_7^t, A_4 = A_8^t, \\ A_1^2 = AA_1 + BA_2 + CA_3 + DA_4 + EA_5 + FA_6 + GA_7 + HA_8, \\ A_2^2 = HA_1 + AA_2 + BA_3 + CA_4 + DA_5 + EA_6 + FA_7 + GA_8, \\ A_3^2 = GA_1 + HA_2 + AA_3 + BA_4 + CA_5 + DA_6 + EA_7 + FA_8, \\ A_4^2 = FA_1 + GA_2 + HA_3 + AA_4 + BA_5 + CA_6 + DA_7 + EA_8, \\ A_5^2 = EA_1 + FA_2 + GA_3 + HA_4 + AA_5 + BA_6 + CA_7 + DA_8, \\ A_6^2 = DA_1 + EA_2 + FA_3 + GA_4 + HA_5 + AA_6 + BA_7 + CA_8, \\ A_7^2 = CA_1 + DA_2 + EA_3 + FA_4 + GA_5 + HA_6 + AA_7 + BA_8, \\ A_8^2 = BA_1 + CA_2 + DA_3 + EA_4 + FA_5 + GA_6 + HA_7 + AA_8, \\ A_1A_2 = A_2A_1 = IA_1 + JA_2 + KA_3 + LA_4 + FA_5 + DA_6 + LA_7 + MA_8, \\ A_1A_3 = A_3A_1 = NA_1 + OA_2 + NA_3 + MA_4 + GA_5 + LA_6 + CA_7 + KA_8, \\ A_1A_4 = A_4A_1 = JA_1 + OA_2 + OA_3 + IA_4 + HA_5 + MA_6 + KA_7 + BA_8, \\ A_1A_5 = A_5A_1 = (2l+1)I_p + AA_1 + IA_2 + AA_3 + JA_4 + AA_5 + IA_6 \\ \qquad + NA_7 + JA_8, \\ A_1A_6 = A_6A_1 = IA_1 + HA_2 + MA_3 + KA_4 + BA_5 + JA_6 + OA_7 + OA_8, \\ A_1A_7 = A_7A_1 = NA_1 + MA_2 + GA_3 + LA_4 + CA_5 + KA_6 + NA_7 + OA_8, \\ A_1A_8 = A_8A_1 = JA_1 + AA_2 + LA_3 + FA_4 + DA_5 + LA_6 + MA_7 + IA_8, \\ A_2A_3 = A_3A_2 = MA_1 + IA_2 + JA_3 + KA_4 + LA_5 + FA_6 + DA_7 + LA_8, \\ A_2A_4 = A_4A_2 = KA_1 + NA_2 + OA_3 + NA_4 + MA_5 + GA_6 + LA_7 + CA_8, \\ A_2A_5 = A_5A_2 = BA_1 + JA_2 + OA_3 + OA_4 + IA_5 + HA_6 + MA_7 + KA_8, \\ A_2A_6 = A_6A_2 = JA_1 + AA_2 + IA_3 + NA_4 + JA_5 + AA_6 + IA_7 + NA_8, \\ A_2A_7 = A_7A_2 = OA_1 + IA_2 + AA_3 + MA_4 + KA_5 + BA_6 + JA_7 + OA_8, \\ A_2A_8 = A_8A_2 = OA_1 + NA_2 + MA_3 + GA_4 + LA_5 + CA_6 + KA_7 + NA_8, \\ A_3A_4 = A_4A_3 = LA_1 + MA_2 + IA_3 + JA_4 + KA_5 + LA_6 + FA_7 + DA_8, \\ A_3A_5 = A_5A_3 = CA_1 + KA_2 + NA_3 + OA_4 + NA_5 + MA_6 + GA_7 + LA_8, \\ A_3A_6 = A_6A_3 = KA_1 + BA_2 + JA_3 + OA_4 + OA_5 + IA_6 + HA_7 + MA_8, \\ A_3A_7 = A_7A_3 = (2l+1)I_p + NA_1 + JA_2 + AA_3 + IA_4 + NA_5 + JA_6 \\ \qquad + AA_7 + IA_8, \\ A_3A_8 = A_8A_3 = OA_1 + OA_2 + IA_3 + HA_4 + MA_5 + KA_6 + BA_7 + JA_8, \\ A_4A_5 = A_5A_4 = DA_1 + MA_2 + LA_3 + IA_4 + JA_5 + KA_6 + LA_7 + FA_8, \\ A_4A_6 = A_6A_4 = LA_1 + CA_2 + KA_3 + NA_4 + OA_5 + NA_6 + MA_7 + GA_8, \\ A_4A_7 = A_7A_4 = MA_1 + KA_2 + BA_3 + JA_4 + OA_5 + OA_6 + IA_7 + HA_8, \\ A_4A_8 = A_8A_4 = (2l+1)I_p + IA_1 + NA_2 + JA_3 + AA_4 + IA_5 + NA_6 \\ \qquad + JA_7 + AA_8, \\ A_5A_6 = A_6A_5 = FA_1 + DA_2 + LA_3 + MA_4 + IA_5 + JA_6 + KA_7 + LA_8, \\ A_5A_7 = A_7A_5 = GA_1 + LA_2 + CA_3 + KA_4 + NA_5 + OA_6 + NA_7 + MA_8, \\ A_5A_8 = A_8A_5 = HA_1 + MA_2 + KA_3 + BA_4 + JA_5 + OA_6 + OA_7 + IA_8, \\ A_6A_7 = A_7A_6 = LA_1 + FA_2 + DA_3 + LA_4 + MA_5 + IA_6 + JA_7 + KA_8, \\ A_6A_8 = A_8A_6 = MA_1 + GA_2 + LA_3 + CA_4 + KA_5 + NA_6 + OA_7 + NA_8, \\ A_7A_8 = A_8A_7 = KA_1 + LA_2 + FA_3 + DA_4 + LA_5 + MA_6 + IA_7 + JA_8. \end{array} \tag{9}$$

Proof. The proof is straightforward from the definition of A_i and lemma 1. □

Lemma 3. *If $p = 16l + 9$ is a prime, then*

$$RR^t = \alpha_0 I_p + \alpha_1 A_1 + \alpha_2 A_2 + \alpha_3 A_3 + \alpha_4 A_4 + \alpha_5 A_5 + \alpha_6 A_6 + \alpha_7 A_7 + \alpha_8 A_8 \tag{10}$$

where

$$\begin{array}{l} \alpha_0 = m_0^2 + \frac{p-1}{8}(m_1^2 + m_2^2 + m_3^2 + m_4^2 + m_5^2 + m_6^2 + m_7^2 + m_8^2), \\ \alpha_1 = \alpha_5 = (m_0m_1 + m_0m_5) + (m_1^2 + m_1m_5 + m_5^2)A \\ \quad + (m_1m_2 + m_4m_8 + m_5m_6)B + (m_1m_3 + m_3m_7 + m_5m_7)C \\ \quad + (m_1m_4 + m_2m_6 + m_5m_8)D + m_1m_5E \\ \quad + (m_1m_6 + m_2m_5 + m_4m_8)F + (m_1m_7 + m_3m_5 + m_3m_7)G \\ \quad + (m_1m_8 + m_2m_6 + m_4m_5)H \\ \quad + (m_1m_2 + m_1m_6 + m_2m_5 + m_4^2 + m_5m_6 + m_8^2)I \\ \quad + (m_1m_4 + m_1m_8 + m_2^2 + m_4m_5 + m_5m_8 + m_6^2)J \\ \quad + (m_2m_3 + m_2m_8 + m_3m_8 + m_4m_6 + m_4m_7 + m_6m_7)K \\ \quad + (m_2m_4 + m_2m_7 + m_3m_6 + m_3m_8 + m_4m_7 + m_6m_8)L \\ \quad + (m_2m_7 + m_2m_8 + m_3m_4 + m_3m_6 + m_4m_6 + m_7m_8)M \\ \quad + (m_1m_3 + m_1m_7 + m_3^2 + m_3m_5 + m_5m_7 + m_7^2)N \\ \quad + (m_2m_3 + m_2m_4 + m_3m_4 + m_6m_7 + m_6m_8 + m_7m_8)O, \\ \alpha_2 = \alpha_6 = (m_0m_2 + m_0m_6) + (m_2^2 + m_2m_6 + m_6^2)A \\ \quad + (m_1m_5 + m_2m_3 + m_6m_7)B + (m_2m_4 + m_4m_8 + m_6m_8)C \\ \quad + (m_1m_6 + m_2m_5 + m_3m_7)D + m_2m_6E \\ \quad + (m_1m_5 + m_2m_7 + m_3m_6)F + (m_2m_8 + m_4m_6 + m_4m_8)G \\ \quad + (m_1m_2 + m_3m_7 + m_5m_6)H \\ \quad + (m_1^2 + m_2m_3 + m_2m_7 + m_3m_6 + m_5^2 + m_6m_7)I \\ \quad + (m_1m_2 + m_1m_6 + m_2m_5 + m_3^2 + m_5m_6 + m_7^2)J \\ \quad + (m_1m_3 + m_1m_4 + m_3m_4 + m_5m_7 + m_5m_8 + m_7m_8)K \\ \quad + (m_1m_4 + m_1m_7 + m_3m_5 + m_3m_8 + m_4m_7 + m_5m_8)L \\ \quad + (m_1m_3 + m_1m_8 + m_3m_8 + m_4m_5 + m_4m_7 + m_5m_7)M \\ \quad + (m_2m_4 + m_2m_8 + m_4^2 + m_4m_6 + m_6m_8 + m_8^2)N \\ \quad + (m_1m_7 + m_1m_8 + m_3m_4 + m_3m_5 + m_4m_5 + m_7m_8)O, \\ \alpha_3 = \alpha_7 = (m_0m_3 + m_0m_7) + (m_3^2 + m_3m_7 + m_7^2)A \\ \quad + (m_2m_6 + m_3m_4 + m_7m_8)B + (m_1m_5 + m_1m_7 + m_3m_5)C \\ \quad + (m_2m_7 + m_3m_6 + m_4m_8)D + m_3m_7E \\ \quad + (m_2m_6 + m_3m_8 + m_4m_7)F + (m_1m_3 + m_1m_5 + m_5m_7)G \\ \quad + (m_2m_3 + m_4m_8 + m_6m_7)H \\ \quad + (m_2^2 + m_3m_4 + m_3m_8 + m_4m_7 + m_6^2 + m_7m_8)I \\ \quad + (m_2m_3 + m_2m_7 + m_3m_6 + m_4^2 + m_6m_7 + m_8^2)J \\ \quad + (m_1m_6 + m_1m_8 + m_2m_4 + m_2m_5 + m_4m_5 + m_6m_8)K \\ \quad + (m_1m_4 + m_1m_6 + m_2m_5 + m_2m_8 + m_4m_6 + m_5m_8)L \\ \quad + (m_1m_2 + m_1m_4 + m_2m_4 + m_5m_6 + m_5m_8 + m_6m_8)M \\ \quad + (m_1^2 + m_1m_3 + m_1m_7 + m_3m_5 + m_5^2 + m_5m_7)N \\ \quad + (m_1m_2 + m_1m_8 + m_2m_8 + m_4m_5 + m_4m_6 + m_5m_6)O, \end{array}$$

$$\begin{aligned}\alpha_4 = \alpha_8 &= (m_0m_4 + m_0m_8) + (m_4^2 + m_4m_8 + m_8^2)A\\ &+(m_1m_8 + m_3m_7 + m_4m_5)B + (m_2m_6 + m_2m_8 + m_4m_6)C\\ &+(m_1m_5 + m_3m_8 + m_4m_7)D + m_4m_8E\\ &+(m_1m_4 + m_3m_7 + m_5m_8)F + (m_2m_4 + m_2m_6 + m_6m_8)G\\ &+(m_1m_5 + m_3m_4 + m_7m_8)H\\ &+(m_1m_4 + m_1m_8 + m_3^2 + m_4m_5 + m_5m_8 + m_7^2)I\\ &+(m_1^2 + m_3m_4 + m_3m_8 + m_4m_7 + m_5^2 + m_7m_8)J\\ &+(m_1m_2 + m_1m_7 + m_2m_7 + m_3m_5 + m_3m_6 + m_5m_6)K\\ &+(m_1m_3 + m_1m_6 + m_2m_5 + m_2m_7 + m_3m_6 + m_5m_7)L\\ &+(m_1m_6 + m_1m_7 + m_2m_3 + m_2m_5 + m_3m_5 + m_6m_7)M\\ &+(m_2^2 + m_2m_4 + m_2m_8 + m_4m_6 + m_6^2 + m_6m_8)N\\ &+(m_1m_2 + m_1m_3 + m_2m_3 + m_5m_6 + m_5m_7 + m_6m_7)O.\end{aligned}$$

Proof. The result comes from Lemma 2, Lemma 3 and a complex computation. □

In order to facilitate, we denote

$$\begin{aligned}&\overrightarrow{m} := (m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8) \in \mathrm{GF}(q)^9,\\ &D_0(\overrightarrow{m}) := \alpha_0,\\ &D_1(\overrightarrow{m}) := \alpha_1 = \alpha_5,\\ &D_2(\overrightarrow{m}) := \alpha_2 = \alpha_6,\\ &D_3(\overrightarrow{m}) := \alpha_3 = \alpha_7,\\ &D_4(\overrightarrow{m}) := \alpha_4 = \alpha_8.\end{aligned} \tag{11}$$

Theorem 1. *Let p be an odd prime of the form $16l+9$ and q be a prime power. Suppose $\alpha \in \mathrm{GF}(q)$, $\overrightarrow{m} \in \mathrm{GF}(q)^9$. Then*

(1) *pure eighth power residue double circulant code $P_p(\overrightarrow{m})$ is self-dual over $\mathrm{GF}(q)$ when the following conditions hold:*

$$\begin{cases} D_0(\overrightarrow{m}) = -1,\\ D_1(\overrightarrow{m}) = 0,\\ D_2(\overrightarrow{m}) = 0,\\ D_3(\overrightarrow{m}) = 0,\\ D_4(\overrightarrow{m}) = 0. \end{cases} \tag{12}$$

(2) *bordered eighth power residue double circulant code $B_p(\alpha, \overrightarrow{m})$ is self-dual over $\mathrm{GF}(q)$ when the following conditions hold:*

$$\begin{cases} \alpha^2 + p = -1,\\ -\alpha + m_0 + \frac{p-1}{8}(m_1 + m_2 + m_3 +\\ \qquad m_4 + m_5 + m_6 + m_7 + m_8) = 0,\\ D_0(\overrightarrow{m}) = -2,\\ D_1(\overrightarrow{m}) = -1,\\ D_2(\overrightarrow{m}) = -1,\\ D_3(\overrightarrow{m}) = -1,\\ D_4(\overrightarrow{m}) = -1. \end{cases} \tag{13}$$

Proof. According to Lemma 3,

$$\begin{aligned}P_p(\overrightarrow{m})P_p(\overrightarrow{m})^t = I_p &+ D_0(\overrightarrow{m})I_p + D_1(\overrightarrow{m})A_1 + D_2(\overrightarrow{m})A_2\\ &+D_3(\overrightarrow{m})A_3 + D_4(\overrightarrow{m})A_4 + D_1(\overrightarrow{m})A_5\\ &+D_2(\overrightarrow{m})A_6 + D_3(\overrightarrow{m})A_7 + D_4(\overrightarrow{m})A_8,\end{aligned} \tag{14}$$

and

$$B_p(\overrightarrow{m})B_p(\overrightarrow{m})^t = (I_{p+1} \quad K)\begin{pmatrix} I_{p+1} \\ K^t \end{pmatrix} = I_{p+1} + KK^t, \tag{15}$$

where

$$\begin{aligned}KK^t &= \begin{pmatrix} \alpha & 1 & \cdots & 1 \\ -1 & & & \\ \vdots & & R & \\ -1 & & & \end{pmatrix} \cdot \begin{pmatrix} \alpha & -1 & \cdots & -1 \\ 1 & & & \\ \vdots & & R^t & \\ 1 & & & \end{pmatrix}\\ &= \begin{pmatrix} \alpha^2 + p & S & \cdots & S \\ S & & & \\ \vdots & & X & \\ S & & & \end{pmatrix}_{(p+1)\times(p+1)}\end{aligned} \tag{16}$$

and

$$\begin{aligned}&X = J_p + D_0(\overrightarrow{m})I_p + D_1(\overrightarrow{m})A_1 + D_2(\overrightarrow{m})A_2 + D_3(\overrightarrow{m})A_3\\ &+D_4(\overrightarrow{m})A_4 + D_1(\overrightarrow{m})A_5 + D_2(\overrightarrow{m})A_6 + D_3(\overrightarrow{m})A_7 + D_4(\overrightarrow{m})A_8,\\ &S = -\alpha + m_0 + \frac{p-1}{8}(m_1 + m_2 + m_3 + m_4\\ &\qquad + m_5 + m_6 + m_7 + m_8).\end{aligned} \tag{17}$$

□

The result can be obtained by the definition of self-dual codes.

4 Eighth Power Residue Double Circulant Self-Dual Codes Over GF(2) and GF(4)

In this section, we give some constructions of self-dual codes over GF(2) and GF(4) by MATLAB and MAGMA. And the corresponding minimum hamming distances are solved. Some codes have good minimum distances, even almost satisfy the bounds.

Theorem 2. *Let p be an odd prime of the form $16l + 9$, several pure eighth power residue double circulant self-dual codes whose generator matrix satisfies the form of $P_p(m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8)$ of length $2p$ over $\mathrm{GF}(2)$ are obtained. The parameters that satisfy the conditions are listed in the following table when $p = 41 = 16 \times 2 + 9$.*

When $p = 41$, the length n of the code is 82, so that the bound of the minimum hamming distance is 16. By our method, the minimum hamming distance of the codes has a maximum of 14, which almost satisfies the bound. The self-dual [82, 41, 14] codes over GF(2) with a good property are obtained.

Theorem 3. *Let ξ be the fixed primitive element of $\mathrm{GF}(4)$ satisfying $\xi^2+\xi+1 = 0$ and p be an odd prime of the form $16l + 9$, pure eighth power residue double circulant self-dual codes $P_p(\overrightarrow{m})$ of length $2p$ over $\mathrm{GF}(4)$ and bordered eighth power residue double circulant codes $B_p(\alpha, \overrightarrow{m})$ of length $2(p+1)$ over $\mathrm{GF}(4)$ can be obtained. Furthermore, it is obvious that equation $\alpha^2 + p = -1$ holds if and only if $\alpha = 0$, because p is an odd prime. And the parameter values except α that meet the conditions are listed in the following table when $p = 41 = 16 \times 2 + 9$ and $p = 73 = 16 \times 4 + 9$.*

Table 1: The parameters of P_p over GF(2) with $p = 41$

Serial number	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	Min-distance
1	0	0	0	1	1	1	0	1	1	10
2	0	0	1	1	0	0	1	1	1	10
3	0	1	0	0	1	1	1	0	1	10
4	0	1	1	0	1	1	0	0	1	10
5	1	0	0	0	1	0	1	1	1	14
6	1	0	0	1	0	1	1	1	0	14
7	1	0	0	1	1	0	1	0	1	12
8	1	0	1	0	1	0	0	1	1	12
9	1	0	1	1	1	0	0	0	1	14
10	1	1	1	0	1	0	1	0	0	12

Table 2: The parameters of P_p over GF(4) with $p = 41$

Serial number	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	Min-distance
1	1	1	1	1	ξ	1	1	1	ξ^2	14
2	1	1	ξ	1	ξ^2	0	0	ξ^2	ξ	12
3	1	1	0	0	0	1	0	1	1	14
4	ξ	1	ξ	1	0	0	1	ξ^2	ξ^2	14
5	0	ξ	0	ξ	ξ^2	ξ	ξ^2	ξ^2	0	14

Table 3: The parameters of B_p over GF(4) with $p = 41$

Serial number	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	Min-distance
1	1	1	1	ξ	1	0	ξ	ξ	ξ	12
2	1	1	ξ	1	ξ^2	ξ^2	ξ	ξ^2	ξ	8
3	1	1	ξ^2	1	0	ξ^2	ξ^2	ξ^2	1	8
4	ξ^2	1	0	0	ξ	ξ^2	ξ	1	0	12
5	0	1	ξ^2	ξ	0	ξ	0	ξ^2	1	12

Table 4: The parameters of P_p over GF(4) with $p = 73$

Serial number	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	Min-distance
1	1	1	ξ	ξ	0	1	ξ^2	ξ^2	0	12
2	1	1	ξ	0	ξ^2	1	ξ^2	0	ξ	12
3	0	ξ	ξ	ξ^2	0	ξ^2	ξ^2	ξ	0	6
4	1	1	0	ξ^2	ξ	1	0	ξ	ξ^2	12
5	1	ξ^2	ξ^2	ξ^2	ξ^2	ξ	ξ	ξ	ξ	12

Table 5: The parameters of B_p over GF(4) with $p = 73$

Serial number	m_0	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	Min-distance
1	1	1	ξ^2	ξ^2	ξ	1	ξ	ξ	ξ^2	8
2	0	1	ξ	0	ξ	1	ξ^2	0	ξ^2	12
3	0	1	0	ξ	ξ	1	0	ξ^2	ξ^2	12
4	0	ξ^2	1	0	ξ^2	ξ^2	1	0	ξ	12
5	0	ξ	ξ	1	0	ξ^2	ξ^2	1	0	12

The pure double circulant self-dual codes [82, 41, 14] codes and bordered double circulant self-dual codes self-dual [84, 42, 12] codes over GF(4) which have good property are listed, especially the values of parameters $m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8$. Besides, we get some other codes when $p = 73$.

5 Conclusion

In this paper, we construct double circulant self-dual codes by higher power residues, especially eighth power residues.

First of all, the relationship of the eighth power residue cyclotomic numbers is given. Suppose that there are eight matrices with nine parameters on the GF(q), and the expression for multiplying any two matrices is represented by the cyclotomic numbers. From the linear combination of eight circulant matrices, we can construct the circulant matrix R. Two kinds of codes are represented by R. One is pure circulant codes, and the other is bordered circulant codes. Combined with the necessary condition of self-dual code ($GG^{\mathrm{T}} = 0$), parameters can be determined to satisfy the condition of self-dual code, which renders the pure double circulant self-dual codes and bordered circulant self-dual codes can be obtained. By programming, the parameters that satisfy the conditions and the minimum hamming distance are given.

We exploit a new way to construct self-dual codes over GF(2) and GF(4) by prime p of the form $16f+9$, and some codes have good properties. Examples of such codes are binary self-dual [82, 41, 14] code, quaternary self-dual [82, 41, 14] code and quaternary self-dual [84, 42, 12] code.

Acknowledgments

The work is financially supported by National Natural Science Foundation of China (No. 61902429, No.11775306), Shandong Provincial Natural Science Foundation of China (No. ZR2017MA001, ZR2019MF070), Fundamental Research Funds for the Central Universities (No. 19CX02058A, No. 17CX02030A), the Open Research Fund from Shandong provincial Key Laboratory of Computer Networks, Grant No. SDKLCN-2017-03, Key Laboratory of Applied Mathematics of Fujian Province University (Putian University)(No.SX201702, No.SX201806), and International Cooperation Exchange Fund of China University of Petroleum (UPCIEF2019020).

References

[1] K. T. Arasu and T. A. Gulliver, "Self-dual codes over $\mathbb{F}_p$ and weighing matrices," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 2051-2055, 2001.

[2] E. R. Berlekamp, F. J. MacWilliams and N. J. A. Sloane, "Gleason's theorem on self-dual," *IEEE Transactions on Information Theory*, vol. 18, pp. 409-414, 1972.

[3] S. Buyuklieva, "On the binary self-dual codes with an automorphism of order 2," *Designs Codes & Cryptography*, vol. 12, no. 1, pp. 39-48, 1997.

[4] S. Bouyuklieva and I. Bouyukliev, "An algorithm for classification of binary self-dual codes," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3933-3940, 2012.

[5] J. H. Convay and V. Pless, "On the enumeration of self-dual codes," *Journal of Combinatorial Theory*, vol. 28, no. 1, pp. 26-53, 1980.

[6] J. H. Convay and J. A. Sloane, "A new upper bound on the minimal distance of self-dual codes," *IEEE Transactions on Information Theory*, vol. 36, no. 6, pp. 1319-1333, 1990.

[7] C. Ding, T. Helleseth, and W. Shan, "On the linear complexity of legendre sequences," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 1276–1278, 1998.

[8] S. T. Dougherty, J. L. Kim and P. Sole, "Double circulant codes from two class association schemes," *Advances in Mathematics of Communications*, vol. 1, no. 1, pp. 45-64, 2007.

[9] S. T. Dougherty, J. Gildea, A. Korban, A. Kaya, A. Tylyshchak and B. Yildiz, "Bordered constructions of self-dual codes from group rings and new extremal binary self-dual codes," *Finite Fields and Their Applications*, vol. 57, pp. 108-127, 2019.

[10] P. Gaborit, "Quadratic double circulant codes over fields," *Journal of Combinatorial Theory, Series A*, vol. 97, no. 1, pp. 85-107, 2002.

[11] M. Harada, M. Kiermaier, A. Wassermann, *et al.*, "New binary singly even self-dual codes," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1612-1617, 2010.

[12] T. Helleseth, "Double circulant quadratic residue codes," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 2154-2155, 2004.

[13] W. Huffman, "On the classification and enumeration of self-dual codes," *Finite Fields and Their Applications*, vol. 11, pp. 451-490, 2005.

[14] W. C. Huffman, R. A. Brualdi and V. S. Pless, *Handbook of Coding Theory*, 1998.

[15] K. Ireland and M. Rosen, "Gauss and jacobi sums," *Mathematical Gazette*, vol. 84, pp. 75-92, 1998.

[16] M. Karlin, "New binary coding results by circulants," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 81-92, 1969.

[17] A. Kaya, B. Yildiz and I. Siap, "New extremal binary self-dual codes from $\mathbb{F}_4 + u\mathbb{F}_4$-lifts of quadratic circulant codes over $\mathbb{F}_4$," *Finite Fields and Their Applications*, vol. 35, pp. 318-329, 2015.

[18] A. Kaya, B. Yildiz, "Various constructions for self-dual codes over rings and new binary self-dual codes," *Discrete Mathematics*, vol. 339, pp. 460-469, 2016.

[19] A. Kaya, "New extremal binary self-dual codes of lengths 64 and 66 from R_2-lifts," *Finite Fields and Their Applications*, vol. 46, pp. 271-279, 2017.

[20] A. Kaya, B. Yildiz and I. Siap, "New extremal binary self-dual codes of length 68 from quadratic residue codes over $\mathbb{F}_2+u\mathbb{F}_2+u^2\mathbb{F}_2$," *Finite Fields and Their Applications*, vol. 29, pp. 160-177, 2014.

[21] A. Kaya, B. Yildiz and I. Siap, "New extremal binary self-dual codes of length 68 from quadratic residue codes over $\mathbb{F}_2 + u\mathbb{F}_2 + u^2\mathbb{F}_2$[J]," *Finite Fields and Their Applications*, vol. 29, pp. 160-177, 2014.

[22] C. L. Mallows and N. J. A. Sloane, "An upper bound for self-dual codes," *Information & Control*, vol. 22, no. 2, pp. 188-200, 1973.

[23] R. Meng, T. Yan, "New constructions of binary interleaved sequences with low autocorrelation," *International Journal of Network Security*, vol. 19, no. 4, pp. 546–550, 2017.

[24] V. Pless and N. J. A. Sloane, "On the classification and enumeration of self-dual codes," *Journal of Combinatorial Theory*, vol. 18, no. 3, pp. 313-335, 1975.

[25] E. M. Rains, "Shadow bounds for self-dual codes," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 134-139, 1998.

[26] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423, 1948.

[27] S. Zhang, T. Yan, Y. Sun, L. Wang, "Linear complexity of two classes of binary interleaved sequences with low autocorrelation," *International Journal of Network Security*, 2019. (`http://ijns.jalaxy.com.tw/contents/ijns-v22-n6/ijns-2020-v22-n6-p834-0.pdf`)

[28] N. J. A. Sloane and J. G. Thompson, "Cyclic self-dual codes," *IEEE Transactions on Information Theory*, vol. 29, pp. 364-366, 1983.

[29] N. Yankov, M. Ivanova and M. H. Lee, "Self-dual codes with an automorphism of order 7 and s-extremal codes of length 68," *Finite Fields and Their Applications*, vol. 51, pp. 17-30, 2018.

[30] T. Zhang and G. Ge, "Fourth power residue double circulant self-dual codes," *IEEE Transactions on Information Theory*, vol. 61, no. 8, pp. 4243-4252, 2015.

Biography

Changsong Jiang was born in 1997 in Sichuan Province of China. He was graduated from China University of Petroleum in 2019. He is currently studying for a postgraduate degree at University of Electronic Science and Technology of China. Email: jiangchso@163.com

Yuhua Sun was born in 1979. She was graduated from Shandong Normal University, China, in 2001. In 2004, she received the M.S. degree in mathematics from the Tongji University, Shanghai and a Ph.D. in Cryptography from the Xidian University. She is currently a lecturer of China University of Petroleum. Her research interests include cryptography, coding and information theory. Email: sunyuha_1@163.com

Xueting Liang was born in 1997 in Anhui Province of China. She is studying at China University of Petroleum. Email:13399613079@163.com

Identity-based Public Key Cryptographic Primitive with Delegated Equality Test Against Insider Attack in Cloud Computing

Seth Alornyo[1,2], Acheampong Edward Mensah[1], and Abraham Opanfo Abbam[1]
(Corresponding author: Seth Alornyo)
School of Information and Software Engineering, University of Electronic Science and Technology of China[1]
4 E 2nd Section, 1st Ring Rd, Jianshe Road, Chenghua, Chengdu, Sichuan, China
Computer Science Department, Koforidua Technical University, Koforidua-Ghana[2]
(Email: sabigseth@outlook.com)
(Received Mar. 3, 2019; Revised and Accepted Oct. 3, 2019; First Online Jan. 23, 2020)

Abstract

The notion of attacks perpetuated by an insider (cloud server) is paramount in this era of cloud computing and data analytics. When a cloud server is delegated with certain responsibilities, it is possible for a cloud server to peddle with users' encrypted data for profit gains. The cloud server takes advantage of it's authorization to launch what we referred to as insider attack. We put forward a new improved scheme on identity-based public key cryptographic primitive which integrate delegated equality test to resist insider attack in cloud computing. Our scheme resist the insider attack perpetuated by the cloud server (insider). We refer to our new scheme as identity-based public key cryptographic primitive with delegated equality test against insider attack in cloud computing (IB-PKC-DETIA). We construct our scheme using a witness based cryptographic primitive with an added pairing operation. Our scheme achieves weak indistinquishable identity chosen ciphertext (W-IND-ID-CCA) security using the random oracle model.

Keywords: Identity Based Encryption; Insider Attack; Witness Based Encryption

1 Introduction

The concept of searchable public key encryption which integrate keyword search (PEKS) was unveiled by [2]. Inspite of this, there has been several works on this cryptographic primitives where a search on ciphertext allows a third party to search over an encrypted data without revealing any information about the ciphertext. A Public key cryptographic primitive with equality test was unveiled by [18] and was used to manage encrypted data for clients. Recently, [11] proposed identity based cryptosystem with integrated equality test (IBE-ET) in cloud computing and it enables the cloud server to verify whether two ciphertext from user A and user B are encryption of the same message. .

However, there has been a recent attack perpetuated by an adversary who is able to launch what is referred to as the insider attack [15]. In this era of cloud computing, equality test function are outsourced to a cloud server to examine whether two ciphertext are encryptions with same message [4]. Such a delegated responsibility to the cloud server gives it the leverage to launch the insider attack on users' ciphertext. This attack when successful enables the cloud server peddle with encrypted data for economic gains. If the cloud server has legitimate access to all users ciphertext and can test their equality, then the cloud server (insider) should be resisted from peddling with users' ciphertext. Recent schemes on insider attack has not been able to fully solve this problem.

Recent works of insider attack by [15] and a security analysis and modification by [19] enables the user to generate a token tok_{ID} to prevent the tester from launching the insider attack. Therefore, their scheme is susceptible to the insider attack because the cloud server was not delegated to perform equality test. When the cloud server is delegated to perform equality test on users ciphertext, it could guess the token tok_{ID} to launch the insider attack. The insider attack resistance scheme proposed by [15] and a security analysis and modification by [19] enable the tester after receiving the secret trapdoor to successfully guess the token tok_{ID} and launch the attack as follows:

1) The cloud server (insider) receives a valid trapdoor T_d and tries to find out the message m and the token from T_d.
2) The cloud server (insider) computes IB-PKC-DETIA ciphertext C of a guessed message $m^{'}$ and a guessed token $tok^{'}_{ID}$.
3) The cloud server (insider) checks whether $Test(C, T_d, tok_{ID})= 1$. The equation holds if a

guess of the message $m^{'}$ and a token $tok_{ID}^{'}$ is successful to the adversary (cloud server). Otherwise, go back to Step 2.

Therefore, if the guess of a message and a token are successful as indicated above, then the cloud server (insider) could launch the insider attack because of the delegated responsibility. Other works of IBE-ET [11] and a security modification by [17] assumed that it's possibe to resist the insider attack. On the contrary, when the cloud server is delegated to conduct equality test, it is possible the adversary can launch the insider attack. Therefore, we propose a new improved scheme to resist insider attack perpetuated by adversary (cloud server) delegated to undertake equality or equivalence test on ciphertext.

1.1 Related Work

Boneh *et al.* [2] first unveiled the primitive of PEKS and was later examined by [11] on their work on off-line keyword guessing attack on recent keyword search schemes. Their work showed that PEKS scheme was vulnerable to the insider attack. A related works on a delegated tester was later unveiled by [14] whereby only the designated tester (server) could perform equality test on the ciphertext. Their scheme concentrated on security of the trapdoor in PEKS and a resistant to insider attack. Chen *et al.* [9,17] also proposed a new general framework for secure public key cryptosystem with keyword search and a dual-server public key primitive with keyword search for secure cloud storage to resist the insider attack so far as there was no collusion by two servers [9]. However, keyword guessing attacks against the insider has being a challenging problem in PEKS until recently, [12] proposed the notion of witness-based searchable cryptographic primitive to resist insider attack in PEKS.

A special type of searchable encryption was unveiled in [18] for a general equality test. However, [11] introduced identity based cryptosystem with equality test (IBE-ET) in cloud computing which integrated the identity-based primitive into public key cryptosystem with equality test [2], it gains the advantages of equality test in [18]. In their construction, search functions were delegated to the service provider.

Existing works on insider attacks mainly focused on PEKS schemes in [8,19] whiles few works on PKEETs extensions [7, 11, 13, 15, 18] and IBE-ET applications in [19] were not resistant to insider attack. To solve the problem of insider attack in IBE-ET, ID-based primitive with equality test against the insider attack was recently put forward by Wu *et al.* [17], the scheme claimed their scheme achieves confidentiality in IBE-ET but the work of Lee *et al.* [9] refuted their claim of weak indistinguishability of IBE-ET. Lee *et al.* [9] modified their security analysis claims to achieve the weak indistinguishability as unveiled in [17]. While in [17], their scheme ensured that the designated users' token tok_{ID} were' changed per a corresponding identity, but in [9] scheme, they ensured that the token was fixed for all group users. Therefore, a fixed token tok_{ID} could successfully enable the cloud server guess a new token $tok_{ID}^{'}$ to launch the insider attack. When a cloud server (insider) is delegated to perform equality test, it is possible for the cloud server to launch the insider attack because a guess of a token is possible. To the best of our knowlege, their scheme cannot resist the insider attack as explained above. A scheme to resist the insider attack with delegated equality test in IBE-ET with the cloud server (insider) authorized to perform equality test is still problem to the research community.

1.2 Our Contribution

Wu *et al.* [17] unveiled a variant to IBE-ET scheme. However, their scheme allowed anyone to perform equality test between two ciphertext hence lack authorization for equality test. The security analysis and modification in [9] did not authorize a third party (cloud server) by generating a trapdoor function for the cloud server to perform equality test. It is not clear whether a computed trapdoor to the cloud server could resist the insider attack as claimed in their security analysis and modification scheme.

To address this problem, we added a pairing operation to the witness cryptographic primitive in [6] to resist insider attack in IBE-ET. Witness based encryption ensure that given a witness relation $R(W, X)$ of an NP language L, an encryption of (m, w) can be tested by a generated trapdoor (m', x). The tester checks if $m' = m$. However, it is difficult to compute w from x under a defined witness relation.

Our scheme achieves Weak-IND-ID-CCA (W-IND-ID-CCA) and a resistant to insider attack. Our scheme achieves a stronger notion of IND-ID-CCA security for IBE-ET using the random oracle model.

1.3 Organization

The rest of the paper is organized as follows. In Section 2, our scheme provide some preliminaries for our construction. In Section 3, our scheme formulate the notion of IB-PKC-DETIA. In Section 4, construction of IB-PKC-DETIA and prove its security in Section 5. In Section 6, we compare our work with other related works. In Section 7, we conclude our paper.

2 Preliminaries

Definition 1. *Billinear map: Let* $\mathbf{G}$ *and* $\mathbf{G}_T$ *be two multiplicative cyclic groups of prime order p. Suppose that* $\mathbf{g}$ *is a generator of* $\mathbf{G}$. *A bilinear map* $\mathbf{e} : \mathbf{G} \times \mathbf{G} \rightarrow \mathbf{G}_T$ *satisfies the following properties:*

1) Bilinearity: For any $\mathbf{g} \in \mathbf{G}$, a and b $\in \mathbf{Z}_p$, $\mathbf{e}(\mathbf{g}^a, \mathbf{g}^b) = \mathbf{e}(\mathbf{g}, \mathbf{g})^{ab}$.

2) Non-degenerate: $\mathbf{e}(\mathbf{g}, \mathbf{g}) \neq 1$.

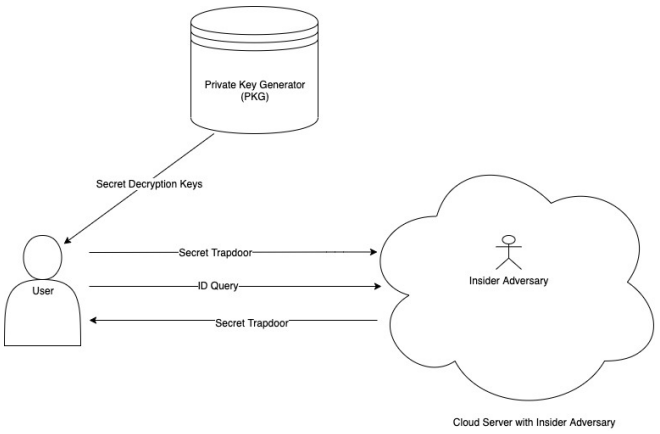


Figure 1: System model for IB-PKC-DETIA

3) Computable: There is an efficient algorithm to compute $\mathbf{e}(\mathbf{g},\mathbf{g})$ for any $\mathbf{g} \in \mathbf{G}$.

Definition 2. *Bilinear Diffie-Hellman (BDH) problem: Let* $\mathbf{G}$,$\mathbf{G_T}$ *be two groups of prime order p. Let* $\mathbf{e} : \mathbf{G} \times \mathbf{G} \rightarrow \mathbf{G_T}$ *be an admissible bilinear map and let* $\mathbf{g}$ *be a generator of* $\mathbf{G}$. *The BDH problem in* $\langle \mathbf{p}, \mathbf{G}, \mathbf{G_T}, \mathbf{e} \rangle$ *is as follows: Given* $\langle \mathbf{g}, \mathbf{g}^a, \mathbf{g}^b, \mathbf{g}^c \rangle$, *for random a,b,c* $\in \mathbf{Z}_p^*$, *for any randomized algorithm* $\mathbf{A}$ *computes value* $\mathbf{e}(\mathbf{g},\mathbf{g})^{abc} \in \mathbf{G}_T$ *with advantage:*

$ADV_{\mathbf{A}}^{BDH} Pr[\mathbf{A}(\mathbf{g}, \mathbf{g}^a, \mathbf{g}^b, \mathbf{g}^c) = \mathbf{e}(\mathbf{g},\mathbf{g})^{abc}]$. The BDH assumption holds if for any polynomial-time algorithm $\mathbf{A}$, it's advantage $Adv_{\mathbf{A}}^{BDH}$ is negligible.

Definition 3. *Witness Relation: Given a witness relation* $R(W, X)$ *on an NP language L [3], a randomly chosen* $w \in W$ *generates an instance* $x \in X$ *defined over the relation R. For any polynomial algorithm:* $A_{IB-PKC-DETIA}, Pr[A_{IB-PKC-DETIA}(k, w) = x] = 1$, *and* $A_{IB-PKC-DETIA}, Pr[A_{IB-PKC-DETIA}(k, x) = w] < \varepsilon(k)$, *where k is a security parameter and* ε *a negligible functionon on k.*

Our model has four roles which includes: users, PKG, cloud server and with adversary (see Figure 1). Users stores their encrypted sensitive data in the cloud. The cloud server with adversary is resisted from peddling with users encrypted sensitive data for economic gains. In this section, we give formal definitions of our scheme. We employ a witness based cryptographic primitive to resist the insider attack in IBE-ET. Our scheme achieves weak chosen ciphertext security (i.e. W-IND-ID-CCA) under the defined security model.

In identity-based public key cryptographic primitive with delegated equality test against insider attack scheme, we specify seven algorithms: $Setup, Extract, WBInstGen,$ $Trapdoor, WBEncrypt, WBDecrypt, Test$, where $\mathbf{M}$ and $\mathbf{C}$ are its plaintext space and ciphertext space, respectively:

1) **Setup**: It takes as input security parameter k and returns the public key K and msk.

2) **Extract**: It takes as imput msk, an arbitrary $ID \in \{0,1\}^*$ and returns a decryption key dk for that identity.

3) **WBInstGen**: It takes as input the security parameter k, an arbitrary $ID \in \{0,1\}^*$ and returns a private witness key $w \in W$ for that identity w_{ID}, where $WInsGen(w) = x$ and $x \in X$ where (w, x) satisfies the witness relation R.

4) **Trapdoor**: It takes as input decryption key dk, an arbitrary $ID \in \{0,1\}^*$, an instance $x \in X$ and returns a trapdoor td for that identity.

5) **WBEncrypt**: It takes as input an identity $ID \in \{0,1\}^*$, a plaintext $m \in M$ with a random chosen witness $w \in W$ and outputs a ciphertext $C = (x, c)$ where $x \in X$ from a generated witness $WInsGen(w) = x$, and (w,x) satisfies the witness relation R.

6) **WBDecrypt**: The algorithm takes as input the ciphertext $c \in C$, a private decrption key dk and a witness $w \in W$ and returns a plaintext $m \in M$, if and only if C is a valid ciphertext with the ID and a witness $w \in W$.

7) **Test**: It takes as input a ciphertext $C_A \in C$ of a receiver with ID_A, a trapdoor td_A for the receiver with ID_A, a ciphertext $C_B \in C$ of a receiver with ID_B and trapdoor td_B for the receiver with ID_B, and returns 1 if C_A and C_B contains the same message. Otherwise return $\perp$.

3 Security Model

Definition 4. *(Weak-IND-ID-CCA). We let* $\sqcup = (Setup, Extract, WBInstGen, Trapdoor, WBEncrypt, WBDecrypt, Test)$ *be the same scheme and a polynomial time algorithm A.*

1) **Setup**: The challenger runs the security parameter on input k and derives K and randomly takes a witness $w \in W$ and generates an instance $x \in X$ of a witness relation $R(W, X)$ defined on an NP language L. It gives the relation R to the adversary.

2) **Phase 1**: The adversary issues query N_1, N_2, $\cdots$, N_m. Each query is of the form:

   - Query (ID_i): The challenger run $H(.)$ to generate dk_i corresponding to the public identity (ID_i). It sends dk_i to A.

   - Trapdoor (ID_i): The challenger runs the private decryption on $WInsGen$ using a randomly chosen witness $w \in W$ of the relation R(W,X)

on an NP language L. The algorithm generates an instance x and compute a trapdoor td_i using dk_i via trapdoor algorithm. Finally, it sends td_i to A.

- Decryption queries (ID_i, C_i, w): The challenger runs the decryption algorithm to decrypt the ciphertext C_i by running the extract algorithm to obtain dk_i corresponding to the public key (ID_i). Finally, it sends the plaintext M_i to A.

3) **Challenge**: After Phase 1 is over, A submits two equal-length message (m_0, m_1) and ID^* to be challenged by the challenger. However, both (m_0, m_1) are not issued in the encryption query and ID^* is also not in the extract query in Phase 1. The challenger randomly picks $b \in \{0,1\}$ and respond with $C^* \leftarrow Enc(M_b, ID^*, w^*)$. The algorithm generates a challenge trapdoor $td^* = (ID^*, x^*)$ by runing the trapdoor $td^* \leftarrow td(dk, M_b, x^*)$ algorithm and returns td^* to A.

4) **Phase 2**: The adversary issues query N_1, N_2, $\cdots$, N_m. Each query is of the form:

- Query (ID_i). The challenger responds as in Phase 1, since $ID_i \neq ID^*$.
- Trapdoor query (ID_i). Where $x \neq x^*$. The challenger respond in the same way as in Phase 1.
- Decryption Query(ID_i, C_i). Where $(ID_i, C_i) \neq (ID^*, C^*)$, the challenger respond in the same way as in Phase 1.

5) Output: A submits a guess $b^{'}$ on b. If $b^{'} = b$, we say A wins the game.

We define A's advantage on breaking the scheme as $Adv_{IB-PKC-DETIA}(k) = Pr[b^{'} = b] - \frac{1}{2}$ is negligible.

In the Weak-IND-ID-CCA (W-IND-ID-CCA) model, the adversary has access to ciphertexts but cannot compute the witness $w \in W$ from $x \in X$ of a relation $R(W, X)$ over NP language L.

4 Construction

Our scheme aims to resist an attack continuum perpetuated by a cloud server delegated to perform equality test. The cloud server (insider) is considered as an adversary A who is authorized only to perform equlaity test but should not be able to peddle with user's ciphertext. However, only the authorized cloud server could perform equality test.

Definition 5. *A witness relation $R(X, W)$ on an NP language L consist of the following polynomial-time algorithm. Given a randomly chosen $w \in W$ over a witness relation $R(w, x)$=1 defined on an NP language L if for any polynomial algorithm: $A_{IB-PKC-DETIA}, Pr[A_{IB-PKC-DETIA}(k, w) = x] = 1$, and $A_{IB-PKC-DETIA}$, $Pr[A_{IB-PKC-DETIA}(k, x) = w] < \varepsilon(k)$, where k is a security parameter and ε a negligible functionon on k.*

1) **Setup**: The system takes a security parameter k and returns the public parameter K and master secret key msk.

- The algorithm generates the pairing parameters $\mathbf{G}$ and $\mathbf{G}_T$ of prime order p and an admissible bilinear map $\mathbf{e} : \mathbf{G} \times \mathbf{G} \rightarrow \mathbf{G}_T$. Choose a random generator $\mathbf{g} \in G$.
- The system choose cryptographic hash functions: $H : \{0,1\}^* \rightarrow \mathbf{G}$, $H_1 : \mathbf{G}_T \rightarrow \mathbf{G}$, $H_2 : R \times \mathbf{G}_T \rightarrow \{0,1\}^{\tau_1+\tau_2}$, where τ_1 and τ_2 are security parameters. The elements of $\mathbf{G}$ are represented in τ_1 bits and elements of $x \in X$ of a witness relation R are represented in τ_2 bits.
- The algorithm randomly picks a witness $w \in W$ and generate an instance $x \in X$ on NP language L of a relation $R(W, X)$ and set $\mathbf{g}_1 = \mathbf{g}^w$ and $\mathbf{g}_2 = \mathbf{g}^x$. The ciphertext space $C \in (\mathbf{G}^* \times \{0.1\})^{\tau_1+\tau_2}$. The message space is $M \in \mathbf{G}^*$. It publishes the public parameter $K = \{\mathbf{p}, \mathbf{G}, \mathbf{G}_T, \mathbf{R}, \mathbf{e}, \mathbf{g}, \mathbf{g}_1, \mathbf{g}_2, H, H_1, H_2\}$.

2) **WBInstGen**: The algorithm takes as input the secret parameter k and arbitrary $ID \in \{0,1\}^*$ and compute $h_{ID} = H(ID) \in \mathbf{G}$ and randomly choose a witness $w \in W$ and generates a corresponding instance $x \in X$ on a witness relation $R(W, X)$.

3) **Extract**: Given a string $ID \in \{0,1\}^*$, the algorithm computes $h_{ID} = H(ID) \in \mathbf{G}$ and set the private decryption key $dk_{ID} = (h_{ID}^w, h_{ID}^x)$ where (w, x) is the master key corresponding to the relation R.

4) **Trapdoor**: On input a string $ID \in \{0,1\}^*$, the algorithm compute $h_{ID} = H(ID) \in \mathbf{G}$ and set $td_{ID} = (h_{ID}^x)$ where x is an instance of the relation R.

5) **WBEncypt**: The algorithm on input the public parameter K, ID, it computes $h_{ID} = H(ID) \in \mathbf{G}$ and encrypt $M \in \mathbf{G}$ by choosing two random numbers $(r_1, r_2) \in Z_q^*$ with a randomly chosen witness and instance (w, x). The algorithm set the ciphertext $C = (C_1, C_2, C_3, C_4)$ as:

$$\begin{aligned} C_1 &= M^x.H_2(e(h_{ID}, g_2)^{r_1}), \\ C_2 &= g^{r_1} \\ C_3 &= g^{r_2}, \\ C_4 &= (M \parallel w) \oplus H_2(C_1 \parallel C_2 \parallel C_3 \parallel e(h_{ID}, g_1)^{r_2}). \end{aligned}$$

6) **Decrypt**: To decrypt, the algorithm requires an input the ciphertext C, private decryption key $dk_{ID} = (h^w_{ID}, h^x_{ID})$ where $C = (C_1, C_2, C_3, C_4)$ corresponding to the ciphertext encrypted with ID. It computes:

$$(M^{'} \parallel w^{'}) = C_4 \oplus H_2(C_1 \parallel C_2 \parallel C_3 \parallel e(C_3, h^w_{ID})),$$

where $w \in W$ of the witness relation $R(W, X)$ of the NP language L. The algorithm checks whether $C_1 = (M^{'} \mid x^{'})$ and $C_2 = g^{r_1^{'}}$. Hence, if both holds, return $M^{'}$. Otherwise $\perp$.

7) **Test**: The algorithm on input a ciphertext C_A, a trapdoor td_A and a given sender's ciphertext C_B. The algorithm test whether $M^x_A = M^x_B$ by computing:

$$T_A = \frac{C_{1_A}}{H_2(e(C_{1_A}, td_{ID_A}))}$$
$$T_B = \frac{C_{1_B}}{H_2(e(C_{1_B}, td_{ID_B}))}$$

If the above equation holds, the algorithm outputs 1. Otherwise 0. Thus:

$$e(T_B, C_{2_A}) = e(T_A, C_{2_B}).$$

Remark 1. With the work in [17], the token generated was changed per identity in their construction whiles a security analysis and modification in [9] had a fixed token for all group users. We note that since the token was fixed in their construction, the insider attack is paramount in their scheme. A randomly chosen witness w under the relation $R(W, X)$ is considered to be secure and should avoid a reuse. A reuse of a randomly chosen witness will compromise the security of our scheme hence should be discarded immediately the decryption process is completed.

Correctness:
Let $C_A \leftarrow WBE(M_A, ID_A, g_{2_A}, x_A)$ and $C_B \leftarrow WBE(M_B, ID_B, g_{2_B}, x_B)$ generated by user A and user B respectively. Then:

$$C_A = (C_{A_1}, C_{A_2}, C_{A_3}, C_{A_4}).$$

Test algorithm computes results as:

$C_A = \frac{C_{1_A}}{H_2(e(C_{2_A}, td_{ID_A}))}$, $C_B = \frac{C_{1_B}}{H_2(e(C_{2_B}, td_{ID_B}))}$.

$T_A = \frac{M^{x_A}_A . H_2(e(C_{2_A}, td_{ID_A}))}{H_2(e(C_{2_A}, td_{ID_A}))}$, $T_B = \frac{M^{x_B}_B . H_2(e(C_{2_B}, td_{ID_B}))}{H_2(e(C_{2_B}, td_{ID_B}))}$.

$T_A = \frac{M^{x_A}_A . H_2(e(g_{1_A}^{r_{1_A}}, h^{x_A}_{ID_A}))}{H_2(e(g_{1_A}^{r_{1_A}}, h^{x_A}_{ID_A})}$, $T_B = \frac{M^{x_B}_B . H_2(e(g_{1_B}^{r_{1_B}}, h^{x_B}_{ID_B}))}{H_2(e(g_{1_B}^{r_{1_B}}, h^{x_B}_{ID_B})}$.

$T_A = M^{x_A}_A$ and $T_B = M^{x_B}_B$.

Therefore, $M^{x_A}_A = M^{x_B}_B$.

The algorithm output 1 if the following equation holds. Otherwise 0. Hence:

$$e(C_{2_A}, T_B) = e(C_{2_A}, T_B)$$

$$e(C_{2_A}, T_B) = e(g^{r_{1_A}}, M^{x_B}_B) = e(g, M_B)^{r_{1_A} x_B}$$

$$e(C_{2_B}, T_A) = e(g^{r_{1_B}}, M^{x_A}_A) = e(g, M_A)^{r_{1_B} x_A}$$

Remark 1. Given a witness $w \in W$, the user should be able to compute x to recover M on a witness relation R. However, given the instance x of a witness relation R, it is difficult to recover its corresponding witness w. Therefore the cloud server can only perform equality test but cannot generate a new ciphertext.

If $M_A = M_B$, it implies that $e(C_{2_A}, T_B) = e(C_{2_B}, T_A)$. Given the witness relation $R(W, X)$ defined over an NP language L. It means that: $A_{IB-PKC-DETIA}$, $Pr[A_{IB-PKC-DETIA}(k, w) = x] = 1$, and $A_{IB-PKC-DETIA}$, $Pr[A_{IB-PKC-DETIA}(k, w) = w] < \varepsilon(k)$, where k is a security parameter and ε is a negligible function on k.

5 Security Analysis

We define W-IND-CCA security for IBE-ET via the following game similar in [18].

A probabilistic polynomial time (PPT) adversary A achieves the advantage ε on breaking $\sqcap = (Setup, Extract, WBInstGen, Trapdoor, WBEncrypt, WBDecrypt, Test)$. Given BDH instance, a PPT adversary B takes advantage of A to solve the BDH problem with a probability of $\varepsilon^{'}$.

Suppose B holds a tuple (g,U,V,R) where $a = log_g U$, $b = log_g V$ and $c = log_g R$ are unknown. Given the generator g of G, B is supposed to output $e(g, g)^{abc} \in G_T$. The game between B and A runs as follows:

Setup. B sets $g_1 = g^{a.r_1} = U^{r_1}$, where $r_1 \leftarrow Z^*_q$ and sets trapdoor $td = x \leftarrow X$ from a witness relation $R(W, X)$. B gives g_1 to A.

Phase 1.

1) H Query: A query the random oracle H. A queries ID_i to obtain h_{ID}. B responds with h_{ID}. If ID_i has been in the H table$(ID_i, h^w_{ID_i}, coin)$. Otherwise, for each ID_i, B responds as follows:
   - B tosses a coin with $Pr[coin_i = 0] = \delta$. If $coin_i = 1$, responds to A with $h_{ID} = g^{w_i}$, $w_i \leftarrow W$. Otherwise, B sets $h_{ID} = g^{w_i y} = V^{w_i}$.
   - B responds with h_{ID_i}, then adds $(ID_i, h_{ID}, w_i, coin)$ in the H table which is initially empty.
2) Extract Query: A queries private key of ID_i. B responds as follows:

- B obtains $H(ID_i) = h_{ID}$ in the H table. If $coin_i = 0$, B responds $\perp$ and terminates the game.
- Otherwise, B responds with $dk_{ID_i} = g_1^{w_i} = U^{r_1 \cdot w_i}$, where (w_i, h_{ID_i}) is in the H table.
- B sends dk_{ID_i} to A, then stores (dk_{ID}, ID_i) in the private key list which is initially empty.

3) H_2 Query: A queries $D_i \in R \times G_T \rightarrow \{0,1\}^{\tau_1+\tau_2}$. B responds with $S_i \in H_2(D_i)$ in the H_2 table. Otherwise, for every D_i, B selects a random string $S_i = \{0,1\}^{\tau_1+\tau_2}$ as $H_2(D)$. B responds A with $H_2(D_i)$ and adds (D_i, S_i) in the H_2 table which is initially empty.

4) Trapdoor Query: B runs the private decryption key queries on (ID_i) to obtain $dk_{ID} = (g_1^{w_i}, g_1^{x_i})$ and responds A with $td_{ID_i} = g_2^{x_i}$. td_{ID_i} is the first element of the decryption key.

5) Encryption Query: A queries M_i encrypted with ID_i. B responds as follows:

- B searches the H table and obtain h_{ID} and computes $h_{ID_i} = (h_{ID_i}^{w_i}, h_{ID_i}^{x_i})$ where (w, x) are randomly chosen from the witness relation $R(W, X)$.
- A selects $r_{1_i}, r_{2_i} \leftarrow Z_q^*$ and computes:

$$\begin{aligned} C_{1_i} &= M^{x_i}.H_2(e(h_{ID_i}, g_2)^{r_1}) \\ C_{2_i} &= g^{r_{1_i}} \\ C_{3_i} &= g^{r_{2_i}} \\ D_i &= (C_{1_i}||C_{2_i}||C_{3_i}||e(h_{ID_i}, g_1)^{r_{2_i}}). \end{aligned}$$

- B queries O_{H_2} to obtain $S_i = H_2(D_i)$.
- B computes $C_{4_i} = (M_i||w_i) \oplus S_i$.

Then B responds with $C_i = (C_{1_i}, C_{2_i}, C_{3_i}, C_{4_i})$.

6) Decryption Query: A queries C_i to be decrypted in ID_i. B responds as follows:

- B searches the H table to obtain h_{ID_i}. IF $coin_i = 1$, obtain dk_{ID_i} of ID_i in the private key list to decrypt C_i. Then B computes the bilinear map with dk_{ID_i} as:

$$e(C_{3_i}, h_{ID_i}) = e(g^{r_{2_i}}, g^{w_i}) = e(g, g)^{r_{2_i} w_i}.$$

- B computes $D_i = (C_{1_i}||C_{2_i}||C_{3_i}||e(h_{ID_i}, g_1)^{r_{2_i}})$ and obtains S_i in the H_2 table. B obtains M_i and r_{2_i} by $C_{4_i} \oplus S_i$.

Finally, B computes $C_{1_i}^*, C_{1_i}^*$ with M_i and r_{2_i} decrypted from C_i. If it is a valid ciphertext that $C_{1_i}^* = C_{1_i}$ and $C_{2_i}^* = C_{2_i}$. B responds with M_i. Otherwise $\perp$.

Challenge: Once Phase 1 is over, A outputs two messages m_0, m_1 of equal length and ID^* to be challenged, where both m_0, m_1 are not issued in encryption Query and ID^* is not queried in Extract Query in Phase 1. B responds as follows:

- B searches the H table, if $coin^* = 1$, B responds with $\perp$ and terminates the game, since $h_{ID}^* = g^{w^*}$. It is observed that for a given witness relation $R(W, X)$, it is difficult to compute the corresponding witness w for a given instance x.
- Otherwise, B randomly selects $b \in \{0,1\}$, however, $dk_{ID} = (h_{ID}^{w^*}, h_{ID}^{x^*})$ and calculates:

$$\begin{aligned} C_1^* &= M_b^x.H_2(e(h_{ID}^*, g_2)^{r_1^*}) \\ C_2^* &= g^{r_1^*} \\ C_3^* &= g^{r_2^*} \\ C_4^* &= (M_b||w^*) \oplus S^*, \end{aligned}$$

where $w \in W$ of a witness relation $R(W, X)$, $S^* = H_2(D^*)$ and $D^* = (C_1^*||C_2^*||C_3^*||e(C_3, h_{ID}^w))$, where h_{ID}^w is unknown and B want A to compute. $C^* = (C_1^*||C_2^*||C_3^*||C_4^*$ is a valid ciphertext for M_b.

- B responds A with C^*.

Phase 2:

1) H Query: A queries as in Phase 1.
2) Extract Query: A queries as in Phase 1, but $ID_i \neq ID^*$.
3) H_2: A issues the query as in Phase 1.
4) Trapdoor Query: A queries as in Phase 1, but responds to trapdoor queries the same as in Phase 1. However, the adversary given the witness relation instance $x \in X$, the adversary cannot compute the corresponding witness $w \in W$ of the relation $R(W, X)$ to generate a new ciphertext C^*.
5) Encryption Query: The message $M_i \in \{m_0, m_1\}$, A queries as in Phase 1.
6) Decryption Query: A queries as in Phase 1, except that ciphertext $(C_i, ID_i) \neq (C^*, ID^*)$.

Result: Given a witness relation R, a randomly chosen witness $w \in W$ generates an instance $x \in X$. Given the instance x to compute $td_{ID} = h_{ID}^x$, it is dificult to compute the corresponding witness $w \in W$.

However, A guess $b^{'}$ on b. If $b^{'} \neq b$ and $w^{'} \neq w$, B responds with failure and terminate the game. If $b^{'} = b$ and $w^{'} = w$, then B gets the results of the BDH tuple by guessing the inputs of H_2 query. However, this is not possible under the define witness relation R on NP language L. B aborts the game because, $|Pr[b^{'} = b] - \frac{1}{2}| \geq \frac{\varepsilon}{e(q_{td}+q_e+q_d+1)}$.

Trapdoor Security: We further provide a trapdoor security (TD) experiment to our scheme:

$$Exp_{IB-PKC-DETIA,A}^{W-IND-ID-CCA}(k).$$

Table 1: Efficiency comparisons of PKEETs variant

PKEETs	IA	Enc	Dec	Test	Del	Security
[18]	N	3Exp	3Exp	2P	N/A	OW-CCA
[16]	N	4Exp	2Exp	4P	3Exp	OW/IND-CCA
[15]	N	5Exp	2Exp	4Exp	N/A	OW/IND-CCA
[13]	N	1P+5Exp	1P+4Exp	4P+2Exp	3Exp	OW/IND-CCA
[11]	N	6Exp	2P+2Exp	4P	2Exp	OW/IND-CCA
[17]	Y	1P+3Exp	1P+2Exp	2P	N/A	W-IND-ID-CCA
Ours	Y	2P+3Exp	1P+1Exp	4P	2Exp	W-IND-ID-CCA

Remark: In this table, $''Exp''$ refers to the exponent computation, $''P''$ refers to the pairing computation, $''IA''$ refers to insider attack, $''Y''$ refers to 'Yes' as a supportive remark, $''N''$ refers to 'No' as not supportive and $''Del''$ refers to the delegation, W-IND-ID-CCA refers to weak indistinguishable chosen ciphertext attack against identity, OW-ID-CCA refers to one-way chosen ciphertext attack against ientity and IND-ID-CPA refers to indistinguishable chosen plaintext attack against identity.

Given a security parameter k, a witness $w \in W$ and A adversary against trapdoor security (TD). The experiment between the adversary A and the challenger is as follows:

WInsGen Generation Phase: The challenger runs the WInsGen algorithm with a random witness parameter $w \in W$. It generates a corresponding instance $x \in X$. The adversary A computes an instance $x^*(x^* \notin X)$ from a randomly chosen witness w^*. Finally, it gives $td^*_{ID} = h^{x^*}_{ID}$ to A.

Phase 1: A adaptively ask the challenger for the following trapdoor oracle:

1) Trapdoor oracle: On input a message M and instance $x \in X$ where $x \neq x^*$ submitted by A. It output the trapdoor $td_{ID} = h^x_{ID}$ by running the trapdoor algorithm.

2) Challenge Phase: A submit two messages (m_0, m_1) with equal length. The challenger picks $b \leftarrow \{0,1\}$ and generate challenge trapdoor $td^*_{ID} = h^{x^*}_{ID}$ corresponding to the challenge ciphertext $M_b||x \oplus H_2(e(h^x_{ID}, C_2))$ by running the WInsGen algorithm and returns td^*_{ID} to A.

Phase 2:

1) TD_{ID} Query: A continue to ask the oracle for trapdoor queries. Oracle responds as in Phase 1.

2) Output: A output its quess $b^{'}$. The adversary win the game if $b^{'} = b$, which shows that the output of experiment is 1 and 0 otherwise. Adversary A advantage in the above experiment is defined as:

$$\begin{aligned} & Adv^{W-IND-TD}_{IB-PKC-DETIA}(w) \\ = \; & |Pr[Exp^{W-IND-TD}_{IB-PKC-DETIA}] - \frac{1}{2}. \end{aligned}$$

6 Comparison

In this section, we made a comparison (Table 1) on the efficiency of algorithms adopted in our scheme with other PKEET variants. Other PKEET variants (Table 1) achieved a one-way chosen ciphertext attack (OW-CCA), one-way indistinquishable chosen ciphertext attack (OW/IND-CCA) and a weak indistinquishable identity chosen ciphertext attack (W-IND-ID-CCA) security. The extended PKEET schemes cost three to four steps to conduct the equality test including analyzing trapdoor and inverse-computing trapdoor.

The above comparison shows that our scheme can resist insider attack, whereas others do not have such ability except in [17]. Even though Wu *et al.*'s scheme resist insider attack, it does not provide delegation to the cloud server (insider) to perform equality test. It is possible for Wu *et al.*'s scheme to fail the insider attack resistance when the cloud server is delegated to perform equality test. However, our scheme ensures that equality test is delegated to the cloud server and the cloudserver is resisted from launching the insider attack.

The experiment results are shown in Figure 2. This experiment is executed on a desktop computer with an i5-4460 CPU @3.2 GHz and 4gigabyte RAM, running Windows 7, 64 bit system and VC++ 6.0, by using PBC Library [10]. The time consumptions were obtained from a repeated simulations to obtain an objective computational cost comparison (see Figure 2) of ours with Yang *et al.* [18], Tang *et al.* [15,16], Ma *et al.* [11,13], and Wu *et al.* [17]. We assume both schemes were experimented on the same desktop computer.

Obviously, our encryption (Enc) computational cost seems higer than other related schemes. This is due to the extra computational overheads by a generation of an instance from a witness relation to resist insider adversary. Decryptions and test computations were comparable to other schemes (see Figure 2). Although time consumption of encryption (Enc) is slightly higher than in [17] scheme, it provides enhanced security to resist insider attack.

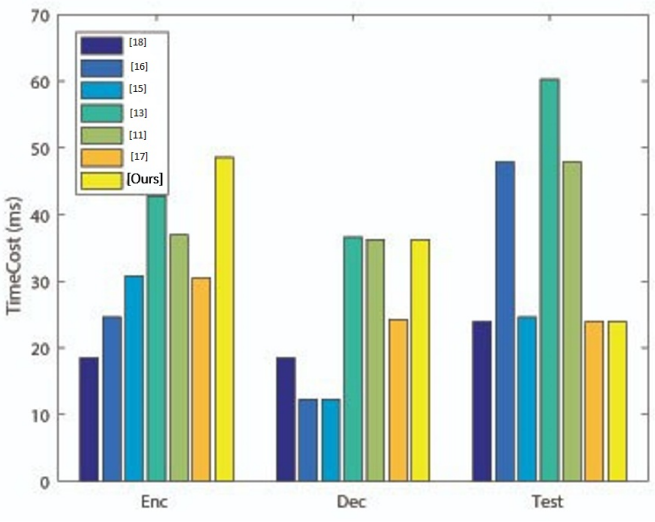


Figure 2: Computational time consumption comparison

7 Conclusion

Our scheme ensures a security improvement in [9,17]. We delegate a cloud server to perform equality test on users ciphertext. Such authorization cause the cloud server to launch the insider attack which our scheme resist such an attack. However, our scheme ensures that even though the cloud server is authorized to perform equality test, it could not launch the insider attack on users cipheretext. Our scheme ensures a resistant to insider attack by the adoption of witness based cryptographic primitive. Our scheme support weak indistinguishable identity chosen ciphertext attack security (W-IND-ID-CCA) with extended trapdoor security (TD).

References

[1] S. Alornyo, M. Asante, X. Hu, and K. K. Mireku, "Encrypted traffic analytic using identity based encryption with equality test for cloud computing," in *IEEE the 7th International Conference on Adaptive Science and Technology (ICAST'18)*, pp. 1-4, 2018.

[2] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 506-522, 2004.

[3] J. W. Byun, H. S. Rhee, H. A. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *Workshop on Secure Data Management*, pp. 75-83, 2006.

[4] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "A new general framework for secure public key encryption with keyword search," in *Australasian Conference on Information Security and Privacy*, pp. 59-76, 2015.

[5] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 789-798, 2016.

[6] S. Garg, C. Gentry, A. Sahai, and B. Waters, "Witness encryption and its applications," in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pp. 467-476, 2013.

[7] K. Huang, R. Tso, Y. C. Chen, S. M. M. Rahman, A. Almogren, and A. Alamri, "PKE-AET: Public key encryption with authorized equality test," *The Computer Journal*, vol. 58, no. 10, pp. 2686-2697, 2015.

[8] I. R. Jeong, J. O. Kwon, D. Hong, and D. H. Lee, "Constructing PEKS schemes secure against keyword guessing attacks is possible?," *Computer Communications*, vol. 32, no. 2, pp. 394-396, 2019.

[9] H. T. Lee, H. Wang, and K. Zhang, "Security analysis and modification of ID-based encryption with equality test from ACISP 2017," *Information Security and Privacy*, pp. 780-786, 2018.

[10] B. Lynn, *The Stanford Pairing based Crypto Library.* (`http://crypto.stanford.edu/pbc/`)

[11] S. Ma, "Identity-based encryption with outsourced equality test in cloud computing," *Information Sciences*, vol. 328, pp. 389-402, 2016.

[12] S. Ma, Y. Mu, W. Susilo, and B. Yang, "Witness-based searchable encryption," *Information Sciences*, vol. 453, pp. 364-378, 2018.

[13] S. Ma, M. Zhang, Q. Huang, and B. Yang, "Public key encryption with delegated equality test in a multi-user setting," *The Computer Journal*, vol. 58, no. 4, pp. 986-1002, 2015.

[14] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Trapdoor security in a searchable public-key encryption scheme with a designated tester," *Journal of Systems and Software*, vol. 83, no. 5, pp. 763-771, 2010.

[15] Q. Tang, "Public key encryption schemes supporting equality test with authorisation of different granularity," *International Journal of Applied Cryptography*, vol. 2, no. 4, pp. 304-321, 2012.

[16] Q. Tang, "Public key encryption supporting plaintext equality test and user-specified authorization," *Security and Communication Networks*, vol. 5, no. 12, pp. 1351-1362, 2012.

[17] T. Wu, S. Ma, Y. Mu, and S. Zeng, "ID-based encryption with equality test against insider attack," in *Australasian Conference on Information Security and Privacy*, pp. 168-183, 2017.

[18] G. Yang, C. H. Tan, Q. Huang, and D. S. Wong, "Probabilistic public key encryption with equality test," in *Cryptographers' Track at the RSA Conference*, pp. 119-131, 2010.

[19] W. C. Yau, S. H. Heng, and B. M. Goi, "Off-line keyword guessing attacks on recent public key encryption with keyword search schemes," in *International Conference on Autonomic and Trusted Computing*, pp. 100-105, 2008.

Acknowledgments

We wish to thank the anonymous reviewers for their comments and contributions.

Biography

Seth Alornyo is a lecturer at Koforidua Technical University. He received his Master of Philosophy(M.Phil) degree from Kwame Nkrumah University of Science and Technology in 2014, bachelor of science degree, computer science in 2012 and a higher national diploma in 2008. Currently, he is pursuing his Ph.D. in Software Engineering at University of Electronic Science and Technology of China. His research interests are Cryptography and Network Security. A Member of IEEE.

Acheampong Edward Mensah received his masters degree of Software Engineering from the University of Electronic Science and Technology of China in 2019. His research interests are in the areas of cryptography and information security.

Abraham Opanfo Abbam received his bachelor in Information and Communication Technology degree from University of Education, Winneba. He is currently pursuing software engineering masters degree program at University of Electronic Science and Technology of China. His research interest lies in the area of Deep Learning specifically Natural Language Processing.

One-Code-Pass User Authentication Based on QR Code and Secret Sharing

Yanjun Liu[1], Chin-Chen Chang[1], and Peng-Cheng Huang[1,2]
(Corresponding author: Peng-Cheng Huang)

Department of Information Engineering and Computer Science, Feng Chia University[1]
Taichung 40724, Taiwan
College of Computer and Information Engineering, Xiamen University of Technology[2]
Xiamen, Fujian, China
(Email: pc4hpc@gmail.com)

(Received July 10, 2018; Revised and Accepted Feb. 7, 2019; First Online Oct. 6, 2019)

Abstract

The quick response (QR) code has gained extensive popularity in information storage and identification in our daily lives due to its small printout size, high storage capacity and error correction ability. Taking into account these fascinating features of the QR code, we propose a user authentication protocol based on QR code and secret sharing. It is the first user authentication protocol that implements the "one-code-pass" functionality in which an authorized user can use only one QR code to access various services from all departments within an organization. In the proposed protocol, a secret is divided into many shares in the form of QR codes held among different departments and users. The original secret must be restored by the cooperation of both the department's share and the user's share for the authentication. Experimental results demonstrate that the proposed protocol has high robustness and is secure against various typical attacks.

Keywords: One-Code-Pass; QR Code; Security; Secret Sharing; User Authentication

1 Introduction

The user authentication mechanism has been regarded as a very important technique to efficiently verify the identity of the user through various kinds of secure communications. In 1981, for the first time Lamport [8] introduced a password-based remote user authentication protocol. Since then, numerous variants [6, 7, 9, 18] of Lamport's protocol have been developed in the literature, noticeably extending the scope of applications for user authentication in the field of secure communications, such as the agreement and distribution of session keys, e-business systems and wireless network communications, *etc.*

At present, user authentication protocols generally fall into four categories, *i.e.*, password-based, smart card-based, biometrics-based and image morphing-based protocols. In a password-based user authentication protocol [8, 18], the user first registers at the remote server by transmitting a selected password to the server privately. Then, the server can provide the user with required services after the authentication process, in which the password presented by the user is checked to confirm that the user is legal. However, a cost-inefficient problem arises due to the fact that a verifier table maintaining users' private information for authentication must be stored on the server and the volume of the verifier table is in proportion to the increased number of users. Therefore, the verifier table needs to occupy much storage space when a lot of users need to be authenticated. Besides, private information of the users may be leaked out since the verifier table is prone to be stolen with malicious intention.

To overcome the aforementioned shortcomings, the smart card is employed extensively in the design of a user authentication protocol. There are two main advantages for the use of smart card [6]. Firstly, the level of security can be enhanced significantly. Each user individually owns a smart card which stores important and confidential information for mutual authentication that can authenticate not only the user but also the remote server. Furthermore, the private information of all users is fully dispersed by smart cards rather than being centralized in a verifier table, substantially attenuating the risk of information disclosure. Secondly, the authentication protocol becomes more cost-efficient because there is no need to maintain a great deal of users' information on the server. A typical smart card based authentication protocol is conducted as follows [9]. First of all, the user submits a registration request to the server, and then, the server delivers a smart card storing private information of both the user and the server to the user over a secure channel. When the user inserts the smart card into a card reader for specific services provided by the server, the identity of the user should be verified to confirm that the user is the

true card owner. The private information maintained in the smart card is extracted by the card reader and then transmitted to the server for mutual authentication. If the authentication is passed, a shared session key is usually established by the authorized user and the server to ensure subsequent secure communication.

Unfortunately, the smart card is liable to be forged since an illegal user may easily invade the secret information preserved in the smart card due to weak computing power [7]. Consequently, biometrics information [16] referring to unique biometric feature of a person that is unable to be stolen or forged, has been employed as a popular tool to further increase the authentication security. Fingerprint, face, iris, voice and gait are the most common biometric information used to verify the identity of the user since it is impossible for two people to present identical data of these biometric features [20, 21]. The biometric information of the true user is usually stored in advance. For authentication, the current biometrics feature is retrieved by the biometric reader and then compared with the stored information. If they can match, the identity of the user is authenticated; otherwise, the authentication process fails.

Recognition rate that indicates the identification accuracy is one of the most important measurements for performance evaluation of a biometrics-based authentication protocol [21]. Therefore, the research subject on recognition rate has attracted scholars' attention in recent years and great recognition rate has been obtained by unceasing improvements on feature matching algorithms. However, this type of authentication protocol has the following weaknesses [16, 20, 21]. Firstly, it just provides weak privacy protection since personal biometric information may be disclosed and then abused by malicious adversaries. Secondly, if the true user's biometric information is destroyed because of diseases or accidents, it will induce incorrect authentication. At last, development of a robust biometric reader is both time-consuming and money-consuming.

To make further efforts on the uplift in security and performance, a new type of authentication protocols based on image morphing has flourished. The image morphing technology suspends from the production of special image effects in the film industry in such a way that given a source image and a target image, a morphed image is created. The morphed image looks like both the source image and the target image. This attractive characteristic can be well applied to identity authentication. Up to date lots of user authentication protocols combining image morphing with smart card are introduced [13, 15, 22]. Zhao and Hsieh [22] presented a card user authentication protocol in which a morphed image MI is first created via the card owner's face image OI and a pre-selected face image SI and then printed on the smart card. When a card user needs to be authenticated, the image OI is recovered by a de-morphing process operating on the images MI and SI. Then, the image OI is compared with the face image of the user. If they are the same, it implies that the user is legal; otherwise, an unauthorized user is successfully detected. Since the original face image of the card owner is no longer stored anywhere, this protocol can furnish efficient privacy protection and higher security level. In 2015, Mao *et al.* [15] introduced a proxy user authentication protocol. In their protocol, the proxy user has authority to act on behalf of the primary user and all users are able to be authenticated by image exchange based on morphing technology.

In 2017, Liu and Chang [13] innovatively extended the image morphing-based authentication protocol to a "one-card-pass" scenario with high practical use. Under this scenario, an organization contains various kinds of departments providing diverse services. An authorized user can register at any department to acquire a smart card storing a generated morphed image. After that, the user can utilize this smart card to obtain services from different departments without registering again if the user passes the authentication such that the face image of the user is identical to that restored by de-morphing the morphed image and another face image maintained on the cloud storage servers. Nevertheless, most of the morphed images generated in authentication protocols look unnatural, which is prone to arouse suspicion among malicious attackers. Therefore, the research of optimization algorithms [14] on the selection of control points to achieve better visual effect of morphed images becomes a very challenging task.

Nowadays, the quick response (QR) code [11, 12] plays a very important role in information storage and identification in our daily lives. The QR code has many fascinating features [11, 12, 19], such as small printout size, high storage capacity and error correction ability. Since the QR code is very easy to be decoded by any standard QR code reader, it also has gained extensive popularity in real-time applications. Due to these advantages, the QR code is very suitable for storing the user's personal information to verify the identity of the user. In 2018, Huang *et al.* [5] applied the aesthetic QR code and the single-key-lock mechanism to a smart-building access control system. The single-key-lock mechanism produces keys for users and locks for doors. After encryption and fusion procedures, the keys are used to generate aesthetic QR codes as the user's credential. This access control system has the attributes of high security and robustness. Different from Huang *et al.*'s access control system, we consider the application of QR code from another aspect. The application scenario of our user authentication protocol is similar to that of Liu and Chang's morphing-based "one-card-pass" method, but ours offers "one-code-pass" functionality via the combination of QR code and secret sharing. That is to say, an authorized user can successfully use only one QR code to access various services from all departments within an organization. Suppose that there is a secret provided by the server of the organization in the proposed protocol. The secret is divided into many shares in the form of beautified QR codes and each department holds its own share beforehand. During the

registration, a user can register at any department and acquire his/her share from the department. When the user wants to access services of any of the departments, the original secret must be restored by the cooperation of both the department's share and the user's share; knowing only one share is unable to derive any valuable information about the secret. If the restored secret is not correct, the user is illegal. Experimental results demonstrate that the proposed protocol is secure against various attacks and the generated beautified QR code has high robustness.

The rest of the paper is organized as follows. In Section 2, we first introduce elementary knowledge of QR code and secret sharing, and then, review related works [13] and [5]. In Section 3, a novel "one-code-pass" user authentication protocol based on QR code and secret sharing is proposed. Section 4 demonstrates the experimental results of the proposed protocol and Section 5 gives the security and performance analyses. Finally, our conclusions are presented in Section 6.

2 Preliminaries

In this section, we first introduce main building blocks of a new user authentication protocol that will be proposed in the next section. After that, we provide a brief review of the morphing-based "one-card-pass" authentication protocol [13] and the QR code-based access control system [5].

2.1 QR Code

The QR code, invented by the Japanese Denso-Wave Company in 1994 [4], is a two-dimensional graphical code that consists of black and white square modules representing bit information. As illustrated in Figure 1, the structure of a standard QR contains message region, padding region and error correction region, together with version information, format information, position detecting patterns, alignment patterns and timing patterns.

The QR code is extensively used in information storage and identification applications due to its advantages on small printout size, high storage capacity and error correction ability [11, 12, 19]. As many as 40 QR code versions are used to determine different storage capacities such that the QR code with a higher version number can provide larger data payload. As another extraordinary feature of the QR code, the error correction capability with four error correction levels (*i.e.* L, M, Q and H) ensures successful decodability when portions of the QR code are dirty or damaged.

One of the most important processes for generating a QR code is to encode information using Reed-Solomon (RS) code. A t-bit RS code, shown in Figure 2, is composed of three segments, *i.e.*, l message bits, m padding bits and n parity bits. The to-be-embedded message is first decoded into a l-bit binary stream, followed by m padding bits. Then, n parity bits are created for detecting and correcting errors when scanning the QR code. Figure 1 demonstrates how to place an RS code into a 2D QR code. Especially, each message/padding/parity bit is located onto one module of the message/padding/error correction region.

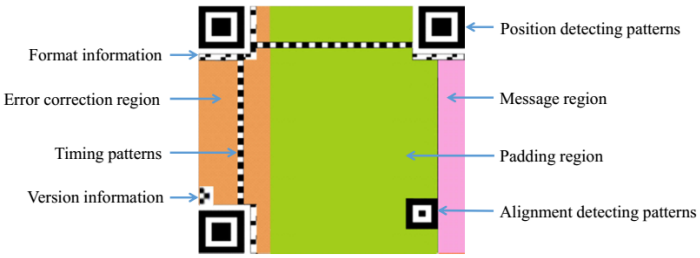


Figure 1: The structure of a QR code

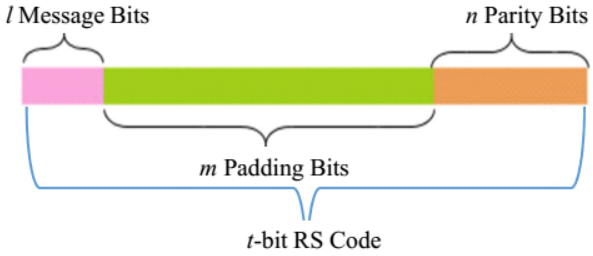


Figure 2: RS code

2.2 Shamir's Threshold Secret Sharing

The concept of secret sharing, independently introduced by Shamir [17] and Blakley [3] in 1979, is an efficient mechanism for data protection. The popularity of Shamir's threshold secret sharing (TSS) has noticeably increased due to its high efficiency and security. In a (t, r) TSS scheme ($t \leq r$), a secret is divided into r shares to be held by r participants. The secret can be simply recovered by the collaboration of at least t shares; otherwise, if fewer than t shares are collected, the original secret cannot be retrieved correctly.

Assume that there is a dealer and r participants, and the secret is denoted as s. The (t, r) TSS scheme contains the share establishment phase and the secret recovery phase, which are described as follows.

2.2.1 Share Establishment Phase

The dealer randomly selects a Lagrange interpolating polynomial g with degree t-1 such that $g(x) = s + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \bmod p$, where p is a prime number, and $a_1, a_2, \ldots, a_{t-1}$ and s are in the finite field GF(p). Then, the dealer chooses r random numbers x_d for $d = 1, 2, \ldots, r$ to establish r shares $s_d = g(x_d)$ for $d = 1, 2, \ldots, r$, and provides each participant with a share.

2.2.2 Secret Recovery Phase

The t participants can cooperate to recover the secret s by releasing their shares. Suppose the t out of r shares are denoted as $s_{bh} \in \{s_1, s_2, \ldots, s_r\}$ for $h = 1, 2, \ldots, t$. According to the principle of the Lagrange interpolating polynomial, $g(x)$ can be recovered by $g(x) = \sum_{h=1}^{t} s_{bh} \prod_{k=1, k\neq h}^{t} \frac{x - x_{bk}}{x_{bk} - x_{bh}} \bmod p$. Thus, the secret s can be immediately obtained by $s = g(0)$.

2.3 Related Works

2.3.1 Morphing-Based "One-Card-Pass" Authentication Protocol

Assume that an organization includes many independent departments. For instance, a university may contain a library, a digital information center, a fitness center, a student association, *etc.* We hope that the "one-card-pass" functionality could be provided in such a way that a faculty member or a student could use only one smart card to gain various services from all of the departments. Accordingly, the objective of the "one-card-pass" user authentication protocol is to verify the identity of the card user in each department after the user registers at any of the departments.

Recently, Liu and Chang [13] proposed the first "one-card-pass" user authentication protocol using image morphing technology. This protocol contains three entities, *i.e.*, a card user, a terminal within a department, and cloud storage servers, and it consists of the registration phase and the authentication phase.

In the registration phase, a legal user U can register at any terminal T within any department. The following steps are conducted to fulfill the registration.

Step 1. The user U sends a registration request including his/her identity number to a terminal T.

Step 2. T takes a digital photo of U. This photo, denoted as OI, is used as the original face image of U.

Step 3. T selects a face image SI stored in the cloud storage servers C according to U's identity number and then C sends SI to T.

Step 4. T generates a morphed image MI using OI as the source image and SI as the target image by a specific morphing algorithm.

Step 5. T stores the information that will be used in the authentication phase in a smart card and prints the morphed image MI on the smart card. Then, T delivers the smart card to U.

Later, in the authentication phase, a terminal T of a department recovers the face image of the true card owner via image de-morphing to verify the validity of the card user U. If the face image of U matches the recovered face image, U passes the authentication and can gain the service provided by this department. The detailed steps in this phase are listed below.

Step 1. User U presents the smart card to a terminal T.

Step 2. T scans the morphed image MI printed on the smart card and extracts the information for authentication from the smart card.

Step 3. T finds the face image SI stored in the cloud storage servers C according to the extracted information.

Step 4. C sends SI to T.

Step 5. T recovers the face image of the true card owner, OI, by de-morphing images MI and SI.

Step 6. T compares the face image of the user U with OI to see if they are identical. If it holds, U is legal.

This protocol is very practical since it implements the "one-card-pass" functionality. The strategy that the real image of the true card owner is not stored anywhere but hidden in a morphed image increases the security to a certain extent. However, the protocol requires an additional set of cloud storage servers that preserve large numbers of face images for the morphing operation, which may lead to potential security problems. Furthermore, since the procedures of printing and scanning may cause noises, the authentication result may not be correct due to the distortions generated to the recovered face image. To solve these problems, in this paper, we will combine QR code and secret sharing to design a new "one-code-pass" user authentication protocol in Section 3.

2.3.2 QR Code-Based Access Control System

Huang *et al.* [5] presented a novel access control system for smart building based on QR code. The enrollment procedure of this system consists of three steps, *i.e.* the keys and locks generation, key encryption, and the QR code beautification, as listed below:

Step 1. Keys and locks generation. The single-key-lock mechanism is used to generate keys for users and locks for doors on a randomly non-singular matrix.

Step 2. Key encryption. Each key is encrypted by a symmetrickey cryptography algorithm to prevent the leak of key. Then, a QR code generated by the encrypted keys is sent to the user.

Step 3. QR code beautification. The owner of original QR code produced by Step 2 cannot be distinguished by human eyes for a number of confused black and white modules. This would induce difficulties in the management of QR codes when the number of users (keys) grows sharply. Therefore, these QR code need to be beautified to show the user's photo on its own. The beautification process includes:

a) Construct the basis vector matrix.

b) Generate an XORed QR code by perform the XOR operation between the Reed-Solomon message of the QR code and the basis vector matrix. In the XORed QR code, most modules in the padding region are modified and middle modules are kept clear to make the preparation of embedding the user's photo.

c) Synthesize an aesthetic QR code from the user's photo and the XORed QR code as the user's credential.

When a user presents the aesthetic QR code to request the access to a door, the reader on the door would decrypt and retrieve the user's key from the aesthetic QR code, and then, verifies the access right via conducting an operation on the key and the lock.

3 The Proposed Protocol

In this section, we propose a novel "one-code-pass" user authentication protocol. The proposed protocol contains the initialization phase, the registration phase and the authentication phase. In the following, we first point out the contribution of the proposed protocol, then address its main idea, and finally elaborate on each phase.

3.1 Contribution

The contribution of the proposed protocol is described as follows:

1) It is the first user authentication protocol that can fully accomplish the goal of "one-code-pass" in a specified organization. More specifically, a user is able to make registration on any department to gain his/her own beautified QR code. By using the QR code, the user is immediately authenticated and directly accesses diverse services from all departments in this organization.

2) The QR code technique is combined with secret sharing mechanism to implement the proposed protocol. A secret is divided into a number of shares that are held among different departments and authorized users. The share actually is a beautified QR code containing some authentication information. Each department gains its share in advance and each user obtains his/her share from the department on which the registration operation for the user is conducted. To verify the identity of the user, the original secret should be recovered by the cooperation of both the department's share and the user's share. If the restored secret is correct, the user is eligible to get services from all departments.

3) The proposed protocol is significantly secure. Based on the secret sharing mechanism, both shares on the user and department sides must be collected to recover the original secret; Just one share is impossible to leak the secret. In addition to this, the proposed protocol is secure against various attacks, such as the impersonation attack, the collusion attack, the replay attack, *etc.* It can also achieve high robustness as it is tolerant of various quality degradation situations.

3.2 Main Idea

To authenticate the identity of a user, a $(2, M+N)$ TSS is exploited in the proposed protocol for an organization containing M users and N departments. At first, some initialization work should be done. The server of the organization selects a secret s and generates N shares such that each share is held by a corresponding department. Then, for each user, he/she can register at any department and acquire his/her own share from the department with the help of the server. It is noted that each share of a department/user is a beautified QR code that not only encodes authentication information, but also embeds a logo/photo of the department/user in its padding region for efficient management of QR codes. When a user wants to access services of any of the departments, the user is authenticated by recovering the original secret s through the cooperation of both the department's share and the user's share. The photo of the authorized user on the beautified QR code also provides an auxiliary way for user authentication. Figure 3 shows the architecture of the proposed protocol, in which a user first registers at the department 1, and then is authenticated by the departments 1 and 2, respectively.

Before addressing the detailed protocol, some notations used in the paper are described in Table 1.

Table 1: Notation description

Notation	Description
S	The server
$U_i (1 \leq i \leq M)$	The user i
ID_{U_i}	The identity number of U_i
Q_{U_i}	The beautified QR code for U_i
$D_j (1 \leq j \leq N)$	The department j
ID_{D_j}	The identity number of D_j
Q_{D_j}	The beautified QR code for D_j
s	The secret provided by S
K	The secret key shared among all departments
$g(\cdot)$	The Lagrange interpolating polynomial
$h(\cdot)$	A collision-free one-way hash function
$\|$	The string concatenation operation

3.3 The Initialization Phase

This phase allows the server to generate a share in the form of a beautified QR code for each department. As-

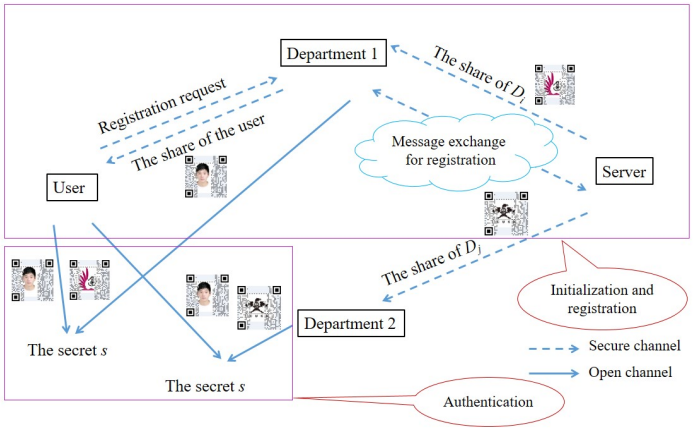


Figure 3: Architecture of the proposed protocol

sume that all of the N departments within the organization share a secret key K. Firstly, the department D_j computes $x_{D_j} = h(ID_{D_j} \parallel K)$ and sends it to the server S. Then, S selects a secret s, a number a_1 and a prime number p to generate a one-degree-polynomial function g as

$$g(x) = s + a_1 x \,(\mathrm{mod}\, p). \tag{1}$$

For each department, S computes $y_{D_j} = g(x_{D_j}) = s + a_1 x_{D_j} (\mathrm{mod}\, p)$. Finally, S creates a beautified QR code Q_{D_j} for the department D_j as the share of D_j by using the information (x_{D_j}, y_{D_j}) and D_j's logo.

Here, we briefly explain how the beautified QR code for a department is generated. The information (x_{D_j}, y_{D_j}) is first placed into modules of the message region to produce a QR code. Then, a beautification algorithm on the QR code is made for efficient management. There are many available beautification algorithms and we use the one presented in [10]. In this algorithm, the modules in the padding region of the QR code is first modified to keep "clean" with the help of a basis vector matrix, and then, the department's logo is embedded into the padding region by a synthetic strategy. Interested readers can refer to [10] for a better understanding.

3.4 The Registration Phase

In this phase, a user can register at any department and acquire his/her own share, also in the form of a beautified QR code. This phase is described in detail as follows and demonstrated in Figure 4.

Step 1. The user U_i sends his/her identity number ID_{U_i} to the department D_j.

Step 2. D_j computes $x_{U_i} = h(ID_{U_i} \parallel K)$ and sends x_{U_i} to the server S.

Step 3. S computes $y_{U_i} = g(x_{U_i}) = s + a_1 x_{U_i} (mod\, p)$ by Equation (1).

Step 4. S generates a beautified QR code Q_{U_i} for the user U_i as the share of U_i by using the information (x_{U_i}, y_{U_i}) and U_i's photo. The method of generating the beautified QR code for the user is the same as that for the department as mentioned in the initialization phase.

Step 5. S sends Q_{U_i} to D_j.

Step 6. D_j sends Q_{U_i} to U_i.

3.5 The Authentication Phase

The authentication is conducted by an authentication device within a department which is equipped with a QR code reader and has a copy of the secret s. The authentication device verifies the identity of the user by recovering the original secret s through the collaboration of the user's QR code and the department's QR code. If the user passes the authentication, he/she can gain services provided by this department. The steps of this phase are addressed as follows and illustrated in Figure 5.

Step 1. The user U_i shows the beautified QR code Q_{U_i} to the authentication device.

Step 2. The authentication device scans the QR code Q_{U_i} of U_i and extracts (x_{U_i}, y_{U_i}).

Step 3. The authentication device scans the QR code Q_{D_j} of the department D_j where it is located and retrieves (x_{D_j}, y_{D_j}).

Step 4. The authentication device uses (x_{U_i}, y_{U_i}) and (x_{D_j}, y_{D_j}) as two shares to restore a secret s'. If $s' \neq s$, the authentication device terminates the phase and the authentication fails; otherwise, it executes Step 5.

Step 5. The authentication device computes $x'_{U_i} = h(ID_{U_i} \parallel K)$ and then checks whether x'_{U_i} equals x_{U_i}. If it holds, the authentication device confirms

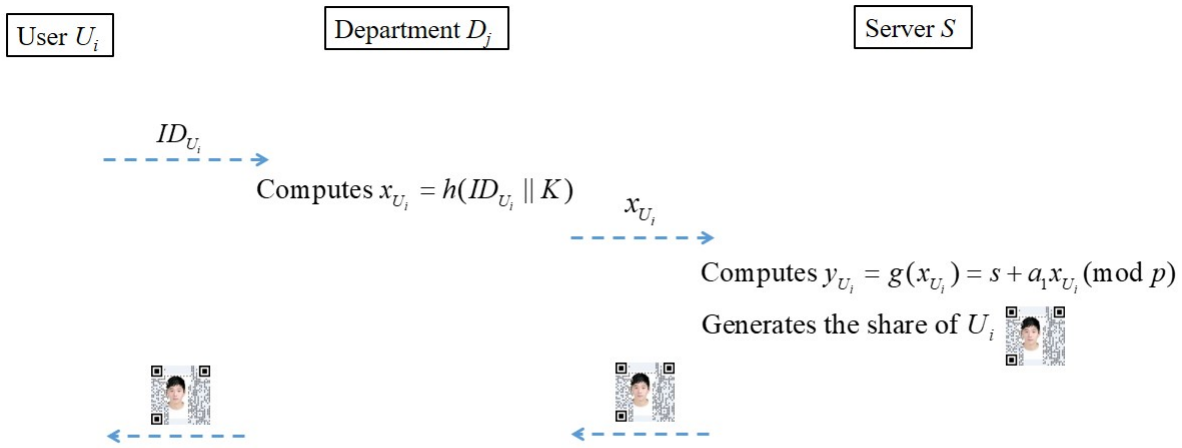


Figure 4: The registration phase of the proposed protocol

that the user is legal; otherwise, the phase is terminated and the authentication fails.

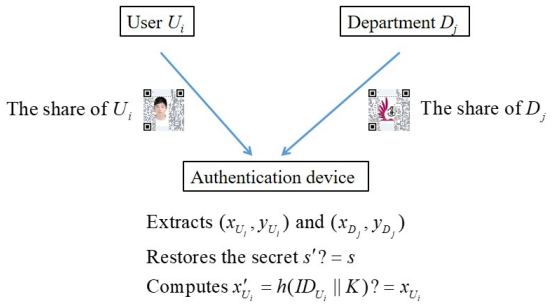


Figure 5: The authentication phase of the proposed protocol

In the authentication phase, it is worth noticing that if the restored secret s' is different from the original secret s by Step 4, the user is definitely illegal. However, if s' is equal to s, it cannot confirm that the user is legal. This is because if an attacker impersonates a user to embed fake values of x''_{U_i} and y''_{U_i} that satisfy $y''_{U_i} = g(x''_{U_i}) = s + a_1 x''_{U_i} (\text{mod } p)$ into the QR code, the correct secret s can still be obtained by the two points (x_{D_j}, y_{D_j}) and (x''_{U_i}, y''_{U_i}) on the function g. Therefore, Step 5 is added to check whether x''_{U_i} equals $h(ID_{U_i} \| K)$ when $s' = s$ to ensure that x''_{U_i} is real.

4 Experimental Results

To evaluate the performance of our proposed protocol, our experiments were implemented by the open source computer vision library OpenCV and C++ program language. In this section, three users and two departments are involved into a concrete example to illustrate the experimental results. In the following example, we select $s = 1024, a_1 = 21, p = 1237$ and $K =$"@MSNLAB". Thus, the function g generated by the server S for the initialization becomes $g(x) = 1024 + 21x (\text{mod } 1237)$. The beautified QR code Q_{D_j} for each of the two departments is generated in the initialization phase and shown in Table 2, where the department's logo, the identity number ID_{D_j} and the information (x_{D_j}, y_{D_j}) used for the generation of Q_{D_j} are also provided. After the registration on the department D_1 or D_2, each of the three users obtains a beautified QR code Q_{U_i} by using the user's photo, the identity number ID_{U_i} and the information (x_{U_i}, y_{U_i}), as shown in Table 3. In addition, the departments' logos in Table 2 are provided by Feng Chia University and the users' photos in Table 3 come from the Yale Face Database [2].

Here, we only take the user authentication process performed by the department D_1 as an example. Now let us demonstrate how the department D_1 verifies the identity of the user U_1 in the proposed protocol. Firstly, the authentication device in D_1 scans D_1's beautified QR code Q_{D_1} (See in Table 2) and U_1's beautified QR code Q_{U_1} (See in Table 3), respectively. Secondly, the information (x_{D_1}, y_{D_1}) and (x_{U_1}, y_{U_1}) are extracted from Q_{D_1} and Q_{U_1}, respectively, and then used as two shares to reconstruct a function g' as

$$\begin{aligned} g'(x) &= y_{U_1} \frac{x - x_{D_1}}{x_{U_1} - x_{D_1}} + y_{D_1} \frac{x - x_{U_1}}{x_{D_1} - x_{U_1}} (mod\, 1237) \\ &= 1024 + 21x (mod\, 1237). \end{aligned}$$

Then, a secret s' is derived by $s' = g'(0) = 1024$, which is equal to the original secret s. Finally, we compute $x'_{U_1} = h(ID_{U_1} \| K)$ and find that x'_{U_1} equals x_{U_1}. This indicates that the user U_1 passes the authentication.

In the following, we also show how the department D_1 detects an illegal user Bob by the proposed protocol. Assume that Bob uses $x_B =$

Table 2: The beautified QR codes for the departments

	Department D_1	Department D_2
Logo		
ID_{D_j}	http://lib.fcu.edu.tw	http://www.sa.fcu.edu.tw
$x_{D_j} = h(ID_{D_j} \parallel K)$	dc1cf84361a49864 24325645bef5be80 2344dc55	742b34c092beb4ea e998a104509635fe 32c394f3
$y_{D_j} = g(x_{D_j})$	607	107
Q_{D_j}		

6ed3693a3ef2fa0421cd0d1bd36750691522ce29, $y_B = 671$ and his photo to forge his QR code Q_B and then shows Q_B to the authentication device. The authentication device uses (x_B, y_B) extracted from Q_B and (x_{D_1}, y_{D_1}) derived from Q_{D_1} to reconstruct a function g'' as

$$\begin{aligned} g''(x) &= y_B \frac{x-x_{D_1}}{x_B-x_{D1}} + y_{D_1} \frac{x-x_B}{x_{D_1}-x_B} (\text{mod } 1237) \\ &= 287 + 16x (\text{mod } 1237). \end{aligned}$$

Obviously, a secret s' is derived by $s' = g''(0) = 287 \neq 1024$, which implies that Bob is an illegal user.

5 Analyses

In this section, we give the security and performance analyses of the proposed protocol, respectively.

5.1 Security Analysis

The proposed protocol can efficiently protect the user's privacy and resist various typical attacks, such as the impersonation attack, the collusion attack and the replay attack. The detailed security analysis from the theoretical aspect is given as follows.

5.1.1 Privacy Protection

One property of the QR code is that the information embedded in the message region of the QR code can be decoded and extracted by any QR code reader. As a result, a malicious attacker must be prevented from retrieving any personal knowledge about the user from the extracted information. To achieve this goal, the user U_i's identity number ID_{U_i} is not embedded directly into the QR code, but first encrypted by a one-way hash function as $x_{U_i} = h(ID_{U_i} \parallel K)$, and then, the encrypted message x_{U_i} and the corresponding value y_{U_i} computed via x_{U_i} by Equation (1) are used to generate the user U_i's QR code Q_{U_i}. When a malicious attacker scans the QR code Q_{U_i} through a QR code reader, he/she only gets the messages x_{U_i} and y_{U_i} but has no idea of the user's identity number ID_{U_i}. By this way, the user's personal information will not be disclosed so that privacy protection is achieved. Besides, it doesn't matter that the photo embedded on the QR code reveals the appearance of the user since it is useless for the user authentication.

5.1.2 Withstanding Impersonation Attack

According to the proposed protocol, the impersonation attack refers to an attack that a malicious intruder forges a QR code and then impersonates a user with the intention of passing the authentication. To launch this kind of attack, the intruder must produce fake values of x_{U_i} and y_{U_i} and use them to forge the QR code. There is a strong possibility that the fake x_{U_i} and y_{U_i} do not satisfy the equation $y_{U_i} = g(x_{U_i}) = s + a_1 x_{U_i} (\text{mod } p)$, thus the secret s restored by the cooperation of the two shares (x_{U_i}, y_{U_i}) and (x_{D_j}, y_{D_j}) is different from the original secret s. This implies that the impersonation attack is successfully detected in this situation. However, there also exists a situation that the intruder knows the function $g(x) = s + a_1 x (\text{mod } p)$ and produces fake x_{U_i} and y_{U_i} satisfying the function g such that the correct secret s can still be obtained. Since under this case the value of s cannot determine whether the user is legal, the proposed protocol will further compute $x'_{U_i} = h(ID_{U_i} \parallel K)$. The value x_{U_i} created by the intruder will not equal x'_{U_i} because the intruder does not know the value of K, which indicates that x_{U_i} is fake and the impersonation attack fails.

5.1.3 Withstanding Collusion Attack

The proposed protocol must be collusion-resistant. The collusion attack in the proposed protocol means that, if multiple authorized users collude, they can obtain important information and use it to help an attacker to pass the authentication. Since it is sufficient for any two shares working together to restore the secret s, two authorized users can release their shares and collude to recover the function $g(x) = s + a_1 x (\text{mod } p)$. The two users may share the function g with an attacker who attempts to impersonate a legal user. The attacker can create the values of x_{U_i} and y_{U_i} satisfying the function g and the correct secret s can be obtained in the authentication phase. Nevertheless, the attacker cannot pass the authentication since the created value x_{U_i} will not equal $x'_{U_i} (x'_{U_i} = h(ID_{U_i} \parallel K))$ as addressed in Subsection 5.1.2. Therefore, the collusion attack cannot be launched successfully.

Table 3: The beautified QR codes for the users

	User U_1	User U_2	User U_3
Photo			
ID_{U_i}	WangJiaQing#M0557591	ZhuZhaoHua#M0419274	LiJing#P0361138
$x_{U_i} = h(ID_{U_i} \parallel K)$	a2e5bc07294a34caaffcc79 264e0f2aec912b37a	3cf5b5ffb9153545bd8bbf1 f0edb154ec2691fc1	53132e6ffe935538492e213f 9c232d308c9c0488
$y_{U_i} = g(x_{U_i})$	788	709	565
Photo			

5.1.4 Withstanding Replay Attack

In a replay attack, a valid data transmission is repeated or delayed by a malicious attacker without detection. The proposed scheme can withstand such attack by the following way. An attacker can steal or duplicate a valid QR code and tries to use it to pass the authentication. One method to prevent it happening is that we can periodically change the value of the secret key K and accordingly update the QR codes for the users and departments based on the function g. Therefore, the QR code held by the attacker always expires since the attacker can never get the latest version of the user's QR code.

5.2 Performance Analysis

To evaluate the performance of the proposed protocol, we analyze the decoding rate and robustness of the QR codes for users and departments in this subsection.

5.2.1 Decoding Rate of Beautified QR Codes

To evaluate the decoding rate of the beautified QR code, we conduct experiments on both Apple's iOS and Google's Android mobile operation system. As shown in Table 4, several applications from Apple APP store and Google Play APP store were installed on two different mobile phones (iPhone 7, XiaoMi 4) and twenty beautified QR codes synthesized from users' authentication information with their photos were used to test the decoding rate of beautified QR codes. All these 20 users' photos are from Yale face database [2] and AT&T Cambridge face database [1]. From Table 4, we can see that all the APPs in these two selected phones can successfully decode the beautified QR code messages at the decoding rate of 100%.

5.2.2 Robustness of the Proposed Protocol

In the application scenario, when the beautified QR code was printed in the plastic card to be an ID card or scanned by a camera in the absence of adequate lighting condition, it usually suffers from several image degradation factors, such as pixel distortion, geometric distortion, noise, blur, and so on. These factors can be considered as a kind of image attack. Sometimes the quality of the QR code image being attacked has degraded significantly. Figure 6(a) shows the result of the beautified QR code Q_{U_i} of user U_1 in Table 3 suffering from the print-and-scan attack. The beautified QR code was printed in 600dpi with the HP LaserJet 500 color M551 printer, and scanned by the 200dpi with HP LaserJet M2727nf scanner. Figure 6(b) shows the result of Q_{U_i} suffering from Gaussian blur with the parameter $\sigma = 5$. Figure 6(c) shows the result of Q_{U_i} suffering from Gaussian noise with parameters $M = 0, V = 0.05$.

The authentication information embedded in these attacked beautified QR codes could still be decoded by any standard QR code reader. It demonstrates that the beautified QR code is tolerant to common attacks and the one-code-pass user authentication protocol is practically usable in the real-world applications.

6 Conclusions

In this paper, we proposed a novel "one-code-pass" user authentication protocol based on QR code and secret sharing such that an authorized user can successfully use only one QR code to access various services from all departments within an organization. In the proposed protocol, a secret is divided into many shares in the form of QR codes held among different departments and users. Each user can register at any department and acquire his/her

Table 4: Decoding rate in different applications

Mobile Phone	Applications (Developer)	Decoding Rate
iPhone 7 (iOS 10.3.2)	WoChaCha QR code (WoChaCha Information Technology)	100%
	Quick scan (iHandy Inc.)	100%
	Quick QR code reader & creator (Fellow Software)	100%
	QR code kit (Sima Biswas)	100%
	QuickMark (SimpleAct Inc.)	100%
XiaoMi 4 (Android 7.0)	WoChaCha QR code (WoChaCha Information Technology)	100%
	QR code extreme(FancyApp)	100%
	QR code reader (Scan.me)	100%
	Free QR code scanner (TWMobile)	100%
	QuickMark (SimpleAct Inc.)	100%

(a) (b) (c)

Figure 6: Results of beautified QR code of user U_1 in Table 3 after image degradation processes. (a) After Print-and-Scan attack; (b) After adding the Gaussian blurring with $\sigma = 5$; (c) After adding the Gaussian noise with parameters $M = 0, V = 0.05$

share including authentication information from the department. When a user wants to access services of any of the departments, the user is authenticated by recovering the original secret through the cooperation of both the department's share and the user's share. Theoretical analyses and experimental simulations show that the proposed protocol can efficiently protect the user's privacy and resist various typical attacks, such as the impersonation attack, the collusion attack and the replay attack.

References

[1] *At&t Face Database*, accessed 31 Oct. 2017. (`http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`)

[2] *Yale Face Database*, accessed 31 Oct. 2017. (`http://cvc.yale.edu/projects/yalefaces/yalefaces.html`)

[3] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the National Computer Conference*, vol. 48, pp. 313–317, 1979.

[4] Denso-Wave Inc. *QR code standardization*, accessed 31 Oct. 2017. (`http://www.qrcode.com/en/index.html`)

[5] P. C. Huang, C. C. Chang, Y. H. Li, and Y. Liu, "Efficient access control system based on aesthetic QR code," *Personal and Ubiquitous Computing*, vol. 22, no. 1, pp. 81–91, 2018.

[6] Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 383–393, 2015.

[7] S. Kumari, S. A. Chaudhry, F. Wu, X. Li, M. S. Farash, and M. K. Khan, "An improved smart card based authentication scheme for session initiation protocol," *Peer-to-Peer Networking and Applications*, vol. 10, no. 1, pp. 92–105, 2017.

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

[9] C. T. Li, "A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card," *IET Information Security*, vol. 7, no. 1, pp. 3–10, 2013.

[10] L. Li, J. Qiu, J. Lu, and C. C. Chang, "An aesthetic QR code solution based on error correction mechanism," *Journal of Systems and Software*, vol. 116, pp. 85–94, 2016.

[11] P. Y. Lin, "Distributed secret sharing approach with cheater prevention based on QR code," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 384–392, 2016.

[12] S. S. Lin, M. C. Hu, and T. Y. Lee, C. H. Lee, "Efficient QR code beautification with high quality visual content," *IEEE Transactions on Multimedia*, vol. 17, no. 9, pp. 1515–1524, 2015.

[13] Y. Liu and C. C. Chang, "A one-card-pass user authentication scheme using image morphing," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21247–21264, 2017.

[14] Q. Mao, K. Bharanitharan, and C. C. Chang, "Edge directed automatic control point selection algorithm for image morphing," *IETE Technical Review*, vol. 30, no. 4, pp. 343–243, 2013.

[15] Q. Mao, K. Bharanitharan, and C. C. Chang, "A proxy user authentication protocol using source-based image morphing," *The Computer Journal*, vol. 58, no. 7, pp. 1573–1584, 2014.

[16] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

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

[18] H. M. Sun, Y. H. Chen, and Y. H. Lin, "oPass: A user authentication protocol resistant to password stealing and password reuse attacks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 651–663, 2012.

[19] I. Tkachenko, W. Puech, C. Destruel, O. Strauss, J. M. Gaudin, and C. Guichard, "Two-level QR code for private message sharing and document authentication," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 571–583, 2016.

[20] J. Yu, G. Wang, Y. Mu, and W. Gao, "An efficient generic framework for three-factor authentication with provably secure instantiation," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2302–2313, 2014.

[21] L. Zhang, Y. Zhang, S. Tang, and H. Luo, "Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2795-2805, 2018.

[22] Q. Zhao and C. H. Hsieh, "Card user authentication based on generalized image morphing," in *2011 3rd International Conference on Awareness Science and Technology (iCAST'11)*, pp. 117–122, Sep. 2011.

Biography

Yanjun Liu received her Ph.D. degree in 2010, in School of Computer Science and Technology from University of Science and Technology of China (USTC), Hefei, China. She has been an assistant professor serving in Anhui University in China since 2010. She currently serves as a senior research fellow in Feng Chia University in Taiwan. Her specialties include E-Business security and electronic imaging techniques.

Chin-Chen Chang is a professor in Feng Chia University. He received the BS degree in Applied Mathematics in 1977 and the M.S. degree in Computer and Decision Sciences in 1979, both from the National Tsing Hua University, Taiwan. He received the Ph.D. degree in Computer Engineering in 1982 from the National Chiao Tung University, Taiwan. He is the author of more than 900 journal papers and has written 36 book chapters. His research interests include computer cryptography, data engineering, and image compression.

Peng-Cheng Huang is a lecturer at the Xiamen University of Technology. He received his BS degree from Xiamen University of Technology in 2007, the MS degree in Computer Architecture from the Fuzhou University in 2010. He is currently pursuing the Ph.D. degree from the Feng Chia University. His current research interests include multimedia security, image processing, and Internet of thing.

Efficient Anonymous Ciphertext-Policy Attribute-based Encryption for General Structures Supporting Leakage-Resilience

Xiaoxu Gao and Leyou Zhang
(Corresponding author: Xiaoxu Gao)
School of Mathematics and Statistics, Xidian University
Xi'an, Shaanxi 710071, China
(Email: gxx_xidian@163.com)
(Received Mar. 5, 2019; Revised and Accepted Nov. 6, 2019; First Online Jan. 23, 2020)

Abstract

The traditional cryptographic schemes cannot guarantee data security and users privacy under the side-channel attacks. Additionally, most of the existing leakage-resilient schemes cannot protect the privacy of the receivers. To achieve the leakage resilience and privacy-preserving, two anonymous attribute-based encryption (ABE) schemes for general access structures are proposed. In the first scheme, the access structure is encoded as minimal sets which provide the higher efficiency in the cost of decryption algorithm. Then we show how to obtain an anonymous leakage-resilient ABE for non-monotone access structures. Both schemes can tolerate the continual leakage when an update algorithm is employed in the event of the occurrence of the leakage information beyond the allowable leakage bound. They are proven to be adaptively secure in the standard model under four static assumptions over composite order bilinear group. The performance analyses confirm efficiency of our schemes.

Keywords: Anonymous; Attribute-based Encryption; Ciphertext-Policy; Leakage-Resilience

1 Introduction

In traditional encryption schemes, security is based on an idealized assumption that the adversary cannot get any information about the private keys and internal state. However, the practice shows this assumption is quite invalid. Many cryptographic schemes are vulnerable to side-channel attacks, where an adversary can learn meaningful information about a system by using some of the physical information that the algorithm outputs, such as running time, power consumption, and fault detection etc.. In order to characterize the leaked information that the adversary knows in the system better, various leakage models are presented. Some of them are motivated by practical issues, while others are for theoretical needs.

- The only computation leaks information model was proposed by Micali [14]. It requires that the leak only occurs in the memory part of the system executing the calculation, and the memory part that does not participate in the calculation is not leaked. The reason given in [14] is as "data can be placed in some form of storage where, when not being accessed and computed upon, it is totally secure."

- The relative leakage model (also named memory-attacks model) was proposed by Alwen *et al.* [1] to deal with cold boot attacks where the part not involved in the operation also leaked information. In this model, the leakage amount is bounded by a predetermined value.

- The bounded retrieval model is a model that stronger than the relative leakage model [2, 21, 25, 27, 29]. In this model, the leakage parameter l is an arbitrary and independent parameter of the system, and secret keys can be increased to allow l bits of leakage without affecting the size of public keys.

- The continuous leakage model [30] was put forward to solve the situation that the leakage bits exceed the predetermined number in which the leakage is unbounded in the lifecycle of the system, but it is bounded between consecutive updates.

- The auxiliary input model was presented by Dodis [8] which required that polynomial time adversary cannot recover sk from $f(sk)$ with negligible probability. Meanwhile, Yuen *et al.* [22] proposed a model which combined the concepts of the auxiliary inputs and continual memory leakage. The scheme [28] also comes from this model.

While the ABE schemes constructed in the above leakage model cannot achieve anonymity except [25, 27]. Additionally, the number of leakage bits in [25] is bounded

and the performance of [27] is inefficient because its decryption time depends not only on the leakage parameter but also on the number of attributes. Hence, it is natural to ask whether there is an efficient anonymous leakage-resilient ciphertext-policy attribute-based encryption scheme resilient to the continual leakage.

1.1 Related Work

ABE [18] was regarded as a highly promising public key primitive for realizing scalable and fine-grained access control systems. Goyal *et al.* [9] formulated the idea of ABE and presented key-policy ABE (KP-ABE). Then Bethencourt *et al.* [4] put forward to the ciphertext-policy ABE (CP-ABE). In KP-ABE, the private keys are associated with access policies and ciphertexts are labeled with sets of attributes. While in CP-ABE, ciphertexts are associated with access policies and private keys are labeled with sets of attributes. ABE has become a research hotspot because it implements one-to-many encryption. Following this trend, many schemes have been proposed [5, 7, 13, 26].

Schemes [4,9] adopted the access tree which is a monotone access structure. A KP-ABE scheme can handle non-monotone access structures over attributes where the access structures can be a boolean formula involving AND, OR, NOT, and threshold operations was proposed by R. Ostrovsky *et al.* [16]. Lewko *et al.* [12] first put forward a CP-ABE scheme and a KP-ABE scheme where the access structures were monotone span program (MSP). Both schemes are proven to be fully secure under Decisional Subgroup assumptions in the standard model over composite order bilinear group. Subsequently, Okamoto and Takashima [15] brought forward a KP-ABE and a CP-ABE for non-monotone access structures which were shown to be fully secure under a standard assumption, the Decisional Linear (DLIN) assumption, in the standard model over prime order bilinear group. Then Waters [19] proposed three efficient selectively secure CP-ABE constructions by employing MSP to express access structure in the standard model under Decisional Parallel Bilinear Diffie-Hellman Exponent (PBDHE) assumption. Lately, Attrapadung *et al.* [3] introduced a KP-ABE scheme for non-monotone access structures with constant size ciphertexts, which was selectively secure under Decisional q-parallel Bilinear Diffie-Hellman Exponent (q-DBDHE) assumption in the standard model over prime order bilinear group. This scheme also adopted the method of Ostrovsky *et al.* [16] to convert the non-monotone access structures to monotone access structures with negative attributes. Based on the fact that there are some monotone access structures for which the size of MSP is at least the number of minimal sets, while the number of minimal sets is constant. Pandit and Barua [17] put forward an ABE which used minimal sets to describe general access structures. They also constructed a corresponding hierarchical (H)KP-ABE scheme. All of the schemes achieve full security in the standard model over composite order bilinear group.

Though ABE can be directly applied to design secure access control, there is an increasing need to protect user's privacy in access control systems. In order to address the problem, the concept of anonymous ABE was introduced in schemes [10,11]. More related works are referred to [23, 24, 27]. In the anonymous CP-ABE, ciphertexts can not reveal the information of corresponding attributes in the access policies. A user obtains his/her secret keys if the corresponding attribute sets satisfy the access structure embedded in the ciphertexts. The user cannot decrypt and guess what access policy was specified by the data owner. As we know, there is no efficient anonymous CP-ABE can achieve constant size ciphertexts and adaptive leakage-resilient security in the standard model.

1.2 Our Contributions

Based on the works of [30] and [6], we put forward efficient anonymous leakage-resilient CP-ABE schemes for general access structures, which has better leakage rate and adaptive security under the four static assumptions in the standard model over composite order bilinear group. In addition, the proposed schemes were built on the relative leakage model, and implicitly used an update algorithm to tolerate continual leakage on the private keys. In the security proof, we use the dual system encryption [20], where we extend the semi-functional keys into two types: truly semi-functional and nominally semi-functional. Normal keys and nominally semi-functional keys can decrypt normal ciphertexts and semi-functional ciphertexts, but truly semi-functional keys cannot decrypt the challenge semi-functional ciphertexts. In addition, the method to prove the indistinguishability of nominally semi-functional and truly semi-functional is similar to [30], so we omitted it in the paper. The access structure used in the first construction is constructed by minimal sets with multi-valued attributes which provides the ability to fast decryption.

1.3 Organizations

The rest of paper is organized as follows. In Section 2, some preliminaries are given. Section 3 gives the definition of leakage resilience of ABE. The security definition is presented in Section 4. The construction of scheme and anonymity, performance, and efficiency analysis are given in Section 5. And security proof is introduced in Section 6. In Section 7, we give an anonymous leakage-resilient CP-ABE scheme for non-monotone access structures. Finally, we conclude this paper in Section 8.

2 Preliminaries

2.1 Notations

1) Angle brackets $\langle \cdot, \cdot \rangle$ denotes two vectors inner product, and parentheses $(\cdot, \cdot, \cdot)$ denotes vectors. The dot

product of vectors is denoted by '$\cdot$' and component-wise multiplication is denoted by '$*$'.

2) The fact that χ is picked uniformly at random from a finite set Ω is denoted by $\chi \xleftarrow{\$} \Omega$, and that all ψ, ω, ζ are picked independently and uniformly at random from Ω is denoted by $\psi, \omega, \zeta \xleftarrow{\$} \Omega$.

3) Let $\vec{\rho} = (\rho_1, \rho_2, ..., \rho_n), \vec{\sigma} = (\sigma_1, \sigma_2, ..., \sigma_n)$, $g^{\vec{\rho}}$ denote the vector of group element $g^{\vec{\rho}} = (g^{\rho_1}, g^{\rho_2}, ..., g^{\rho_n})$, the inner product vectors $\vec{\rho}$ and vector $\vec{\sigma}$ is denoted by $\langle \vec{\rho}, \vec{\sigma} \rangle$ and the bilinear group inner product is denoted by $\hat{e}_n(g^{\vec{\rho}}, g^{\vec{\sigma}})$. *i.e.*, $\langle \vec{\rho}, \vec{\sigma} \rangle = \sum_{i \in [n]} \rho_i \sigma_i$, and $\hat{e}_n(g^{\vec{\rho}}, g^{\vec{\sigma}}) = \prod_{i \in [n]} \hat{e}(g^{\rho_i}, g^{\sigma_i}) = \hat{e}(g, g)^{\langle \vec{\rho}, \vec{\sigma} \rangle}$.

4) A negligible function of λ is denoted by $negl(\lambda)$.

5) $\{1, 2, ..., n\}$ is denoted by $[n]$.

2.2 Minimal Set and Its Critical Set

Definition 1. *Let Γ be a monotonic access structure over the set of attributes $V = \{a_1, a_2, ..., a_n\}$. If $\forall A \in \Gamma \backslash \{B\}$ where $B \in \Gamma$, we have $A \subset B$ invalid, then B is called a minimal authorized set. The collection of all minimal sets in Γ is called the basis of Γ.*

Definition 2. ***(Dual of access structure)*** *If $V \backslash A = A^c \notin \Gamma$ where $A \subset V$, then the collection of sets A is composed of the dual of access structure $\Gamma^{\perp}$ of an access structure Γ over V.*

Definition 3. ***(Critical set of minimal sets)*** *If every $Y_i \in \mathcal{H}$ contains a set $B_i \subset Y_i$, there is $|B_i| \geq 2$:*

The set B_i uniquely determines Y_i in the set $\mathcal{H}$. i.e., *no other set in $\mathcal{H}$ contains B_i.*

$\forall Z \subset B_i$, set $S_Z = \bigcup_{Y_j \in \mathcal{H}, Y_j \bigcap Z \neq \emptyset} (Y_j \backslash Z)$ does not contain any element of $\mathcal{B}$.

If (I) and (II) hold, where $\mathcal{B} = \{Y_1, Y_2, ..., Y_r\}$ is the set of minimal set of an access structure Γ, and $\mathcal{H} \in \mathcal{B}$ be a subset of minimal sets, then $\mathcal{H}$ is called a critical set of minimal sets for $\mathcal{B}$.

2.3 Complexity Assumptions

Assumption 1. Pick $g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}, g_4 \xleftarrow{\$} \mathbb{G}_{p_4}$, $T_1 \xleftarrow{\$} \mathbb{G}_{p_1 p_4}, T_2 \xleftarrow{\$} \mathbb{G}_{p_1 p_2 p_4}$ and set $E = (\mathbb{G}, g_1, g_3, g_4)$. Define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 1 to be

$$Adv^1_{\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(E, T_1) = 1] - Pr[\mathcal{A}(E, T_2) = 1]| \quad (1)$$

We say that Assumption 1 holds if for all PPT algorithm $\mathcal{A}$, $Adv^1_{\mathcal{A}}(\lambda) \leq negl(\lambda)$ holds for the security λ.

Assumption 2. Pick $g_1, U_1 \xleftarrow{\$} \mathbb{G}_{p_1}, U_2, W_2 \xleftarrow{\$} \mathbb{G}_{p_2}$, $g_3, W_3 \xleftarrow{\$} \mathbb{G}_{p_3}, g_4 \xleftarrow{\$} \mathbb{G}_{p_4}, T_1 \xleftarrow{\$} \mathbb{G}_{p_1 p_2 p_3}, T_2 \xleftarrow{\$} \mathbb{G}_{p_1 p_3}$ and set $E = (\mathbb{G}, g_1, g_3, g_4, U_1 U_2, W_2 W_3)$. Define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 2 to be

$$Adv^2_{\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(E, T_1) = 1] - Pr[\mathcal{A}(E, T_2) = 1]| \quad (2)$$

We say that Assumption 2 holds if for all PPT algorithm $\mathcal{A}$, $Adv^2_{\mathcal{A}}(\lambda) \leq negl(\lambda)$ holds for the security λ.

Assumption 3. Pick $\alpha, s, r \xleftarrow{\$} \mathbb{Z}_N, g_1 \xleftarrow{\$} \mathbb{G}_{p_1}, g_4 \xleftarrow{\$} \mathbb{G}_{p_4}$, $U_2, W_2, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}, T_1 = \hat{e}(g_1, g_1)^{\alpha s}, T_2 \xleftarrow{\$} \mathbb{G}_T$, and set $E = (\mathbb{G}, g_1, g_2, g_3, g_4, g_2^r, U_2^r, g_1^{\alpha} U_2, g_1^s W_2)$. Define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 3 to be

$$Adv^3_{\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(E, T_1) = 1] - Pr[\mathcal{A}(E, T_2) = 1]| \quad (3)$$

We say that Assumption 3 holds if for all PPT algorithm $\mathcal{A}$, $Adv^3_{\mathcal{A}}(\lambda) \leq negl(\lambda)$ holds for the security λ.

Assumption 4. Pick $s, \hat{r}, \hat{s} \xleftarrow{\$} \mathbb{Z}_N$, $g_1, U_1 \xleftarrow{\$} \mathbb{G}_{p_1}$, $g_4, U_4 \xleftarrow{\$} \mathbb{G}_{p_4}, U_2, W_2, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, g_3 \xleftarrow{\$} \mathbb{G}_{p_3}, W_{24}, D_{24} \xleftarrow{\$} \mathbb{G}_{p_2 p_4}, T_1 \xleftarrow{\$} U_1^s D_{24}, T_2 \xleftarrow{\$} \mathbb{G}_{p_1 p_2 p_4}$, and set $E = (\mathbb{G}, g_1, g_2, g_3, g_4, U_1 U_4, U_1^{\hat{r}} U_2, g_1^{\hat{r}} W_2, g_1^s W_{24}, U_1 g_3^{\hat{s}})$. Define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 4 to be

$$Adv^4_{\mathcal{A}}(\lambda) = |Pr[\mathcal{A}(E, T_1) = 1] - Pr[\mathcal{A}(E, T_2) = 1]| \quad (4)$$

We say that Assumption 4 holds if for all PPT algorithm $\mathcal{A}$, $Adv^4_{\mathcal{A}}(\lambda) \leq negl(\lambda)$ holds for the security λ.

3 Leakage Resilience of CP-ABE

A CP-ABE scheme with continual leakage model is composed of the following five algorithms:

Setup$((\lambda, V, l) \rightarrow (PK, MSK))$**:** The setup algorithm takes a security parameter λ, a description of attribute universe set V and a leakage bound l as input. It outputs system public keys PK and master secret keys MSK.

KeyGen$((PK, MSK, S) \rightarrow SK_S)$**:** The key generation algorithm inputs the public keys PK, master secret keys MSK and an attribute set S, returns secret keys SK_S.

UpdateUSK$((PK, S, SK_S) \rightarrow SK'_S)$**:** On input the public keys PK, a set of attributes S and the secret keys SK_S, it outputs re-randomized secret keys SK'_S.

Encrypt$((PK, M, \Gamma) \rightarrow CT_\Gamma)$**:** The encryption algorithm takes the public keys PK, a message M, and an access structure Γ over the universe of attributes as input, and outputs ciphertexts CT_Γ such that only users whose attribute sets satisfy the access structure Γ should be able to extract M.

Decrypt$((PK, CT_\Gamma, SK_S) \rightarrow M)$**:** The algorithm takes the public keys PK, ciphertexts CT_Γ and secret keys SK_S as input, outputs the message M if and only if the attribute set S of key SK_S satisfies the access structure Γ.

4 Security Definition

For key leakage attacks, we provide a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ to achieve an anonymous leakage-resilient ciphertext-policy attribute-based encryption scheme. The security parameter and the upper bound of leakage are denoted by λ and l respectively.

Setup: The challenger $\mathcal{C}$ runs the setup algorithm to generate the public keys and master keys (PK, MSK), and sends the public keys to the adversary $\mathcal{A}$ while keeps the master secret keys. At the same time, $\mathcal{C}$ creates two initial empty lists: $\mathcal{Q} = (hd, S, SK_S, L_{SK})$, $\mathcal{R} = (hd, S)$ to store records, where all records are associated with a handle hd and L_{SK} means total leakage bits.

Phase 1: In this stage, $\mathcal{A}$ can adaptively perform the following queries:

- *Key Generation queries*: $\mathcal{A}$ submits the attribute set S to $\mathcal{C}$, and $\mathcal{C}$ runs the key generation algorithm to generate the private keys SK_S. Challenger sets $hd = hd + 1$, then adds $(hd, S, SK_S, 0)$ to the set $\mathcal{Q}$. In this query, $\mathcal{C}$ only gives $\mathcal{A}$ hd of the generated keys rather than the concrete keys itself.
- *Leakage queries*: $\mathcal{A}$ gives a polynomial-time computable arbitrary function $f : \{0,1\}^\star \rightarrow \{0,1\}^\star$ with a queried handle hd of the keys to $\mathcal{C}$. $\mathcal{C}$ finds the tuple (hd, S, SK_S, L_{SK}) and checks if $L_{SK} + |f(SK)| \leq l$ established. If this is true, it returns $f(SK)$ to $\mathcal{A}$ and updates L_{SK} with $L_{SK} + |f(SK)|$. If the check fails, it returns $\perp$ to the $\mathcal{A}$.
- *Reveal queries*: $\mathcal{A}$ gives the handle hd for a specified key SK_S to $\mathcal{C}$. The $\mathcal{C}$ scans $\mathcal{Q}$ to find the requested entry and returns the private keys SK_S to $\mathcal{A}$. Then $\mathcal{C}$ removes the item from the set $\mathcal{Q}$ and adds it to the set $\mathcal{R}$.
- *Update queries*: $\mathcal{A}$ issues a key update query for SK_S. $\mathcal{C}$ searches the record in $\mathcal{Q}$. If it is not found, $\mathcal{C}$ returns the keys with key generation algorithm and sets $L_{SK} = 0$. Otherwise, $\mathcal{C}$ returns with UpdateUSK algorithm and updates the corresponding $L_{SK} = 0$.

Challenge: $\mathcal{A}$ outputs two pairs of message and access structure (M_0, Γ_0), (M_1, Γ_1) to $\mathcal{C}$, where for every $S \in \mathcal{R}$, neither satisfies Γ_0 nor satisfies Γ_1. With the restriction that the length of the message M_0 equals to the length of the message M_1, $\mathcal{C}$ selects $b \in \{0,1\}$ randomly and encrypts the message M_b under the access structure Γ_b, and sends the resulting ciphertexts to $\mathcal{A}$.

Phase 2: This phase is the same as Phase 1 with the additional restrictions that reveal queries and update queries can be performed, and cannot execute leakage queries. The attributes of the query do not satisfy the challenge access structure.

Guess: $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ and wins the game if $b' = b$.

We say that an anonymous attribute-based encryption scheme is l leakage-resilient and adaptively secure against chosen plaintext attacks (ANON-IND-CPA) if for all polynomial time adaptive adversaries $\mathcal{A}$, the advantage of $\mathcal{A}$ in the above mentioned game is negligible, where the advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}}^{ANON-IND-CPA}(\lambda, l) = |Pr[b' = b] - \frac{1}{2}|$.

5 Anonymous Leakage-Resilient CP-ABE for MAS

5.1 Concrete Construction

Our scheme relies on a composite order bilinear group where its order is $N = p_1p_2p_3p_4$, and p_1, p_2, p_3, p_4 are distinct primes. The main system is built in $\mathbb{G}_{p_1}$ subgroup, while the subgroup $\mathbb{G}_{p_2}$ acts as the semi-functional space. The subgroup $\mathbb{G}_{p_3}$ provides the additional randomness on keys to isolate keys in our hybrid games. $\mathbb{G}_{p_4}$ will make the scheme achieve anonymity. Then we would extend the composite order group to multiple dimensional to tolerate the possible leakage.

Let $V = \{attr_1, attr_2, ..., attr_n\}$ be a set of attributes. Each attribute contains n_i possible values and $v_{i,j}$ represents the jth value of $attri$, and $I \subset \{1, 2, ..., n\}$ is the attribute name index.

Setup$((\lambda, V, l) \rightarrow (PK, MSK))$**:** The setup algorithm takes as input a security parameter λ, the attribute universe description V and a leakage upper bound l. Then the algorithm generates the public keys and master secret keys as follows. Run the bilinear group generator to produce $\Phi = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \hat{e})$. Define $negl = p_2^{-\tau}$ as the allowable maximum probability in succeeding in leakage guess and compute $\omega = \lceil 1 + 2\tau + \frac{l}{\log p_2} \rceil$, where τ is a positive constant. In practice, $\omega \approx \lceil 1 + \frac{l}{\log p_2} \rceil$. Select $g_1, X_1 \in \mathbb{G}_{p_1}$, $g_3 \in \mathbb{G}_{p_3}$, $g_4, X_4 \in \mathbb{G}_{p_4}$, $\alpha \in \mathbb{Z}_N$, $\vec{\rho} \in \mathbb{Z}_N^\omega$ randomly. For each $i \in [n], j \in [n_i]$, choose random values $t_{i,j} \in \mathbb{Z}_N$ and set the public keys as follows.

$$PK = (N, g_1, g_4, g_1^{\vec{\rho}}, y, Y, T_{i,j}; \forall i \in [n], j \in [n_i]),$$

where

$$y = \hat{e}(g_1, g_1)^{\alpha}, Y = X_1 X_4, T_{i,j} = g^{t_{i,j}}.$$

The master secret keys are

$$MSK = (X_1, g_3, \alpha).$$

KeyGen$((PK, MSK, S) \rightarrow SK_S)$**:** This algorithm takes PK, MSK and an attribute set $S = \{v_{1,x_1}, v_{2,x_2}, ..., v_{n',x'_n}\}$ as input, where $n' \leq n, 1 \leq x_i \leq n_i$ for each $1 \leq i \leq n'$. Then the algorithm chooses random values $t, y_2, y_3 \in \mathbb{Z}_N, \vec{y}_1, \vec{\sigma} \in \mathbb{Z}_N^{\omega}$, and $y_{i,j} \in \mathbb{Z}_N$ for $v_{i,j} \in S$, and outputs the secret keys as follows.

$$SK_S = (S, \vec{K}_1, K_2, K_3, K_{i,j}; \forall v_{i,j} \in S),$$

in which

$$\vec{K}_1 = g_1^{\vec{\sigma}} * g_3^{\vec{y}_1}, \qquad K_3 = g_1^t g_3^{y_3},$$
$$K_2 = g_1^{\alpha + \langle \vec{\rho}, \vec{\sigma} \rangle} X_1^t g_3^{y_2}, \qquad K_{i,j} = T_{i,j}^t g_3^{y_{i,j}}.$$

UpdateUSK$((PK, S, SK_S) \rightarrow SK'_S)$**:** The key update algorithm selects $\Delta t, \Delta y_2, \Delta y_3 \in \mathbb{Z}_N, \Delta \vec{y}_1, \Delta \vec{\sigma} \in \mathbb{Z}_N^{\omega}$ randomly, picks $\Delta y_{i,j} \in \mathbb{Z}_N$ for $v_{i,j} \in S$ at random, and outputs a new key SK'_S:

$$SK'_S = (S, \vec{K}'_1, K'_2, K'_3, K'_{i,j}; \forall v_{i,j} \in S),$$

where

$$\vec{K}'_1 = \vec{K}_1 * g_1^{\Delta \vec{\sigma}} * g_3^{\Delta \vec{y}_1}, \qquad K'_3 = K_3 g_1^{\Delta t} g_3^{\Delta y_3},$$
$$K'_2 = K_2 g_1^{\langle \vec{\rho}, \Delta \vec{\sigma} \rangle} X_1^{\Delta t} g_3^{\Delta y_2}, \qquad K'_{i,j} = K_{i,j} T_{i,j}^{\Delta t} g_3^{\Delta y_{i,j}}.$$

Encrypt$((PK, M, \Gamma) \rightarrow CT_{\Gamma})$**:** At first, this algorithm converts the monotonic access structure Γ to the set of minimal sets $\mathcal{B} = \{B_1, B_2, ..., B_{\tilde{m}}\}$, where $B_k (k \in [\tilde{m}])$ is a set of attribute values. It selects $s, s_1, s_2, ..., s_{\tilde{m}} \in \mathbb{Z}_N, \vec{d} \in \mathbb{Z}_N^{\omega}, W_k, V_k \in \mathbb{G}_{p_4} (k \in [\tilde{m}])$ randomly, then outputs the resulting ciphertexts CT_{Γ} and the index $I_{B_k} \subset \{1, 2, ..., n\} (k \in [\tilde{m}])$ corresponding to attribute $B_k (k \in [\tilde{m}])$.

$$CT_{\Gamma} = (\{I_{B_k}\}_{k \in [\tilde{m}]}, C_0, \vec{C}_1, C_2, \vec{C}_3, \vec{C}_4),$$

where

$$C_0 = M y^s, \qquad \vec{C}_1 = g_1^{s\vec{\rho}} * g_4^{\vec{d}},$$
$$C_2 = g_1^s g_4, \qquad \vec{C}_4 = (C_{4,k})_{k \in [\tilde{m}]} = (g_1^{s_k} V_k)_{k \in [\tilde{m}]},$$
$$\vec{C}_3 = \{C_{3,k}\}_{k \in [\tilde{m}]} = \{Y^s (\prod_{v_{i,j} \in B_k} T_{i,j})^{s_k} W_k\}_{k \in [\tilde{m}]}.$$

Decrypt$((PK, CT_{\Gamma}, SK_S) \rightarrow M)$**:** If the attributes set S satisfies the access structure specified by $\mathcal{B}$, then S must be a superset of a minimal set in $\mathcal{B}$. Let $B_k \subset S$ for some $k \in [\tilde{m}]$, this algorithm calculates

$$M = \frac{C_0 \cdot \hat{e}_{\omega}(\vec{C}_1, \vec{K}_1) \hat{e}(C_{3,k}, K_3)}{\hat{e}(C_2, K_2) \hat{e}(C_{4,k}, \prod_{v_{i,j} \in B_k} K_{i,j})}$$

5.2 Correctness

The correctness can be checked by applying the orthogonality of $\mathbb{G}_{p_i}$, where $i = 1, 2, 3, 4$. If the attributes set S satisfies the access structure specified by $\mathcal{B}$, then one can obtain the below equations hold.

$$\begin{aligned}
\hat{e}_{\omega}(\vec{C}_1, \vec{K}_1) &= \hat{e}_{\omega}(g_1^{s\vec{\rho}} * g_4^{\vec{d}}, g_1^{\vec{\sigma}} * g_3^{\vec{y}_1}) \\
&= \hat{e}_{\omega}(g_1^{s\vec{\rho}}, g_1^{\vec{\sigma}}) \\
&= \hat{e}(g_1, g_1)^{s\langle \vec{\rho}, \vec{\sigma} \rangle} \\
\hat{e}(C_2, K_2) &= \hat{e}(g_1^s g_4, g_1^{\alpha + \langle \vec{\rho}, \vec{\sigma} \rangle} X_1^t g_3^{y_2}) \\
&= \hat{e}(g_1, g_1)^{\alpha s + s\langle \vec{\rho}, \vec{\sigma} \rangle} \hat{e}(g_1, X_1)^{st} \\
\hat{e}(C_{3,k}, K_3) &= \hat{e}(Y^s (\prod_{v_{i,j} \in B_k} T_{i,j})^{s_k} W_k, g_1^t g_3^{y_3}) \\
&= \hat{e}(g_1, Y)^{st} \hat{e}(\prod_{v_{i,j} \in B_k} T_{i,j}, g_1)^{s_k t} \\
\hat{e}(C_{4,k}, \prod_{v_{i,j} \in B_k} K_{i,j}) &= \hat{e}(g_1^{s_k} V_k, \prod_{v_{i,j} \in B_k} T_{i,j}^t g_3^{y_{i,j}}) \\
&= \hat{e}(g_1, \prod_{v_{i,j} \in B_k} T_{i,j})^{s_k t}
\end{aligned}$$

5.3 Anonymity Analysis

This section will show that the proposed scheme achieves the anonymity over the composite order bilinear group. Compared with scheme [30], our scheme adds some random elements in $\mathbb{G}_{p_4}$ to each part of the ciphertexts. These random elements will not make an effect on the decryption process. However, they are necessary for anonymity of the scheme. Because if there is no such elements, for some minimal sets B_k^*, the adversary may determine whether the ciphertext component $C_{3,k}$ of the ciphertext $\vec{C}_3$ is encrypted under B_k^* or not. In our scheme, by utilizing the DDH-test $\hat{e}(C_{3,k}, g_1) \stackrel{?}{=} \hat{e}(Y, C_2) \hat{e}(\prod_{v_{i,j^*} \in B_k^*} T_{i,j^*}, C_{4,k})$ to determine whether the ciphertext component $C_{3,k}$ is encrypted under the B_k^* or not. The DDH-test $\hat{e}(C_{3,k}, g_1) \stackrel{?}{=} \hat{e}(Y, C_2) \hat{e}(\prod_{v_{i,j^*} \in B_k^*} T_{i,j^*}, C_{4,k})$ is the same as $\frac{\hat{e}(C_{3,k}, g_1)}{\hat{e}(Y, C_2) \hat{e}(\prod_{v_{i,j^*} \in B_k^*} T_{i,j^*}, C_{4,k})} \stackrel{?}{=} 1$. The followings are the detailed analyses.

$$\begin{aligned}
\hat{e}(C_{3,k}, g_1) &= \hat{e}(Y^s (\prod_{v_{i,j} \in B_k} T_{i,j})^{s_k} W_k, g_1) \\
&= \hat{e}(Y^s, g_1) \hat{e}((\prod_{v_{i,j} \in B_k} T_{i,j})^{s_k}, g_1) \\
&= \hat{e}(Y, g_1)^s \hat{e}(\prod_{v_{i,j} \in B_k} T_{i,j}, g_1)^{s_k} \\
\hat{e}(Y, C_2) &= \hat{e}(Y, g_1)^s \hat{e}(Y, g_4)
\end{aligned}$$

$$\begin{aligned}
\hat{e}(\prod_{v_{i,j^*}\in B_k^*} T_{i,j^*}, C_{4,k}) &= \hat{e}(\prod_{v_{i,j^*}\in B_k^*} T_{i,j^*}, g_1^{s_k} V_k) \\
&= \hat{e}(\prod_{v_{i,j^*}\in B_k^*} T_{i,j^*}, g_1^{s_k}) \\
&= \hat{e}(\prod_{v_{i,j^*}\in B_k^*} T_{i,j^*}, g_1)^{s_k} \\
\frac{\hat{e}(C_{3,k}, g_1)}{\hat{e}(Y, C_2)\hat{e}(\prod_{v_{i,j^*}\in B_k^*} T_{i,j^*}, C_{4,k})} &= \frac{\hat{e}(\prod_{v_{i,j}\in B_k} T_{i,j}, g_1)^{s_k}}{\hat{e}(Y, g_4)\hat{e}(\prod_{v_{i,j^*}\in B_k^*} T_{i,j^*}, g_1)^{s_k}}
\end{aligned}$$

If $B_k = B_k^*$, then $v_{i,j} = v_{i,j^*}$ for all i, $1 \le i \le n' \le n$. Therefore $\frac{\hat{e}(C_{3,k}, g_1)}{\hat{e}(Y, C_2)\hat{e}(\prod_{v_{i,j^*}\in B_k^*} T_{i,j^*}, C_{4,k})} = \frac{1}{\hat{e}(Y, g_4)}$.

If $B_k \neq B_k^*$, then there exists at last one k', where $1 \le k' \le n' \le n$ such that $v_{k',j} \neq v_{k',j^*}$. Without loss of generality, let $v_{k',j} = v_{k',j^*}$, for all $1 \le i \le n' \le n$ except $i = k'$. Then $t_{i,j} = t_{i,j^*}$. Therefore $\frac{\hat{e}(C_{3,k}, g_1)}{\hat{e}(Y, C_2)\hat{e}(\prod_{v_{i,j^*}\in B_k^*} T_{i,j^*}, C_{4,k})} = \frac{\hat{e}(T_{k',j}, g_1)^{s_k}}{\hat{e}(Y, g_4)\hat{e}(T_{k',j^*}, g_1)^{s_k}}$.

In both cases, $B_k = B_k^*$ and $B_k \neq B_k^*$, the DDH-test gives a random element of $\mathbb{G}_T$ so that the adversary will be not able to determine whether the component $C_{3,k}$ of the ciphertext $\vec{C_3}$ is encrypted under the B_k^* or not. So the access structure is hidden, and our scheme is anonymous.

5.4 Performance Analysis

As shown in Table 1, we give the performance comparisons of schemes [21,27,30] and the proposed scheme in the access policy, leakage model, support multi-functionality and anonymity. All these schemes are constructed under the key leakage model, in which the access structure of [21,27] is denoted by the linear secret sharing (LSSS), while that of [30] and the proposed scheme are represented by the minimal sets. In addition, it can be found that [21,30] do not support anonymity and scheme [21,27] do not support multi-show functionality. However, our scheme achieves the anonymity and attribute multi-show ability simultaneously.

5.5 Efficiency Analysis

We present the performance evaluation based on our DMA implementation prototype. Our experiment is implemented on Pairing-Based Cryptography (PBC) library to implement the scheme. We will compare the computational efficiency of our scheme with scheme [21,27,30]. In the Figures 1, 2 and 3, the leakage parameter is set to be $\omega = 5$.

Figure 1 shows the comparison of key generation time with different number of attributes, where the number of attribute changes from 10 to 50.

Figure 2 presents the comparison of update time with different number of attributes.

Figure 3 provides the comparison of encryption time with different number of minimal sets, where the number of minimal sets varies from 5 to 25.

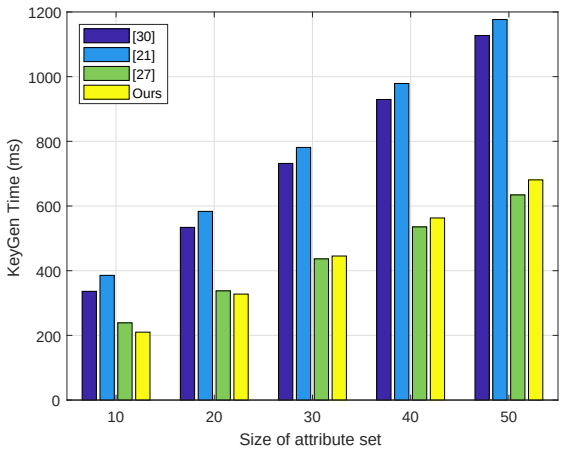


Figure 1: KeyGen time with different number of attributes

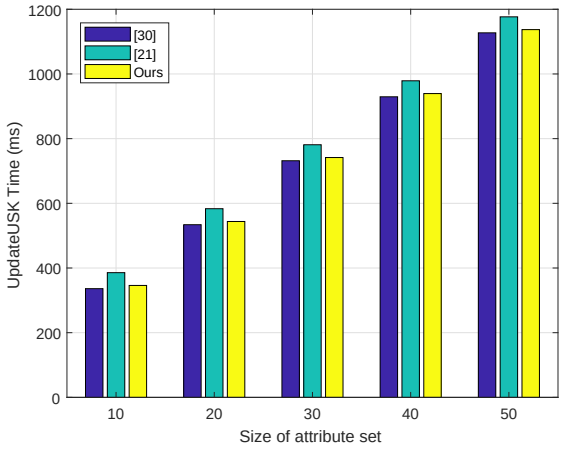


Figure 2: UpdateUSK time with different number of attributes

Figure 4 gives the comparison of decryption time with different leakage parameters, where the leakage parameter changes from 5 to 25.

From the analysis of the experimental results, we can see that our scheme has advantages over [27] when the number of attributes is between 10 and 20. While, compared with [21,30], the proposed scheme spends less time. Moreover, the proposed scheme has obvious advantage over [27] in decryption. In summary, our scheme is quite practical and efficient.

6 Security Proof

In the proof, we generate normal private keys and ciphertexts which are used in the real scheme. Then we generate semi-functional keys and ciphertexts which are used in the proofs. They are shown as follows.

- **KeyGenSF**. $SK_S = (S, \vec{K}_1, K_2, K_3, K_{i,j}; \forall v_{i,j} \in S)$ be the normal keys. The semi-functional keys are as follows.

Table 1: Performance analysis

Scheme	Access policy	Leakage Model	Multi-show attr	Anonymity
[21]	LSSS	Continual leakage	No	No
[30]	Minimal Sets	Continual leakage	Yes	No
[27]	LSSS	Bounded leakage	Not	Yes
Ours	Minimal Sets	Continual leakage	Yes	Yes

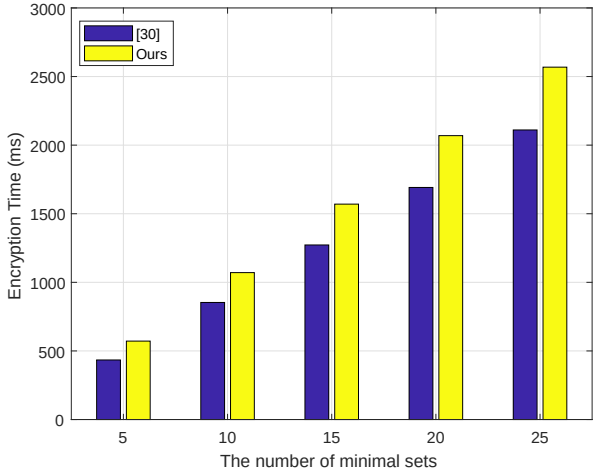


Figure 3: Encryption time with different number of minimal sets

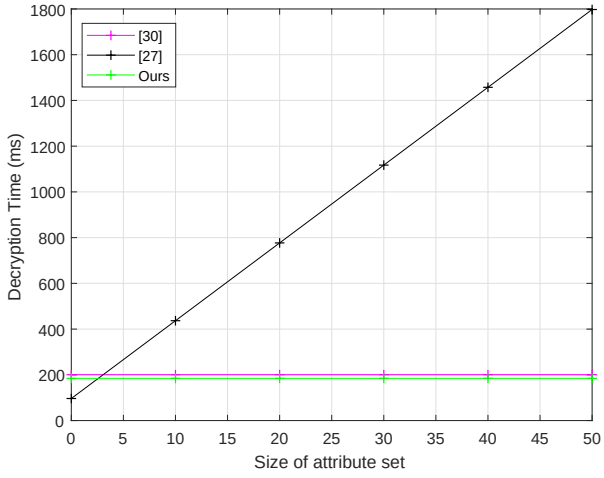


Figure 4: Decryption time with different number of attributes

  - Type 1: $S\bar{K}_S = (S, \vec{K}_1 * g_2^{\vec{d}_1}, K_2 g_2^{d_2}, K_3 g_2^{d_3}, K_{i,j} g_2^{d_{i,j}}; \forall v_{i,j} \in S)$, where g_2 is a generator of group $\mathbb{G}_{p_2}$ and $\vec{d}_1, d_2, d_3, d_{i,j}$ are random elements in $\mathbb{Z}_N$.
  - Type 2: $S\bar{K}_S = (S, \vec{K}_1, K_2 g_2^{d_2}, K_3, K_{i,j}; \forall v_{i,j} \in S)$

- **EncSF**. Let $CT_\Gamma = (\{I_{B_k}\}_{k\in[\tilde{m}]}, C_0, \vec{C}_1, C_2, \vec{C}_3, \vec{C}_4)$ be a normal ciphertext. The semi-functional ciphertexts are converted as: $C\bar{T}_\Gamma = (\{I_{B_k}\}_{k\in[\tilde{m}]}, C_0, \vec{C}_1 * g_2^{\vec{e}_1}, C_2 g_2^{e_2}, \vec{C}_3 * g_2^{\vec{e}_3}, \vec{C}_4)$, where $\vec{e}_1, e_2, \vec{e}_3$ are random elements in $\mathbb{Z}_N$.

If we use the Type 1 of semi-functional keys to decrypt a semi-functional ciphertext, we will obtain extra term $\hat{e}(g_2, g_2)^{\langle \vec{d}_1, \vec{e}_1\rangle - d_2 e_2 + d_3 e_{3,k}}$. If $\langle \vec{d}_1, \vec{e}_1\rangle - d_2 e_2 + d_3 e_{3,k} = 0$, we can call it a nominally semi-functional key, otherwise it is truly semi-functional.

The proof uses a series of indistinguishable games to prove the indistinguishability between in the $Game_{real}$ and $Game_{final1}$. There will be $2Q+4$ games between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, where Q is the number of key queries times and k' is from 0 to Q. The concrete definition of games is as follows.

$Game_{real}$: This is the real anonymous ABE security game, where all private keys and the ciphertexts are in normal form.

$Game_0$: All private keys are in normal form, and the challenge ciphertexts are in semi-functional form.

$Game_{k',1}$: The challenge ciphertexts are semi-functional. The first $k'-1$ keys are semi-functional of Type 2, and the k'^{th} key is the semi-functional form of Type 1. The remaining keys are normal.

$Game_{k',2}$: The challenge ciphertexts are semi-functional. The first k' keys are semi-functional of Type 2, and the remaining keys are normal.

$Game_{final0}$: All private keys are the Type 2 of semi-functional keys. And the challenge ciphertexts are semi-functional where C_0 is random in group $\mathbb{G}_T$.

$Game_{final1}$: It is same as $Game_{final0}$ except that the $\vec{C}_3$ is random in group $\mathbb{G}_{p_1 p_2 p_4}$.

We can see that in $Game_{Q,2}$, all of the keys are semi-functional. And in the last game, the adversary has no advantage.

Lemma 1. *Suppose there exists a PPT adversary $\mathcal{A}$ who can distinguish $Game_{real}$ and $Game_0$ with the non-negligible advantage ϵ, then there is a PPT algorithm $\mathcal{B}$ with the advantage ϵ in breaking Assumption 1.*

Proof. $\mathcal{B}$ receives $E = (\mathbb{G}, g_1, g_3, g_4)$ from the challenger $\mathcal{C}$ and simulates $Game_{real}$ or $Game_0$ with $\mathcal{A}$ depending on whether $T \xleftarrow{\$} \mathbb{G}_{p_1 p_4}$ or $T \xleftarrow{\$} \mathbb{G}_{p_1 p_2 p_4}$. □

Setup: $\mathcal{B}$ takes the security parameter λ and leakage upper bound l as input and outputs the description of group: $\Phi = (N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \hat{e})$. Then $\mathcal{B}$ generates the public keys as follows. Set $\omega = \lceil 1 + \frac{l}{logp_2} \rceil$. Select $\alpha, a, b, t_{i,j} \in \mathbb{Z}_N$ randomly, and set $Y = X_1X_4 = g_1^a g_4^b$. Choose $\vec{\rho} \in \mathbb{Z}_N^\omega$ randomly and generate $PK = (N, g_1, g_4, \hat{e}(g_1, g_1)^\alpha, Y, g_1^{\vec{\rho}}, T_{i,j} = g_1^{t_{i,j}}; \forall i \in [n], j \in [n_i])$.

Phase 1: $\mathcal{B}$ generates normal keys for attribute sets in the key generation queries (keep in mind that the private keys of $\mathcal{A}$ query are either in normal form or in semi-functional form). In addition, $\mathcal{B}$ can answer the queries of leakage, reveal and update.

Challenge: $\mathcal{A}$ sends two equal length message M_0, M_1 and access structures Γ_0, Γ_1. $\mathcal{B}$ selects random $b \in \{0, 1\}$ and encrypts M_b under the access structure Γ_b. $\mathcal{B}$ encodes the access structure as the set of minimal sets $\mathcal{B}^* = \{B_1, B_2, ..., B_{\tilde{m}}\}$, where $B_k (k \in [\tilde{m}])$ is a set of attribute values. Select random element $s_1, s_2, ..., s_{\tilde{m}} \in Z_N$ and generate the challenge ciphertexts $C\bar{T}_\Gamma = (\{I_{B_k}\}_{k\in[\tilde{m}]}, C_0 = M_b\hat{e}(g_1^\alpha, T), \vec{C}_1 = T^{\vec{\rho}} * g_4^{\vec{d}}, C_2 = Tg_4, \vec{C}_3 = \{T^a(\prod_{v_{i,j}\in B_k} T_{i,j})^{s_k} W_k\}_{k\in[\tilde{m}]}, \vec{C}_4 = (g_1^{s_k} V_k)_{k\in[\tilde{m}]})$.

Phase 2: It is similar with Phase 1. $\mathcal{B}$ can answer the queries of reveal and update with the restriction that the attribute sets of adversary queries cannot meet the challenge access structure.

Guess: The adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins the game.

If $T \xleftarrow{\$} \mathbb{G}_{p_1p_2p_4}$, set T as g_2^c. It implicitly sets the semi-functional factor of the challenge ciphertexts as $(c\vec{\rho}, c, ac\vec{1}, 0)$. In this situation, $\mathcal{B}$ simulates the $Game_0$. Otherwise, $\mathcal{B}$ simulates the $Game_{real}$.

So if $\mathcal{A}$ can distinguish two games with a non-negligible advantage ϵ, $\mathcal{B}$ can use the algorithm to break the Assumption 1 with the same advantage.

Lemma 2. *Suppose there exists a PPT adversary $\mathcal{A}$ can distinguish $Game_{k'-1,2}$ and $Game_{k',1}$ with the non-negligible advantage ϵ, then is a PPT algorithm $\mathcal{B}$ with the advantage ϵ in breaking the Assumption 2.*

Proof $\mathcal{B}$ receives $E = (\mathbb{G}, g_1, g_3, g_4, U_1U_2, W_2W_3)$ and simulates $Game_{k'-1,2}$ or $Game_{k',1}$ with $\mathcal{A}$ depending on whether $T \xleftarrow{\$} \mathbb{G}_{p_1p_2p_3}$ or $T \xleftarrow{\$} \mathbb{G}_{p_1p_3}$.

Setup: It is same as Lemma 1.

Phase 1: Because $\mathcal{B}$ knows the master keys, it can answer all private key queries.

If $i' > k'$, $\mathcal{B}$ generates normal keys.

If $i' < k'$, $\mathcal{B}$ selects $t, h, y_2, y_3 \in \mathbb{Z}_N$ and $\vec{\sigma}, \vec{y}_1 \in \mathbb{Z}_N^\omega$ randomly, and picks $y_{i,j} \in \mathbb{Z}_N$ at random for generating Type 2 of semi-functional keys.

$$\begin{aligned} S\bar{K}_S &= (S, \vec{K}_1, K_2, K_3, K_{i,j}; \forall v_{i,j} \in S) \\ &= (S, g_1^{\vec{\sigma}} * g_3^{\vec{y}_1}, g_1^{\alpha+\langle\vec{\rho},\vec{\sigma}\rangle} X_1^t (W_2W_3)^h g_3^{y_2}, g_1^t g_3^{y_3}, \\ &\quad T_{i,j}^t g_3^{y_{i,j}}; \forall v_{i,j} \in S) \end{aligned}$$

And if $i' = k'$, $\mathcal{B}$ selects $\vec{\sigma} \in \mathbb{Z}_N^\omega$ that satisfies $\langle\vec{\sigma}, \vec{\rho}\rangle = 0$, outputs the private key as follows:

$$\begin{aligned} S\bar{K}_S &= (S, \vec{K}_1, K_2, K_3, K_{i,j}; \forall v_{i,j} \in S) \\ &= (S, T^{\vec{\sigma}} * g_3^{\vec{y}_1}, g_1^\alpha T^a g_3^{y_2}, Tg_3^{y_3}, T^{t_{i,j}} g_3^{y_{i,j}}; \\ &\quad \forall v_{i,j} \in S) \end{aligned}$$

If $T \xleftarrow{\$} \mathbb{G}_{p_1p_3}$, the private key is a normal key, $\mathcal{B}$ simulates the $Game_{k'-1,2}$. If $T \xleftarrow{\$} \mathbb{G}_{p_1p_2p_3}$, the private key is a semi-functional key of Type 1. In this case, $\mathcal{B}$ simulates $Game_{k',1}$.

Set the part of T in G_{p_2} is g_2^θ, then $\vec{d}_1 = \theta\vec{\sigma}, d_2 = a\theta, d_3 = \theta$. In addition, $\mathcal{A}$ can ask the oracles of leak and update.

Challenge: It is similar with Lemma 1. Set $U_1 = g_1^s, U_2 = g_2^\xi$ and calculate the ciphertexts

$$CT_\Gamma = (\{I_{B_k}\}_{k\in[\tilde{m}]}, C_0, \vec{C}_1, C_2, \vec{C}_3, \vec{C}_4),$$

in which

$$\begin{aligned} C_0 &= M_b\hat{e}(g_1^\alpha, U_1U_2), & \vec{C}_1 &= (U_1U_2)^{\vec{\rho}} * g_4^{\vec{d}}, \\ C_2 &= (U_1U_2)g_4, & \vec{C}_4 &= (g_1^{s_k} V_k)_{k\in[\tilde{m}]}, \\ \vec{C}_3 &= \{(U_1U_2)^a (\prod_{v_{i,j}\in B_k} T_{i,j})^{s_k} W_k\}_{k\in[\tilde{m}]}. \end{aligned}$$

Phase 2: It is similar with Phase 1. $\mathcal{B}$ can answer the queries of reveal and update with the restriction that the attribute sets corresponding to any private key of the query cannot satisfy the challenge access structure.

Guess: The adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins the game.

Based on the above descriptions, we obtain the semi-functional factor of the challenge ciphertexts is $(\xi\vec{\rho}, \xi, a\xi\vec{1}, 0)$. And the equation $\langle\vec{d}_1, \vec{e}_1\rangle - d_2e_2 + d_3e_{3,k} = 0$ holds. If the attributes of k'^{th} keys satisfy the challenge access structure, it is a nominally semi-functional key. Following the Lemma 5 in [30], the leakage of the key can not help adversary detect the k'^{th} private key is normal or semi-functional.

Lemma 3. *Suppose there exists a PPT adversary $\mathcal{A}$ can distinguish $Game_{k',1}$ and $Game_{k',2}$ with the non-negligible advantage ϵ, then there is a PPT algorithm $\mathcal{B}$ with the advantage ϵ in breaking the Assumption 2.*

Proof. Unlike construction of Lemma 2, the construction of k'^{th} key is as follows.

$$\begin{aligned} S\bar{K}_S =&(S, \vec{K}_1, K_2, K_3, K_{i,j}; \forall v_{i,j} \in S) \\ =&(S, T^{\vec{\sigma}} * g_3^{\vec{y}_1}, g_1^{\alpha} T^a g_3^{y_2} (W_2 W_3)^d, T g_3^{y_3}, T^{t_{i,j}} g_3^{y_{i,j}}; \\ & \forall v_{i,j} \in S). \end{aligned}$$

□

If $T \xleftarrow{\$} \mathbb{G}_{p_1 p_3}$, this private key is a Type 2 of semi-functional key. Then $\mathcal{B}$ simulates the $Game_{k',2}$. If $T \xleftarrow{\$} \mathbb{G}_{p_1 p_2 p_3}$, this private key is the Type 1 of semi-functional key. In this case, $\mathcal{B}$ simulates $Game_{k',1}$. So if $\mathcal{A}$ can distinguish these two games with the advantage ϵ, $\mathcal{B}$ can break the Assumption 2 with the same advantage.

Lemma 4. *Suppose there is a PPT adversary $\mathcal{A}$ can distinguish $Game_{Q,2}$ and $Game_{final0}$ with the non-negligible advantage ϵ, then there exists a PPT algorithm $\mathcal{B}$ with advantage ϵ in breaking Assumption 3.*

Proof. $\mathcal{B}$ receives the instance $E = (\mathbb{G}, g_1, g_2, g_3, g_4, g_2^r, U_2^r, g_1^{\alpha} U_2, g_1^s W_2)$, simulates $Game_{Q,2}$ or $Game_{final0}$. □

Setup: $\mathcal{B}$ selects $a, b, t_{i,j} \in \mathbb{Z}_N$, and sets $Y = X_1 X_4 = g_1^a g_4^b$. Select $\vec{\rho} \in \mathbb{Z}_N^{\omega}$ randomly and generate the $PK = (N, g_1, g_4, \hat{e}(g_1^{\alpha} U_2, g_1), Y, g_1^{\vec{\rho}}, T_{i,j} = g^{t_{i,j}}; \forall i \in [n], j \in [n_i])$.

Phase 1: All of the keys generated are the Type 2 of semi-functional keys. $\mathcal{B}$ selects $\vec{y}_1, \vec{\sigma} \in \mathbb{Z}_N^{\omega}, t, y_2, y_3 \in \mathbb{Z}_N$, $y_{i,j} \in \mathbb{Z}_N$ randomly, and outputs the secret keys $S\bar{K}_S = (S, \vec{K}_1, K_2, K_3, K_{i,j}; \forall v_{i,j} \in S) = (S, g_1^{\vec{\sigma}} * g_3^{\vec{y}_1}, (g_1^{\alpha} U_2) X_1^t g_1^{\langle \vec{\rho}, \vec{\sigma} \rangle} g_3^{y_2}, g_1^t g_3^{y_3}, T_{i,j}^t g_3^{y_{i,j}}; \forall v_{i,j} \in S)$.

In addition, $\mathcal{A}$ can ask the oracles of leak and update.

Challenge: Similar to Lemma 1, $\mathcal{B}$ calculates the challenge ciphertexts $CT_{\Gamma} = (\{I_{B_k}\}_{k \in [\tilde{m}]}, C_0 = M_b T, \vec{C}_1 = (g_1^s U_2)^{\vec{\rho}} * g_4^{\vec{d}}, C_2 = (g_1^s U_2) g_4, \vec{C}_3 = \{(g_1^s U_2)^a \cdot (\prod_{v_{i,j} \in B_k} T_{i,j})^{s_k} W_k\}_{k \in [\tilde{m}]}, \vec{C}_4 = (g_1^{s_k} V_k)_{k \in [\tilde{m}]})$.

Phase 2: It is similar with Phase 1. $\mathcal{B}$ can answer the queries of reveal and update with the restriction that the attribute sets corresponding to any private key of the query cannot satisfy the challenge access structure.

Guess: The adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins the game.

Obviously, we can learn that if $T = \hat{e}(g_1, g_1)^{\alpha s}$, it is a semi-functional ciphertext of message M_b. Otherwise $T \xleftarrow{\$} \mathbb{G}_T$, it is a random element of $\mathbb{G}_T$. So if the adversary $\mathcal{A}$ can distinguish these two games, $\mathcal{B}$ can break the assumption 3 with the same advantage.

Lemma 5. *Suppose there exists a PPT adversary $\mathcal{A}$ can distinguish $Game_{final0}$ and $Game_{final1}$ with the non-negligible advantage ϵ, then there is a PPT algorithm $\mathcal{B}$ with the advantage ϵ in breaking the Assumption 4.*

Proof. $\mathcal{B}$ receives the instance $E = (\mathbb{G}, g_1, g_2, g_3, g_4, U_1 U_4, U_1^{\hat{r}} U_2, g_1^{\hat{r}} W_2, g_1^s W_{24}, U_1 g_3^{\hat{s}})$ and simulates the $Game_{final0}$ or $Game_{final1}$. □

Setup: $\mathcal{B}$ selects $t_{i,j}, \alpha \in \mathbb{Z}_N$, $\vec{\rho} \in \mathbb{Z}_N^{\omega}$ randomly, and sets $PK = (N, g_1, g_4, \hat{e}(g_1, g_1)^{\alpha}, Y = U_1 U_4, g_1^{\vec{\rho}}, T_{i,j} = g^{t_{i,j}}; \forall i \in [n], j \in [n_i])$.

Phase 1: $\mathcal{B}$ selects $\vec{y}_1, \vec{\sigma} \in \mathbb{Z}_N^{\omega}, t, y_2, y_3 \in \mathbb{Z}_N, y_{i,j} \in \mathbb{Z}_N$ at random, calculates and outputs the secret keys as follows. $S\bar{K}_S = (S, \vec{K}_1, K_2, K_3, K_{i,j}; \forall v_{i,j} \in S) = (S, g_1^{\vec{\sigma}} * g_3^{\vec{y}_1}, g_1^{\alpha + \langle \vec{\rho}, \vec{\sigma} \rangle} (U_1 g_3^{\hat{s}})^t g_2 g_3^{y_2}, g_1^t g_3^{y_3}, T_{i,j}^t g_3^{y_{i,j}}; \forall v_{i,j} \in S)$.

In addition, $\mathcal{A}$ can ask the oracles of leak and update.

Challenge: $\mathcal{B}$ generates the challenge ciphertexts as follows: $C_0 \xleftarrow{\$} \mathbb{G}_T, \vec{C}_1 = (g_1^s W_{24})^{\vec{\rho}} * g_4^{\vec{d}}, C_2 = (g_1^s W_{24}) \cdot g_4, \vec{C}_3 = \{T(\prod_{v_{i,j} \in B_k} T_{i,j})^{s_k} W_k\}_{k \in [\tilde{m}]}, \vec{C}_4 = (g_1^{s_k} V_k)_{k \in [\tilde{m}]}$.

Phase 2: It is similar with Phase 1. $\mathcal{B}$ can answer the queries of reveal and update with the restriction that the attributes sets of the adversary queries cannot meet the challenge access structure.

Guess: The adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins this game.

If $T \xleftarrow{\$} U_1^s D_{24}$, it is a semi-functional ciphertext, and $\mathcal{B}$ simulates the $Game_{final0}$. If $T \xleftarrow{\$} \mathbb{G}_{p_1 p_2 p_4}$, it is a random element, and $\mathcal{B}$ simulates the $Game_{final1}$. These two games are indistinguishable.

Theorem 1. *If the Assumptions 1, 2, 3 and 4 hold and for $l = (\omega - 1 - 2\tau)$ where τ is a positive constant, then the proposed scheme is anonymous and l leakage-resilient.*

Proof. Suppose the Assumption 1, 2, 3 and 4 hold. We can learn from the lemma 1 to lemma 5 that the adversary $\mathcal{A}$ can not distinguish between the $Game_{real}$ and $Game_{final1}$. So the value of b is hidden from the $\mathcal{A}$. The upper bound of the leakage between consecutive updates is l, so it is anonymous and l leakage-resilient. □

7 Anonymous Leakage-Resilient CP-ABE for Non-MAS

In this section, we give the construction of the anonymous leakage-resilient CP-ABE for non-monotone access structures. In this scheme, a non-monotone access structure is represented by the set of authorized sets in the non-monotone access structure.

Setup$((\lambda, V, l) \to (PK, MSK))$**:** Similar to Section 5, the setup algorithm sets the public keys as

$$PK = (N, g_1, g_4, g_1^{\vec{\rho}}, y, Y, T_{i,j}; \forall i \in [n], j \in [n_i]),$$

Table 2: Adversaries gain advantages over two consecutive games

Adjacent games	Adversary gain advantage differences	Related lemmas
$Game_{real}$ and $Game_0$	$\lvert Adv_{\mathcal{A}}^{Game_{real}} - Adv_{\mathcal{A}}^{Game_0}\rvert \leq \epsilon$	Lemma 1
$Game_{k'-1,2}$ and $Game_{k',1}$	$\lvert Adv_{\mathcal{A}}^{Game_{k'-1,2}} - Adv_{\mathcal{A}}^{Game_{k',1}}\rvert \leq \epsilon$	Lemma 2
$Game_{k',1}$ and $Game_{k',2}$	$\lvert Adv_{\mathcal{A}}^{Game_{k',1}} - Adv_{\mathcal{A}}^{Game_{k',2}}\rvert \leq \epsilon$	Lemma 3
$Game_{Q,2}$ and $Game_{final0}$	$\lvert Adv_{\mathcal{A}}^{Game_{Q,2}} - Adv_{\mathcal{A}}^{Game_{final0}}\rvert \leq \epsilon$	Lemma 4
$Game_{final0}$ and $Game_{final1}$	$\lvert Adv_{\mathcal{A}}^{Game_{final0}} - Adv_{\mathcal{A}}^{Game_{final1}}\rvert \leq \epsilon$	Lemma 5

where

$$y = e(g_1, g_1)^{\alpha}, Y = X_1X_4, T_{i,j} = g^{t_{i,j}}.$$

The master secret keys are

$$MSK = (X_1, g_3, \alpha).$$

Finally the algorithm publishs PK and keeps MSK.

KeyGen$((PK, MSK, S) \rightarrow SK_S)$**:** On input public keys PK, the master keys MSK, an attribute set $S = \{v_{1,x_1}, v_{2,x_2}, \cdots, v_{n',x'_n}\}$, where $n' \leq n, 1 \leq x_i \leq n_i$ for each $1 \leq i \leq n'$, this algorithm selects $t, y_2, y_3, y_4 \in \mathbb{Z}_N, \vec{y}_1, \vec{\sigma} \in \mathbb{Z}_N^{\omega}$, calculates and outputs the secret keys as follows.

$$SK_S = (S, \vec{K}_1, K_2, K_3, K_4),$$

in which

$$\vec{K}_1 = g_1^{\vec{\sigma}} * g_3^{\vec{y}_1}, \qquad K_3 = g_1^t g_3^{y_3},$$
$$K_2 = g_1^{\alpha + \langle \vec{\rho}, \vec{\sigma} \rangle} X_1^t g_3^{y_2}, \qquad K_4 = (\prod_{v_{i,j} \in S} T_{i,j})^t g_3^{y_4}.$$

UpdateUSK$((PK, S, SK_S) \rightarrow SK'_S)$**:** The update algorithm selects $\Delta t, \Delta y_2, \Delta y_3, \Delta y_4 \in \mathbb{Z}_N, \Delta\vec{y}_1, \Delta\vec{\sigma} \in \mathbb{Z}_N^{\omega}$ at random, and outputs a new key SK'_S:

$$SK'_S = (S, \vec{K}'_1, K'_2, K'_3, K'_4),$$

where

$$\vec{K}'_1 = \vec{K}_1 * g_1^{\Delta\vec{\sigma}} * g_3^{\Delta\vec{y}_1}, \quad K'_3 = K_3 g_1^{\Delta t} g_3^{\Delta y_3},$$
$$K'_2 = K_2 g_1^{\langle \vec{\rho}, \Delta\vec{\sigma} \rangle} X_1^{\Delta t} g_3^{\Delta y_2}, K'_4 = K_4 (\prod_{v_{i,j} \in S} T_{i,j})^{\Delta t} g_3^{\Delta y_4}.$$

Encrypt$((PK, M, \Gamma) \rightarrow CT_{\Gamma})$**:** Let Γ be a non-monotonic access structure where $\Gamma = \{B_1, B_2, ..., B_{\tilde{m}}\}$ and $B_k (k \in [\tilde{m}])$ is a set of attribute values and $\tilde{m}$ is the size of the non-monotone access structure Γ. This algorithm selects $s, s_1, s_2, ..., s_{\tilde{m}} \in \mathbb{Z}_N, \vec{d} \in \mathbb{Z}_N^{\omega}$, $W_k, V_k \in \mathbb{G}_{p_4} (k \in [\tilde{m}])$ at random and outputs the ciphertexts CT_{Γ} and the index $I_{B_k} \subset \{1, 2, ..., n\} (k \in [\tilde{m}])$ corresponding to attribute $B_k (k \in [\tilde{m}])$.

$$CT_{\Gamma} = (\{I_{B_k}\}_{k \in [\tilde{m}]}, C_0, \vec{C}_1, C_2, \vec{C}_3, \vec{C}_4),$$

where

$$C_0 = My^s, \qquad \vec{C}_1 = g_1^{s\vec{\rho}} * g_4^{\vec{d}},$$
$$C_2 = g_1^s g_4, \qquad \vec{C}_4 = (C_{4,k})_{k \in [\tilde{m}]} = (g_1^{s_k} V_k)_{k \in [\tilde{m}]},$$
$$\vec{C}_3 = \{C_{3,k}\}_{k \in [\tilde{m}]} = \{Y^s (\prod_{v_{i,j} \in B_k} T_{i,j})^{s_k} W_k\}_{k \in [\tilde{m}]}.$$

Decrypt$((PK, CT_{\Gamma}, SK_S) \rightarrow M)$**:** If the attribute set S satisfies the non-monotone access structure Γ, then $S \in \Gamma$, *i.e.*, $S = B_k (k \in [\tilde{m}])$ for $B_k \in \Gamma$. This algorithm calculates

$$M = \frac{C_0 \cdot \hat{e}_{\omega}(\vec{C}_1, \vec{K}_1) \hat{e}(C_{3,k}, K_3)}{\hat{e}(C_2, K_2) \hat{e}(C_{4,k}, K_4)}.$$

Theorem 2. *If Assumptions 1, 2, 3 and 4 hold, then the proposed scheme is anonymous and l leakage resilience.*

Proof. The security proof of anonymous leakage-resilient CP-ABE scheme for non-MAS can be derived from the proof of the scheme for MAS with minor modification. The minor modification is that in the simulation of key components for each $v_{i,j} \in S$ is just multiplied to get a single key component $K_4 = \prod_{v_{i,j} \in S} K_{i,j}$. □

8 Conclusion

In this paper, an anonymous leakage-resilient CP-ABE scheme for monotone access structures is proposed at first, in which the access structure is converted as minimal sets that can provide fast decryption. By using similar ideas, we present an anonymous leakage-resilient CP-ABE scheme with the constant size ciphertexts for non-monotone access structures. Both schemes are proven to be adaptively secure in the standard model under four static assumptions over composite order bilinear group and can tolerate continual leakage on the private keys when a update algorithm is implicitly employed to periodically update the private keys. However, our schemes cannot achieve the optimal leakage rate to ensure the efficiency, so designing an efficient ABE scheme with optimal leakage rate will be our future work.

Acknowledgments

This work was supported in part by the International S&T Cooperation Program of Shaanxi Province No. 2019KW-

056, the National Cryptography Development Fund under grant (MMJJ20180209).

References

[1] J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, and D. Wichs, "Public-key encryption in the bounded-retrieval model," *Lecture Notes in Computer Science*, vol. 2009, no. 5, pp. 113–134, 2010.

[2] J. Alwen, Y. Dodis, and D. Wichs, "Leakage-resilient public-key cryptography in the bounded-retrieval model," in *Advances in Cryptology*, pp. 36–54, 2009.

[3] N. Attrapadug, B. Libert, and E. D. Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Public Key Cryptography (PKC'11)*, pp. 90–108, 2011.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.

[5] Z. Cao, L. Liu, and Z. Guo, "Ruminations on attribute-based encryption," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 9–19, 2018.

[6] A. D. Caro, V. Iovino, and G. Persiano, "Fully secure anonymous hibe and secret-key anonymous ibe with short ciphertexts," in *International Conference on Pairing-Based Cryptography*, pp. 347–366, 2010.

[7] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *International Journal of Network Security*, vol. 16, no. 1, pp. 1–13, 2014.

[8] Y. Dodis, Y. T. Kalai, and S. Lovett, "On cryptography with auxiliary input," in *Proceedings of the 41st Annual ACM symposium on Theory of Computing*, pp. 621–630, 2009.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[10] A. Kapadia, P. Tsang, and S. W. Smith, "Attribute-based publishing with hidden credentials and hidden policies," *NDSS*, vol. 7, pp. 179–192, 2007.

[11] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Advances in Cryptology*, pp. 146–162, 2008.

[12] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *International Conference on Theory and Application of Cryptographic Techniques*, pp. 62–91, 2010.

[13] L. Liu, Z. Cao, and C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, pp. 29–35, 2018.

[14] S. Micali and L. Reyzin, "Physically observable cryptography," in *Theory of Cryptography*, pp. 278–296, 2004.

[15] T. Okamoto and K. Takashima, "Fully secure functional encyption with general relations from the decisional linear assumption," *Crypto*, vol. 6223, pp. 191–208, 2010.

[16] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access strtures," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*, pp. 195–203, 2007.

[17] T. Pandit and R. Barua, "Efficient fully secure attribute-based encryption schemes for general access structures," in *Provable Security*, pp. 193–214, 2012.

[18] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology*, pp. 457–473, 2005.

[19] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably seure realization," *Lecture Notes in Computer Science*, vol. 2008, pp. 321–334, 2008.

[20] B. Waters, "Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions," in *International Cryptology Conference on Advances in Cryptology*, pp. 619–636, 2009.

[21] Q. Yu and J. Li, "Continuous leakage resilient ciphertext-policy attribute-based encryption supporting attribute revocation," *Computer Engineering and Applications*, vol. 52, no. 20, pp. 29–38, 2016.

[22] T. H. Yuen, S. S. M. Chow, Y. Zhang, and S. M. Yiu, "Identity-based encryption resilient to continual auxiliary leakage," in *International Conference on Theory and Applications of Cryptographic Techniques*, pp. 117–134, 2012.

[23] L. Zhang, Y. Cui, and Y. Mu, "Improving privacy-preserving CP-ABE with hidden access policy," in *The 4th International Conference on Cloud Computing and Security*, pp. 596–605, 2018.

[24] L. Zhang, G. Hu, Y. Mu, and F. Rezaeibagha, "Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system," *IEEE Access*, no. 7, pp. 33202–33213, 2019.

[25] L. Zhang and Y. Shang, "Leakage-resilient attribute-based encryption with CCA2 security," *International Journal of Network Security*, vol. 22, no. 6, pp. 1–9, 2019.

[26] L. Zhang and H. Yin, "Recipient anonymous ciphertext-policy attribute-based broadcast encryption," *International Journal of Network Security*, vol. 20, no. 1, pp. 168–176, 2018.

[27] L. Zhang and J. Zhang, "Anonymous CP-ABE against side-channel attacks in cloud computing," *Journal of Information Science and Engineering*, vol. 33, no. 3, pp. 789–805, 2017.

[28] L. Zhang, J. Zhang, and Y. Hu, "Attribute-based encryption resilient to continual auxiliary leakage with

constant size ciphertexts," *Journal of China Universities of Posts and Telecommunications*, vol. 23, no. 3, pp. 18–28, 2016.

[29] L. Zhang, J. Zhang, and Y. Mu, "Novel leakage-resilient attribute-based encryption from hash proof system," *Computer Journal*, vol. 60, no. 4, pp. 541–554, 2017.

[30] M. Zhang, W. Shi, C. Wang, Z. Chen, and Y. Mu, "Leakage-resilient attribute-based encryption with fast decryption: Models, analysis and constructions," in *International Conference on Information Security Practice and Experience*, pp. 75–90, 2013.

Biography

Xiaoxu Gao is a master degree student in the school of mathematics and statistics, Xidian University. Her research interests focus on computer and network security.

Leyou Zhang is a professor in the school of mathematics and statistics at Xidian University, Xi'an China. He received his PhD from Xidian University in 2009. From Dec. 2013 to Dec. 2014, he is a research fellow in the school of computer science and software engineering at the University of Wollongong. His current research interests include network security, computer security, and cryptography.

A Distributed Density-based Outlier Detection Algorithm on Big Data

Lin Mei[1,2] and Fengli Zhang[1]
(Corresponding author: Fengli Zhang)
School of information and software engineering, University of Electronic Science and Technology of China[1]
Chengdu, China
School of Computer Science and Technology, Southwest Minzu University[2]
Chengdu, China
(Email: fzhang@uestc.edu.cn)
(Received Mar. 11, 2019; Revised and Accepted Nov. 6, 2019; First Online Jan. 29, 2020)

Abstract

As one of popular issues in data mining area, outlier detection aims to find the objects which show abnormal behaviors from original datasets' distribution, and it can be applied in various applications such as bank fraud, network intrusion detection, system health monitoring, medical care, public safety and security, and etc. Recently, the density-based outlier detection has been proposed which is the highly efficient and significant method for outlier detection processing. It adopts the relative density of an object to indicate the degree of an object is an outlier compared with its neighbors. Specifically, it aims at computing the Local Outlier Factor (LOF). In this paper, we propose a novel distributed density-based outlier detection method for large-scale data processing, namely IGBP. First, we split the data space into several grids and then allocates these grids into the data nodes with greedy algorithm in a distributed environment. Besides, we propose a distributed LOF computing method with KD-tree for detecting density-based outliers in parallel way. The validity of the proposed approaches is finally verified by experiments, Experimental results which demonstrate that our proposed method outperform the baselines.

Keywords: Density-based Outlier; Distributed Algorithm; Greedy Algorithm; Kd-Tree; LOF

1 Introduction

With the development of big data techniques in large scale data processing, outlier detection is one of importance but complex tasks in data mining area, it can be widely used in various applications such as bank fraud, network intrusion detection, system health monitoring, medical care and public security protection, and etc. Outlier detection can help us to discover valuable knowledge and abnormal patterns. Therefore, it has become one of hotspot directions in data mining, recently.

Hawkins who has addressed that an outlier is an observation that deviates obviously from other observations as to arouse suspicion that it was generated by a different mechanism [9]. In recent years, there are many studies about outlier detection. For example, distance-based outlier detection [13] and density-based outlier detection [5] are well representative works in traditional outlier detection methods. However, most of them only consider the centralized environments or single node processing. With the increasing scale of big data, the performance of these proposed methods cannot satisfy computing requirements of users. For instance, in the area of credit card fraud detection, we obtain the users' trade information as a dataset. If the credit card is stolen, its transaction pattern usually changes dramatically, especially that the locations of transactions and the purchased items are often unusual for the authentic card owner. Therefore, we define these abnormal transaction records as outliers. Then, the techniques of the outlier detection can help us to identify outliers to find out whether user accounts have been theft. And it can avoid the property damage. Besides, Outlier detection technology also can used to detect the cheating of game bots in Massively Multiplayer Online Role-playing Games [17].

Fortunately, there are some recent studies which attempt to utilize distributed computing environment to speed up the computation, and there are several related methods [10, 11, 14] for distributed outlier detection were proposed. For example, E. Lozano and E.Acufia [16] propose a master-slave architecture for distributed computing. More specifically, each slave node computes its neighborhood set and sends it to the master node. And the master node will collect all the partial neighborhood sets and compute LOFs of all the tuples. However, a large number of tuples in proposed method are aggregated to the master node, and then lots of calculations are needed

to conduct the result. Thus, the master node will be the bottleneck when the data scale is increasing. And a high-performance master node is necessary. Recently, Mei Bai et al. adopt a coordinator and a number of datanodes for outlier detection problem [3]. The coordinator is responsible for the overall scheduling. Each datanode stores several data subsets in grids, and calculates LOFs in the data subsets. The coordinator in this frame only takes charge of the scheduling and all the actual computations are allocated to the datanodes. Thus, the coordinator would not be a bottleneck if the data scale is large. In order to reduce the network overhead, their grid allocation algorithm allocates the adjacent grids to the same datanodes. However, their proposed algorithm only need a small quantity of network communications between pairs of datanotes and the average numbers of tuples in each grid are likely to be very different. Hence, to address these limitations, we focus on improve the computational complexity which is the greatest bottleneck of this issue.

In this paper, we aim to model density-based outlier detection in a distributed manner. The general idea is that we attempt to compare the density around an object with the density around its local neighbors. The basic assumption of density-based outlier detection method is that the density around a non-outlier object is similar to the density around its neighbors, but the density around an outlier object is significantly different from the density around its neighbors [8]. Through our sufficient analysis, we discover that workload and network communications in computing architecture will increase while the scale of data is increasing. But the increased speed of workload is higher than network communication. Besides, if the dimensionality of data increases, the performance will be much better. Therefore, we propose an improved algorithm in this paper which aims at detecting density-based outliers in distributed environments efficiently compared to [3]. Moreover, our experiments are implemented to demonstrate the effectiveness and high efficiency. We will show the detailed description of algorithm in Section 4.

The rest of this paper is organized as follows. In Section 2, we will overview the related works. Section 3 states the problem of density-based outlier detection in a distributed environment. Section 4 detailedly presents our improved algorithm. Section 5 gives the experimental results. In the end, we conclude this paper in Section 6.

2 Related Work

We first make briefly overview of outlier detection in Section 2.1. Then the previous methods of distributed outlier computing are described in Section 2.2.

2.1 Outlier Detection

Hawkins firstly give a definition about outliers in 1980 [9]. Afterwards, Beckman and Cook [4] present more improved definition and survey. Markou and Singh [18, 19] present a review about statistical approaches for outlier detection. Especially in [19], they present a review about neural network based approaches for outlier detection. Fujimaki *et al.* present a semi-supervised outlier detection method by using a set of labeled "normal" objects [7]. Dasgupta and Majumdar also propose a semi-supervised method [6] for outlier detection task. Subsequently, distance-based outliers was developed by Knorr and Ng [13]. And the index-based, nested loop–based and grid-based approaches are well explored to speed up distance-based outlier detection [12, 13].

Other proximity-based approach is the density-based outlier detection [5], In order to detect a tuple whether is an outlier or not, a local outlier factor (LOF) which defines in [5] represents the degree of this tuple to be an outlier is assigned to each tuple [3]. LOF is based on a concept of a local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density. By comparing the local density of a tuple to the local densities of its neighbors, one can identify the regions of similar density, and tuples that have a substantially lower density than their neighbors. These are considered to be outliers [8]. Besides, the HilOut algorithm was proposed by Angiulli and Pizzuti [2]. Aggarwal and Yu [1] develop the sparsity coefficient-based subspace outlier detection method. Kriegel *et al.* proposed angle-based outlier detection [15].

2.2 Outlier Detection in Distributed Environments

Lozano and Acufia propose a distributed algorithm to compute density-based outliers [16]. However, owing to all the tuples are transferred to the master node, the workload on the master node is quite heavy. Thus, this method is unable to achieve good performance when the data scale is huge.

Mei *et al.* adopt a coordinator and a number of datanodes [3]. And the coordinator is responsible for the overall scheduling. Each datanode stores several data subsets in grids, and calculates LOFs in the data subsets. After comparison, the coordinator in this frame is only in charge of the scheduling and all the actual computations are allocated to the datanodes. Hence, the coordinator would not be a bottleneck if the data scale is huge. In order to reduce the network overhead, their gird allocation algorithm allocates the adjacent grids to the same datanodes. However, in fact, their algorithm only need a small amount of network communications between pairs of datanotes. In addition, the time and space complexity of LOF are very high. In this paper, we present an improved algorithm based on theirs to efficiently detect density-based outliers in distributed environments. Moreover, our experiments are implemented to demonstrate the validity.

3 Preliminaries

3.1 Problem Formalism

we will show some definitions to better solve our problem and they will be applied in our proposed method.

Local density: It is estimated by the typical distance at which a tuple can be reached from its neighbors. And distance is usually adopted in LOF, which is an additional measure to produce more stable results within clusters.

Reachability distance: Let $d(A, B)$ be the distance between A and B, k-distance(A) be the distance of the object A to the k-th nearest neighbor. To simplify the description, we define the distance as Euclidean distance in the rest of this paper similar to [21]. Hence,

1) There are at least k tuples A' such that $d(A, A') \leq d(A, B)$.
2) There are at most $k-1$ tuples A'' such that $d(A, A'') < d(A, B)$.

Note that the set of the k nearest neighbors includes all tuples at this distance, which can in the case of a "tie" be more than k tuples. We denote the set of k nearest neighbors as $N_k(A) = A'|d(A, A') \leq d(A)$. This distance is used to define what is called reachability distance:

$$Rd_k(A, B) = \max\{k\text{-distance}(A), d(A, B)\}.$$

As shown in following Figure 1, it illustrates the original intention of reachability distance. Tuples B and D have the same reachability distance (k=3), while G is not a k nearest neighbor.

Thus, the reachability distance of an object A from B is the true distance of the two objects, but at least the k-distance(A). Objects that belong to the k nearest neighbors of A are considered to be equally distant. The reason for this distance is to get more stable results. Note that this is not a distance in the mathematical definition, since it is not symmetric.

The local reachability density of an object A is defined by

$$LRD(A) = \frac{1}{\frac{\sum_{B \in N_k(A)} Rd_k B,A}{|N_k(A)|}}$$

where the inverse of the average reachability distance of the object A from its neighbors. Note that it is not the average reachability of the neighbors from A (which by definition would be the k-distance(A)), but the distance at which A can be "reached" from its neighbors. With duplicate points, this value can become infinite.

The local reachability densities are then compared with those of the neighbors using

$$LOF_k(A) = \frac{\sum_{B \in N_k(A)} \frac{LRD(B)}{LRD(A)}}{|N_k(A)|}$$

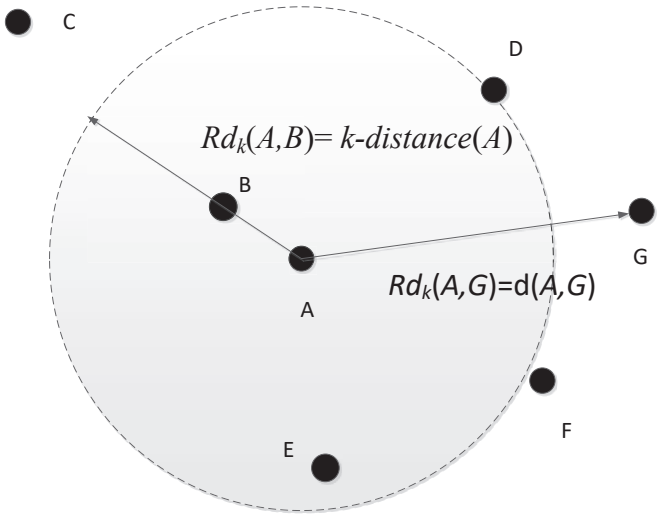


Figure 1: k-distance neighborhood and reachability distance when k=3

where the average local reachability density of the neighbors divided by the object's own local reachability density. A value of approximately a given threshold indicates that the object is comparable to its neighbors (and thus it is not an outlier). A value less than the threshold indicates a denser region (which would be an inlier), while values significantly larger than the threshold indicate outliers.

The traditional methods usually form all pair Euclidean distance matrix, and then run KNN query to proceed further. It is $\Theta(n^2)$ in terms of both space and time complexity. However, it can be improved with KD-tree [20].

3.2 Distributed Environment

As Figure 2 shows, we utilize a distributed framework that consists of a coordinator and a number of datanodes. The coordinator is for the overall scheduling similar to [3]. Each datanode is to store a portion of a complete data set. Most of previous algorithms utilize a master-slave architecture for outlier detection, while lots of computations are performed on the master node. Following [3], we also do that the coordinator in our frame only takes charge of the scheduling and all the actual computations are allocated to the datanodes. Besides, Density-based outlier detection is to compute the LOF of each tuple for a given integer k. First, we use improved GBP (IGBP) algorithm to split the data set into several subsets and assign them to the datanodes. After that, our algorithm work via two main steps. First, each datanode processes the local tuples. And LOFs of some tuples can be computed directly in the local nodes. Second, we will output LOFs of the rest of the tuples by a few necessary network communications compared with [3].

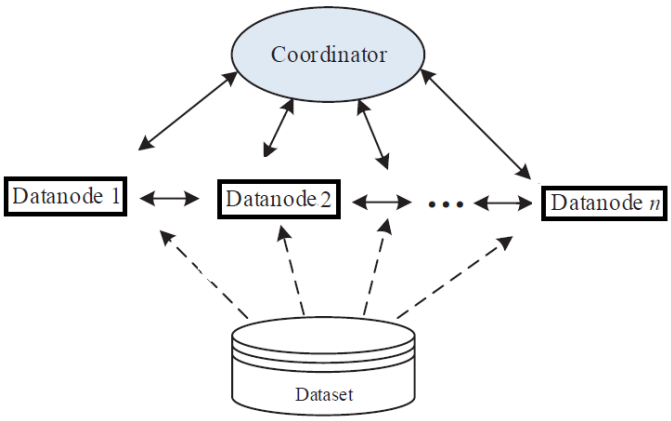


Figure 2: Computation frame

4 The Improved GBP Algorithm (IGBP)

4.1 Grid-based Partition

In our IGBP algorithm, we first attempt to split the whole d-dimensional space into several isometric grids. While grid-based partition method conducts limitations in the high-dimensional data, Hence, we cut each dimension into several equal segments (the number of segments is denoted by s). After that, the space is partitioned into s^d grids. Let $g_{x_1,x_2,\cdots,x_d}$ be the gird that is at the x_i-th position for dimension i. Next, we give the definition of adjacent grid:

$$N(g_{x_1,x_2,\cdots,x_d}) = \{g_{y_1,y_2,\cdots,y_d} | \max(1, x_i - 1) \leq y_i \leq \min(s, x_i + 1), g_{y_1,y_2,\cdots,y_d} \neq g_{x_1,x_2,\cdots,x_d}\}.$$

Next, we allocate these grids to the datanodes. In order to speed up the computations of density-based outliers, we propose an allocation method through considering the following two factors.

1) To obtain high parallelism, we set the number of tuples on each datanode almost the same (balance the workload).

2) According to Section 3.1, we compute the k-distance neighborhood for each tuple and reduce the network overhead if we allocate the adjacent grids to the same datanodes as possible. The details of the proposed method are shown in Algorithm 1.

Algorithm 1 Grid allocation

Input: The grid set G; The datanode set N;

Output: output: Allocation plan;

```
sort the grids in G according to the number of tuples in a grid in the descending order;
for each grid g in G do
    if there exist datanodes with no grid then
        n ← randomly choose a datanode with no grid;
    else
        Initialize a datanode set N';
        for each datanode n_i in N do
            if n_i has the least tuples then
                insert n_i into N';
            end if
            if there exists at least one datanode in N' with grids which belong to adjacent grids N_g then
                n ← choose the datanode in N' with the largest number of grids which belong to adjacent gridsN_g;
            else
                n←randomly choose a datanode in N';
            end if
        end for
    end if
    allocate g to n;
end for
```

4.2 Distributed LOF Computing

In above part, we have split the data set into several grids and allocated them to the corresponding datanodes. Next, we turn to compute the LOF for each tuple in parallel way. By analyzing the definitions in Section 3, it is clear that LRD is the premise of LOF. In order to calculate LRDs effectively, we have to compute the k-distances and k-distance neighborhoods for all the tuples first, which is the core part of this section.

However, in distributed environments, the situation is more complex. It's difficult to compute the actual k-distances of all the tuples. For example, in Figure 3, considering the tuples in grid g_1, the local k-distance neighborhood of tuple A is identical to its actual k-distance neighborhood. However, tuple B shows a complex situation. Its local k-distance neighborhood is C, D, E, which cannot be computed unless D, E is transmitted from g_2 to gird g_1.

For previous work in [3], they classify the tuples in a grid into 2 categories. A tuple whose neighborhoods can be computed in local grid is a grid-local tuple. Otherwise, it is a cross-grid tuple. After that, they proposed an algorithm to solve this problem. This part is not the most significant point in our paper. For distributed LOF computing, we will make a example to explain our proposed method which is different from previous work.

First, as shown in Figure 4, there is a set P with 120 tuples in 2-dimensional space, and the number of datanodes is 10. Thus, we set $s = 4$ and split the space into 16

Table 1: Improved algorithm process

Sequence number	Grid ID	Number of tuples	Allocated datanode	Average number of tuples
1	$g_{1,2}$	11	n_1	/
2	$g_{1,1}$	10	n_2	/
3	$g_{3,2}$	10	n_3	/
4	$g_{1,4}$	10	n_4	/
5	$g_{3,4}$	9	n_5	/
6	$g_{3,3}$	9	n_6	/
7	$g_{3,1}$	8	n_7	/
8	$g_{4,1}$	8	n_8	/
9	$g_{4,4}$	7	n_9	/
10	$g_{1,3}$	7	n_{10}	8.9
11	$g_{2,2}$	6	n_{10}	9.5
12	$g_{2,4}$	6	n_9	10.1
13	$g_{2,1}$	6	n_7	10.7
14	$g_{4,2}$	5	n_8	11.2
15	$g_{4,3}$	4	n_6	11.6
16	$g_{2,3}$	4	n_5	12

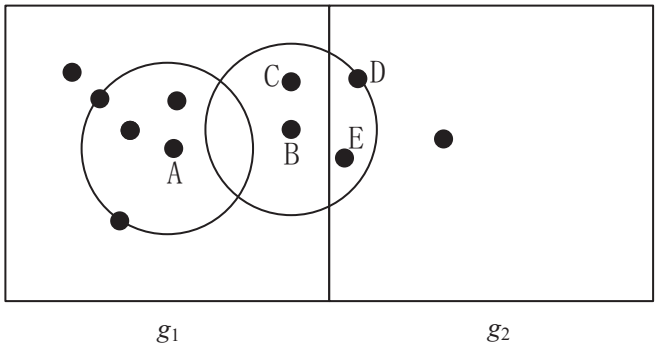


Figure 3: Example of DLC (k=3)

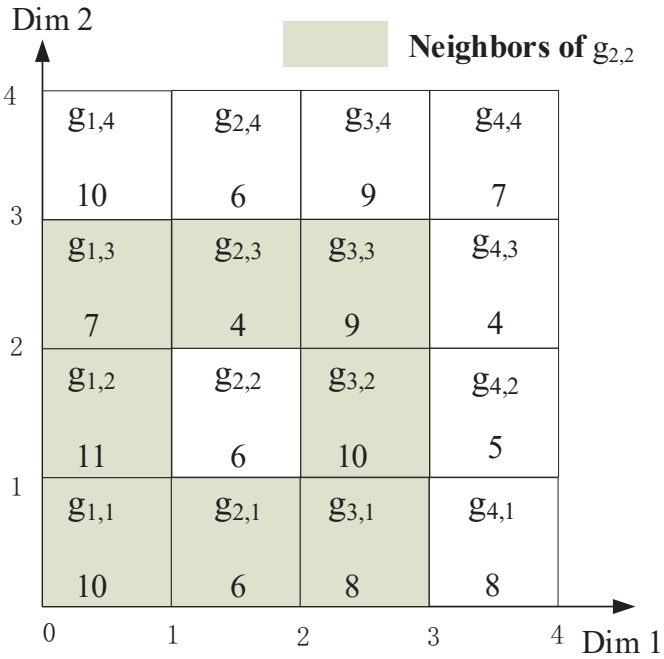


Figure 4: Example of grids

grids. The related number of tuples is shown at the bottom of each grid. According to the algorithm 1, we sort the grids by the number of tuples in a grid, and the result is shown in Table 1. Then, for each of the first 10 grids, we randomly allocate it to a datanode with no grid. After that, totally 89 tuples have been allocated, and the average number of tuples per datanode is 8.9. When allocating the 11th grid $g_{2,2}$, there are 4 datanodes whose numbers of tuples are not larger than 8.9, including n_7; n_8; n_9; n_{10}. We choose n_{10} because the number of tuples in n_{10} is smallest. Using the same method, we allocate all the grids to the corresponding datanodes.

After allocation, the number of tuples in data notes $\{n_1, n_2, \cdots, n_{10}\}$ is $\{11, 10, 10, 10, 13, 13, 14, 13, 13, 13\}$. We use σ (standard deviation) to indicate how spread out a data distribution is. A low standard deviation means that this algorithm balances the workload well. We use computational formula of standard deviation to obtain:

$$\sigma = \sqrt{\frac{11^2+10^2+10^2+10^2+13^2+13^2+14^2+13^2+13^2+13^2}{10}} \approx 1.483$$

Second, according to the previous algorithm in [3], the number of tuples in datanotes $\{n_1, n_2, \cdots, n_{10}\}$ is $\{11, 10, 10, 10, 15, 13, 14, 13, 11, 13\}$. We also obtain $\sigma \approx 1.732$.

5 Experimental Evaluation

We implement our proposed approaches using python programming language, and evaluate the performance in a cluster (with 4 data nodes and 1 coordinator) where each node (coordinator or datanode) has a Intel Core i5 @ 2.53 GHz CPU, 32G main memory. We first use a synthetic

Table 2: The influence of data scale

Method	100000 tuples		500000 tuples		1000000 tuples	
	k=10	k=20	k=10	k=20	k=10	k=20
In [16]	782ms	1243ms	5341ms	7988ms	13568	27755
In [3]	512ms	910ms	4122ms	6278	9742	22495
In IGBP	547ms	932ms	3907ms	5611	8472	16625

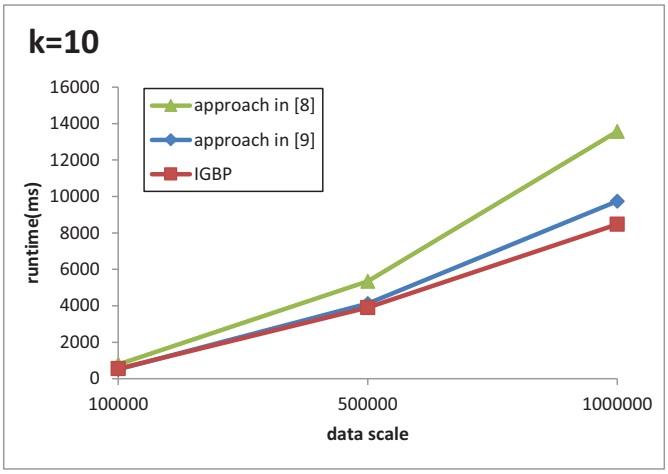


Figure 5: Runtime comparison

Table 3: Open datasets

	Number of Instances	Number of Attributes
Shuttle	58000	9
Census	48842	14
Drug	215063	6

Table 4: Results in three datasets

Methods	Runtime(ms)		
	Shuttle	Census	Drug
In [16]	12339	34987	47788
In [3]	3374	7120	9759
IGBP	**3115**	**5780**	**6880**

dataset to verify the efficiency of our proposed method IGBP. Then we choose three open datasets Shuttle, Census and Drug which have been widely used in outlier detection to further show the highly efficiency and effectiveness of IGBP.

5.1 IGBP with Synthetic Data

We generate various synthetic data sets to analyze the performance of our methods. Specifically, we compare our proposed method with [3, 16]. We generate several clustering center points. And the tuples in each cluster follow a Gaussian distribution. Finally, we add some generated noise into dataset, and the dimension of the data set is 3. The detailed parameter settings and the runtime are summarized in Table 2.

As shown in the table, with the increase of data scale, the runtime for our algorithm show better performance than previous work [3, 16]. Besides, experimental results demonstrate that workload balance is more significant than network overhead in this issue. Figure 5 illustrates that the impact of different data scale, with the increase of data scale, our proposed method outperform better baselines.

5.2 IGBP with Open Datasets

In this part, we use three popular real-world datasets to evaluate our proposed method. Shuttle, Census and Drug which have been widely used in outlier detection. The details has shown in following Table 3.

As Table 4 shows that our proposed method have low time consumption than [3, 16] based on three real-world datasets. More specifically, with the increase of data scale, we find that our proposed method have higher efficiency compared with [3, 16].

6 Conclusions

In this paper, we focus on the problem of density-based outlier detection in distributed environments for high-dimensional and large-scale data sets. We first introduce the basic concept of LOF. We summarized the approach in [3] to solve the problem of distributed LOF computing in detail. The advantages and disadvantages of the approach are discussed. Then, we propose an improved algorithm based on greedy algorithm, namely IGBP. With the experimental results, we show the efficiency and effectiveness of the proposed approaches compared with previous work. The results demonstrate that our algorithm outperform baseline. In future, we will considerate more efficient policies to balance the time consumption and space consumption.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61472064,61602096) and the Sichuan Provincial Science and Technology Department Project (2018GZ0087,2016FZ0002).

References

[1] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in *ACM Sigmod Record*, vol. 30, pp. 37–46, 2001.

[2] F. Angiulli and C. Pizzuti, "Outlier mining in large high-dimensional data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 203–215, 2005.

[3] M. Bai, X. Wang, J. Xin, and G. Wang, "An efficient algorithm for distributed density-based outlier detection on big data," *Neurocomputing*, vol. 181, pp. 19–28, 2016.

[4] R. J. Beckman and R. D. Cook, "Outlier ... ...s," *Technometrics*, vol. 25, no. 2, pp. 119–149, 1983.

[5] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *ACM Sigmod Record*, vol 29, pp. 93–104, 2000.

[6] D. Dasgupta and N. S. Majumdar, "Anomaly detection in multidimensional data using negative selection algorithm," in *Proceedings of the Congress on Evolutionary Computation (CEC'02)*, vol. 2, pp. 1039–1044, 2002.

[7] R. Fujimaki, T. Yairi, and K. Machida, "An approach to spacecraft anomaly detection problem using kernel feature space," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 401–410, 2005.

[8] J. Han, J. Pei, and M. Kamber, "Data mining: Concepts and techniques," *A volume in the Morgan Kaufmann Series in Data Management Systems*, 2012. (`https://www.sciencedirect.com/book/9780123814791/data-mining-concepts-and-techniques`)

[9] D. Hawkins, "Identification of outliers," *Monographs on Statistics and Applied Probability*, vol. 11, 1980.

[10] Q. He, Y. Ma, Q. Wang, F. Zhuang, and Z. Shi, "Parallel outlier detection using kd-tree based on mapreduce," in *IEEE Third International Conference on Cloud Computing Technology and Science*, pp. 75–80, 2011.

[11] E. Hung and D. W. Cheung, "Parallel mining of outliers in large database," *Distributed and Parallel Databases*, vol. 12, no. 1, pp. 5–26, 2002.

[12] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *The VLDB Journal—The International Journal on Very Large Data Bases*, vol. 8, no. 3-4, pp. 237–253, 2000.

[13] E. M. Knox and R. T. Ng, "Algorithms for mining distancebased outliers in large datasets," in *Proceedings of the International Conference on Very Large Data Bases*, pp. 392–403, 1998.

[14] A. Koufakou, J. Secretan, J. Reeder, K. Cardona, and M. Georgiopoulos. "Fast parallel outlier detection for categorical datasets using mapreduce," in *IEEE International Joint Conference on Neural Networks*, pp. 3298–3304, 2008.

[15] H. P. Kriegel, A. Zimek, *et al.*, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 444–452, 2008.

[16] E. Lozano and E. Acufia, "Parallel algorithms for distance-based and density-based outliers," in *Fifth IEEE International Conference on Data Mining*, pp. 4, 2005.

[17] Y. Lu, Y. Zhu, M. Itomlenskis, S. Vyaghri, and H. Fu, "Mmoprg bot detection based on traffic analysis," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 18–26, 2015.

[18] M. Markou and S. Singh, "Novelty detection: A review—part 1: statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[19] M. Markou and S. Singh, "Novelty detection: A review—part 2:: Neural network based approaches," *Signal Processing*, vol. 83, no. 12, pp. 2499–2521, 2003.

[20] F. P. Miller, A. F. Vandome, and J. McBrewster, *Kd-Tree*, 2009. (`https://www.amazon.fr/kd-tree-Frederic-P-Miller/dp/6130094590`)

[21] E. Schubert, A. Zimek, and H. P. Kriegel, "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 190–237, 2014.

Biography

LinMei received B.Sc and M.Sc in computer science and technology from University of Electronic Science and Technology of China in 2003 and 2006, He is currently a lecturer in Southwest Minzu University. And pursing Ph.D degree in University of Electronic Science and Technology of China. His research interests include network security and cloud computing.

Fengli Zhang received B.Sc, M.Sc and Ph.D in computer science and technology from University of Electronic Science and Technology of China in 1983, 1986 and 2007. She is currently a professor in University of Electronic Science and Technology of China, her research interest include database management, network security and big data processing.

Binary Executable Files Homology Detection with Genetic Algorithm

Jinyue Bian[1] and Quan Qian[1,2]
(Corresponding author: Quan Qian)
School of Computer Engineering and Science, Shanghai University[1]
Shanghai 200444, China
Materials Genome Institute, Shanghai University, Shanghai, China[2]
(Email: qqian@shu.edu.cn)
(Received Mar. 16, 2019; Revised and Accepted Dec. 29, 2019; First Online May 9, 2020)

Abstract

Software homology detection is very meaningful for software copyright protection and malicious code variants detection. In this paper, we propose a genetic algorithm to justify the binary code similarity. First of all, the binary executable files are converted into control flow graph , and then use genetic algorithm to compute the similarity among control flow graphs, which is regarded as the evaluation metric for software homology detection. The experimental results show that the method is not only effective, but also the average time efficiency is 0.3 times that of the classical algorithm of graph edit distance.

Keywords: Binary Executable Files; Control Flow Graph; Genetic Algorithm; Homology Detection

1 Introduction

With the rapid development of software, code plagiarism is emerging endlessly, which seriously threatens the software intellectual property rights. In addition, in March 2018, a total of 7,235,983 viruses were found in the National Computer Virus Emergency Center,36,931 new viruses were added, and 76,606,087 computers were infected [13]. Viruses or malware's escape from detection through code variants, but the core code does not change much. Therefore, detection of code similarity is very necessary. And the existing methods can be divided into two categories, source code based or binary code based. Considering sometimes we can not get the source code. Therefore, binary code based homology detection is more promising. That is, the similarity analysis based on binary code is very important.

For binary code similarity detection, graph based method plays an important role in which Control Flow Graph (CFG) is one of the most commonly used method. Therefore, for binary code homology detection, it can be transferred to graph similarity matching problems. That is, given two graphs, graph matching involves establishing the corresponding relations between their vertexes and considering the consistency of edge sets at the same time. CFG similarity comparison methods include graph edit distance (GED), string matching, execution sequence comparison, matching program basic blocks, and so on. However, in general, the computation complexity of these methods are quite large. Therefore, in this paper, we propose a new graph matching algorithm for binary code similarity analysis. First, the binary file will be converted into CFG with relatively complete control flow information using the dynamic and static combination technique. And then use genetic algorithm (GA) to calculate the similarity between CFGs. The algorithm can be used to accurately identify the isomorphism subgraph relationship and the exact identical CFGs, which can shorten the running time greatly, so as judging the software homology effectively.

The organization of the paper is as follows: Section II describes some related work. Section III presents the framework of the proposed method. The experimental results and detailed analysis are discussed in Section IV. Section V concludes the whole paper and outlines some directions of the future work.

2 Related Work

So far, there are certain amounts of research on homology identification related area. Here, we will give some background work in two aspects, including sequenced based analysis method and graph based method that closely related to this paper.

2.1 Sequence-based Analysis Method

Aiming at the problem of source code plagiarism, Guo proposed an improved code plagiarism detection algorithm based on abstract syntax tree, which can detect plagiarism effectively [22]. Koschke demonstrated how suffix trees can be used to obtain a scalable comparison,

and presented a method to improve the accuracy through user feedback and automatic data mining [10]. Liu presented an improved abstract syntax tree, which can effectively detect code plagiarism by modifying the variable type and adding meaningless variables [14].

However, the above mentioned methods are all sequence-based ones, that can be bypassed by interference techniques, such as instruction rearrangement, equivalent instruction sequence replacement, branch inversion, *etc.* And the essence of interference is that malicious code can produce homogeneous code with different syntax but same semantics. Therefore, the other direction is dynamic based analysis, which generally relies on dynamic execution log to analysis the program behaviour. For instance, through analyzing the anomaly and similarity of process access behavior in data flow dependent networks, Mao *et al.* introduced an active learning method by minimizing risk estimation, which can improve the detection effect of malicious code apparently [23]. Although dynamic based method can extract the code running features, it relies on virtual running environment and also can be challenged by anti-virtual machine attacks [21].Yang *et al.* introduces a method of defect detection based on homology detection technology for open source software [28].

So, we can use more information, for instance, the function call diagram, to further detect malicious code. In this aspect, Chae proposed a software plagiarism detection system using an API labeled CFG (A-CFG) that abstracts the functionalities of a program [2]. Lim presented a method to compare CFGs by matching the basic block of a binary program, which can effectively identify the similar CFGs [12]. Wu gave a parallel method to extract the function call graph from the source code, and introduced a new software structure information comparison algorithm to effectively check the homology of the software [24].

2.2 Graph Matching Based Method

Graph matching is a classical problem in computer science. At the same time, GA also has some applications in graph matching, code similarity detection and other security areas [20]. Moon proposed a malware detection system using a hybrid GA, in which a malware is represented as a directed dependency graph and transforms the malware detection problem to the subgraph isomorphism problem [8]. Jaeun gave a multi-objective GA with a local search heuristic, comparing the degrees of each vertex of two graphs that are mapped, and counted the number of mismatched vertices [5]. Kim proposed a new cost function based on the program dependency graph using the GA to measure the similarity of the program, the method was proved to be feasible [7]. Xiang presented an improved GA, which mainly studied the isomorphism of subgraphs, and designed a special crossover function and a new fitness function to measure the evolution process [25].

In this paper, we combine static and dynamic method to generate CFG, and then use graph matching to evaluate the binary software homology. The main contributions of the paper are as follows:

- Design a special GA to evaluate the similarity of binary files, which can be used to accurately identify the isomorphism subgraph and the exact identical CFGs, which is normally regarded as a NP-complete problem. And the experimental result shows when the number of CFG nodes is large, our method can still get the CFG similarity and comparing with other classical methods, the time overhead of our algorithm is obviously reduced.
- For GA based CFG matching, we give a complete framework including CFG mapping to chromosome, operation design for the crossover, mutation, selection and fitness evolution.

3 The Proposed Method

The method proposed can be divided into two steps. The first step is CFG extraction from the binary executable files combining static and dynamic recovery method. And the second step is CFG encoded and read into GA to compute the similarities among different graphs. The flowchart of the method is shown in Figure 1, where the implementation of each step is described in detail below.

3.1 Binary Files' CFG Extraction

Although static method has high code coverage, it cannot obtain complete control flow. Although dynamic method can obtain the exact program execution information, the code coverage is low. So, the hybrid recovery method is the combination of dynamic and static method, which ensures not only the high code coverage but also can solve the indirect jump issues. And then use symbolic execution and reverse slicing techniques to further traverse the binary file to obtain complete control flow information.

3.1.1 Symbolic Execution

Symbolic execution is the symbol substitution for real values when running a program. The advantage is that we can traverse the paths of a program as much as possible by using the variable symbols. KLEE [9] and S2E [3] are two typical symbolic execution tools based on source code and binary code respectively. In symbolic execution, we can convert the program operation process into a mathematical expression. Whenever a judgment and a jump statement are encountered, symbolic execution gathers the path constraints of the current execution path into the constraint set of that path. Through constraint solver, the path reachability can be obtained by solving the constraint set. Combining the actual execution with the symbol execution, that first emerged in 2005, in DART [18] and CUTE [19], many problems encountered in traditional static symbol execution have been solved.

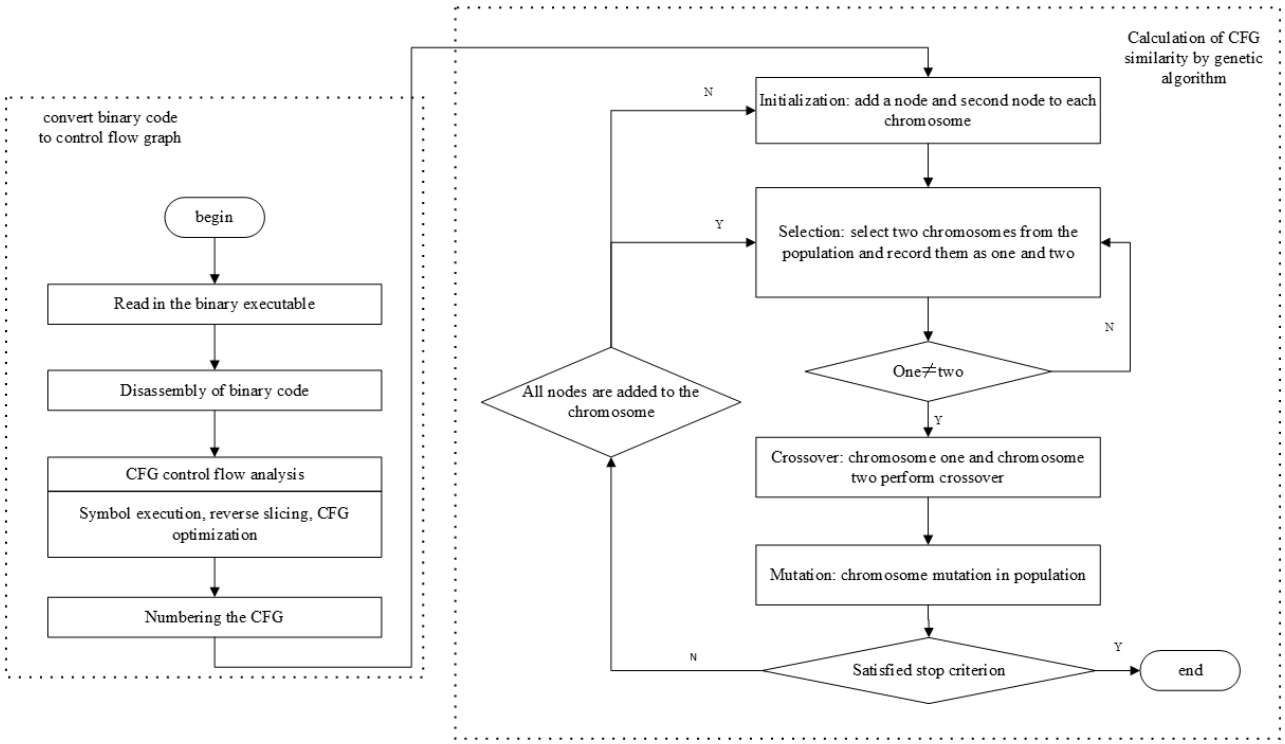


Figure 1: The flowchart of binary code homology detection

However, there are some challenges to symbolic execution, for instance, the path explosion, solver unable to solve, *etc.* For reducing the path explosion, heuristic functions, reliable program analysis and software verification techniques are some common strategies for reducing the complexity of path exploration. For constraint solving problems, there are two kinds of optimization methods. One is the elimination of uncorrelated constraints and the other is incremental solution.

3.1.2 Reverse Slicing

Program slicing is an important technique for program analysis, Negi *et al.* highlights the different test cases and comparative analysis of program slicing methods which corresponds to the applications which are usually utilized in software modification activities [15]. Given the slicing standard $< p, V >$, the forward slicing of program p contains all statements and control conditions affected by variables in V, while the backward slice of program p contains all statements and control conditions that have direct or indirect effects on variables in V.

3.1.3 CFG Generation and Optimization

Currently, the common tools for generating CFG are FXE (forced execution engine) [26], IDA Pro [17], *et al.* FXE can solve the problem of indirect branch jump by dynamically executing code, it is only suitable for the binary executable of Windows PE format under x86 architecture. IDA Pro can generate CFG and function call graph, but the repeated nodes in the graph are not merged or deleted, also for binary codes, it lacks structural information, so IDA Pro's disassembly results are far from ideal [29].

In this paper, we use a new CFG tool Angr [27], which is a binary analysis platform based on Python. It implements different symbolic execution strategies, such as veritesting [1]. The Angr system is originally designed for the DARPA Challenge. It can load different binary format files and their dependent libraries, and integrate many advanced binary analysis techniques to generate CFGs by combining dynamic and static methods. Moreover, Angr can optimize the basic block overlap in the CFG, and the overlapping parts are merged and subdivided to obtain more accurate control flow information. Meanwhile, since Angr analyses and removes the edges generated by loop structure and pseudo-return, it simplifies the CFG and alleviates the explosion problem of program state space, which improves the reconstructing ability of CFG.

3.2 GA for CFGs Similarity Calculation

Given two directed graphs, and then number the nodes in the graph [4], for example, as shown in Figure 2. We use GA to select the best individual after several generations of evolution, such as crossover and mutation, until the evolution stop criteria is satisfied.

3.2.1 Chromosome Initialization

The GA adds a node of the matching graph to each chromosome once each iteration, a corresponding node of the matched graph is generated by random function until all

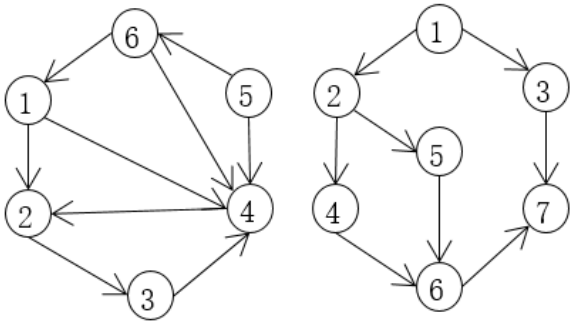


Figure 2: Two examples of numbering nodes for directed graphs

the nodes of the matching graph are added to the chromosome. According to Figure 2, the initialization process is shown in Table 1, all the left nodes in the chromosome comes from a relatively small matching graph (the left one of Figure 2) and the right nodes come from the matched graph (the right one of Figure 2). About the matching graph, adding some rules for initialization:

- When the in-degree of the left node is zero, and the out-degree of the right node is zero, they do not match. And if the out-degree of the left node is zero, and the in-degree of the right node is zero, it will not match. For example, in Figure 2, node 5 of the left graph doesn't match node 7 of the right graph, because the in-degree of node 5 is zero and the out-degree of node 7 is zero.

- The current right node number is different from the previous right node number in the chromosome, and the number does not exceed the total number of nodes in the matched graph. For example, in the first chromosome of Table 1, the node corresponding to the left node 2 must not be node 5 and the number doesn't exceed 7.

3.2.2 Chromosomes Selection

Both crossover and mutation operations in GA rely on selection function. The selection operation in this paper is similar to roulette-wheel selection. First of all, a random function produces a number between 0 and 1 denoted as a. Sum the fitness value of all chromosomes in the population as S; Next, calculate the cumulative fitness value for each chromosome, that is the sum of the fitness from the first chromosome to the current one k, marked as S_k. Set S_k as the divisor and S the dividend, the resulting quotient is recorded as b_k and compared with a, as shown in Equation (1). If $a \leqslant b_k$, returns the current chromosome number k.

For example, in Table 1, according to Equation (1), the cumulative fitness value of the first chromosome is $b_1 = \frac{f1}{f1+f2+f3+f4}$, and the cumulative fitness value of the second chromosome is $b_2 = \frac{f1+f2}{f1+f2+f3+f4}$. If $a \leqslant b_2$, returns the current chromosome number 2, and so on.

$$b_k = \frac{S_k}{S} = \frac{\sum_{i=1}^{k} f_i}{\sum_{i=1}^{L} f_i}. \tag{1}$$

Where f denotes the fitness value and $L = sizepop$, $k \leqslant sizepop$.

3.2.3 Chromosomes Crossover

Before a crossover, we generate a number between 0 and 1 through a random function, denoted as $rand$. If $rand \leqslant cross$, then do crossover operation, otherwise, keep silent.

Two chromosomes are extracted from the population by the selection function, the right nodes of the two chromosomes (noted as $Parent\_one$ and $Parent\_two$) are crossed, and the beginning position of the middle part is marked as "$begin$", the ending position of the middle part is marked as "end". $begin$ and end generated by random functions must satisfy $begin < end$, and the difference is not equal to the length of the whole chromosome. Let the middle part of the $Parent\_one$ as the middle part of the $child\_one$ after crossing, then let the nodes of the $Parent\_two$ in turn to fill in the $child\_one$, and after that do conflict detection. If there is a conflict, traversing backward in turn. After the traversal, if the nodes of the $child\_one$ chromosome is not filled up, then use the random function to generate the missing node and then perform conflict detection. According to the above crossover rules, can obtain two child chromosomes after the crossover.

For example, take the right node of the first and second chromosome in Table 1 to do a crossover and assume that $begin = 3$, $end = 5$, the specific crossover steps with the $Parent\_one$ in mind are shown in Figure 3. Thus, the chromosome produced by the crossover are $1 \to 6, 2 \to 5, 3 \to 4, 4 \to 2, 5 \to 1, 6 \to 3$.

Combine two chromosomal populations before and after the crossover and then sorted according to the fitness value in ascending order. If the population size is set to $sizepop$, then select the top $sizepop$ chromosome to form the population after crossover.

Table 1: An example of chromosome initialization

iterations #	First chromosome	Second chromosome	Third chromosome	Fourth chromosome
1	$1 \to 5$	$1 \to 6$	$1 \to 1$	$1 \to 2$
2	$1 \to 5, 2 \to 6$	$1 \to 6, 2 \to 1$	$1 \to 1, 2 \to 4$	$1 \to 2, 2 \to 3$
3	$1 \to 5, 2 \to 6$ $3 \to 4$	$1 \to 6, 2 \to 1$ $3 \to 4$	$1 \to 1, 2 \to 4$ $3 \to 2$	$1 \to 2, 2 \to 3$ $3 \to 6$
4	$1 \to 5, 2 \to 6$ $3 \to 4, 4 \to 2$	$1 \to 6, 2 \to 1$ $3 \to 4, 4 \to 2$	$1 \to 1, 2 \to 4$ $3 \to 2, 4 \to 3$	$1 \to 2, 2 \to 3$ $3 \to 6, 4 \to 4$
5	$1 \to 5, 2 \to 6$ $3 \to 4, 4 \to 2$ $5 \to 1$	$1 \to 6, 2 \to 1$ $3 \to 4, 4 \to 2$ $5 \to 5$	$1 \to 1, 2 \to 4$ $3 \to 2, 4 \to 3$ $5 \to 6$	$1 \to 2, 2 \to 3$ $3 \to 6, 4 \to 4$ $5 \to 1$
6	$1 \to 5, 2 \to 6$ $3 \to 4, 4 \to 2$ $5 \to 1, 6 \to 3$	$1 \to 6, 2 \to 1$ $3 \to 4, 4 \to 2$ $5 \to 5, 6 \to 3$	$1 \to 1, 2 \to 4$ $3 \to 2, 4 \to 3$ $5 \to 6, 6 \to 5$	$1 \to 2, 2 \to 3$ $3 \to 6, 4 \to 4$ $5 \to 1, 6 \to 5$
fitness function value	f1	f2	f3	f4

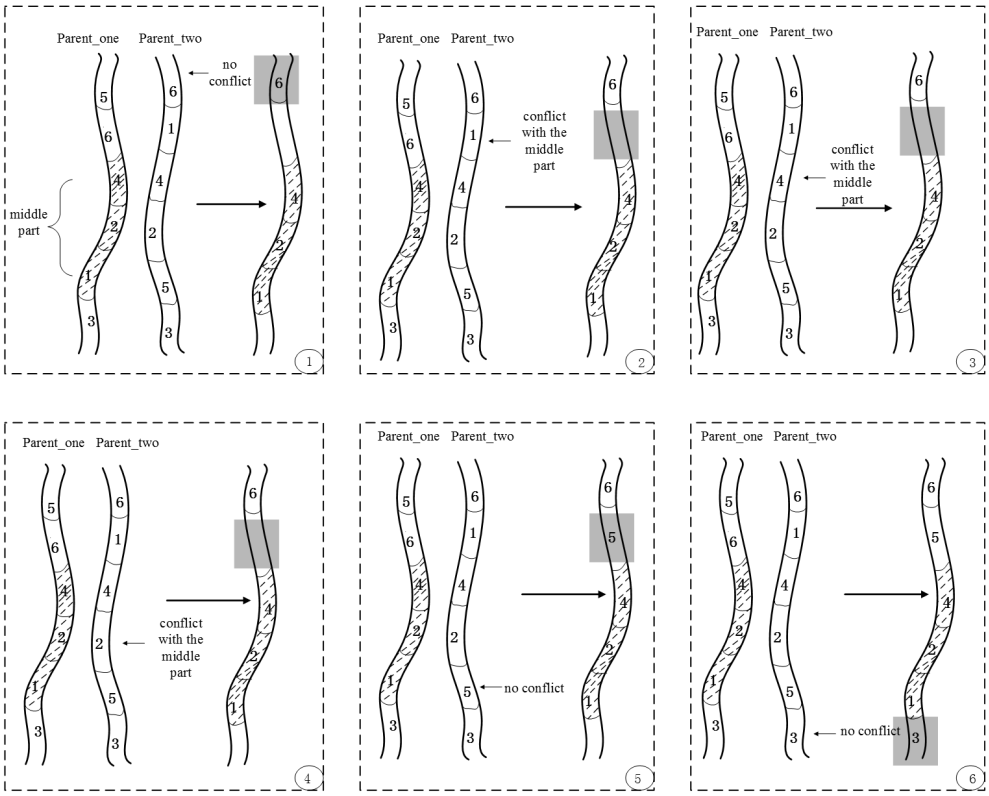


Figure 3: An example of crossover step

3.2.4 Chromosome Mutation

Each chromosome in population is selected in turn on the basis of the crossover operation. We generate a number between 0 and 1 through a random function, which is denoted as *rand*. If $rand \leqslant mutation$, the mutation operation is performed, otherwise, no mutation. When the nodes of the second graph are not all added to the chromosome, we use different random functions to generate a mutation position (*position*) and a mutation value (*num*) in the chromosome, and then do conflict detection. If there is no conflict, replace them. Otherwise, a new mutation is generated. If all the nodes of the second graph have been added to the chromosomes, use random functions to generate two different positions in the chromosome (i and j), and exchange the right nodes of the two positions as the mutated chromosomes.

For example, the third chromosome in Table 1, not all nodes of the second graph are added to the chromosome at the sixth iteration, and suppose $position = 3$, $num = 4$, it is obvious that the mutation value generated at this point conflicts with the subsequent right node, so suppose we regenerate a mutation value $num = 7$, and there is no conflict, then the chromosome mutation is shown as the left figure in Figure 4. Therefore, after the third chromosome mutation, it's $1 \to 1, 2 \to 4, 3 \to 7, 4 \to 3, 5 \to 6, 6 \to 5$. If there are 6 nodes in the second graph and all of them have been added to the chromosome, assuming $i = 2$, $j = 5$, the chromosome mutation is shown as the right figure in Figure 4. Thus, after the 3rd chromosome mutation, it's $1 \to 1, 2 \to 6, 3 \to 2, 4 \to 3, 5 \to 4, 6 \to 5$.

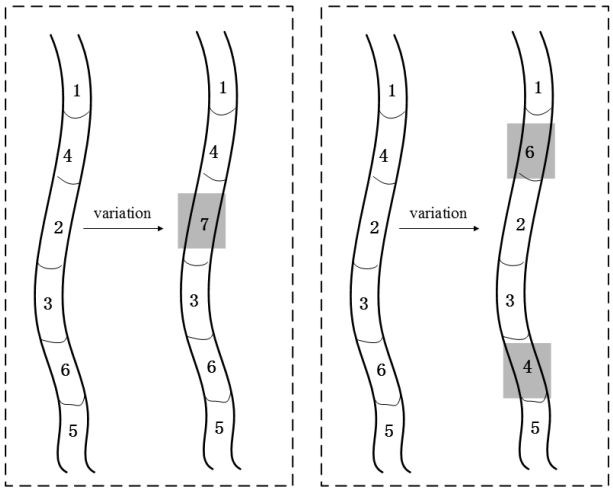


Figure 4: An example of mutation process

According to the rule of mutation, we perform the operation of chromosome variation, and we compare the fitness value of chromosomes before and after mutation. If the fitness value of the chromosome is not reduced after mutation, the mutation operation is canceled.

3.2.5 Fitness Calculation for Each Chromosome

In this paper, the fitness function F is divided into two parts $F1$, $F2$, and define $F = F1 + F2$. $F1$ is composed of the mismatched nodes number $f1$ and the mismatched edges number $f2$, that is, $F1 = f1 + f2$, where $f1$ and $f2$ are both for smaller graphs. For example, in Table 1, for the 4th chromosome, after 3 times of iteration, the results are: $1 \to 2, 2 \to 3, 3 \to 6$, at this time $f1 = 3, f2 = 2$.

However, only according to the results obtained by $F1$ is not scientific, because when two graphs are exactly the same and two graphs are isomorphic subgraphs, the results of $F1$ are all zero. Therefore, we add the $F2$ part, as shown in Equation (2), if the number of nodes in the two CFGs is equal, then $F2$ is zero, otherwise it is not zero.

$$F2 = 1 - \frac{node\_one}{node\_two}. \tag{2}$$

Where *node_one* represents the number of nodes of the smaller graph, and *node_two* the larger graph.

Therefore, the fitness function of the proposed method is shown as Equation (3).

$$F = F1 + F2 = f1 + f2 + 1 - \frac{node\_one}{node\_two}. \tag{3}$$

With this fitness function, if $F1 = 0$, $F2 = 0$, the nodes and edges of two CFGs are exactly the same. And if $F1 = 0$, $F2 \neq 0$, the two CFGs are isomorphic subgraphs.

3.2.6 Stop Criteria of Population Evolution

When any one of the followings three rules occurs, the GA will stop iteration and return the chromosome with the smallest fitness value.

- The evolution times reach the predefined number;
- During evolution, the F1 value of a chromosome equals 0;
- After several iterations, comparing with the population changes before and after each evolution, there are almost no changes in the fitness value.

4 Experiments

4.1 Experimental Data and Environment

Kernel32.dll is a very important dynamic link library file in Windows, which belongs to kernel level file. It controls the system memory management, the data input and output operation and interrupt processing. When the Windows start, the kernel32.dll resides in a specific write-protection area in memory, preventing other programs from occupying the memory area [16]. We select *loadimagefile*, *loadappinitdlls* and *basepappinitdlls* in kernel32.dll and *write.exe* as experimental data. In detail, we extract the experimental data files from Windows 7 and Windows 10, altogether 8 files. *write.exe*,

loadimagefile, *loadappinitdlls*, *basepappinitdlls* of Windows 10 system. For Windows 7 system, named them as *write_7.exe*, *loadimagefile_7*, *loadappinitdlls_7* and *basepappinitdlls_7*, and numbered them as $1, \ldots 8$, for subsequent description. All the experiments is under the Windows 10 system, and the related software include the IDA, Angr, and VS2017. In addition, for GA parameters, we set the number of population (*sizepop*) 20, the number of evolution (*maxgen*) 2000, the crossover rate (*cross*) 0.3, and the mutation rate (*mutation*) 0.8.

4.2 Comparative Experimental Methods

Among all graph matching algorithms, GED is the classical one with good fault tolerance and be suitable for various types of graphs. Although there are other graph similarity computation methods, especially the lately hot graph neural network (GNN), it needs not only massive data to train the network but also rich computational resources. Therefore, we select a series of GED algorithms as comparative ones, which can be more targeted when compared with the experimental results of the proposed method.

4.2.1 GED

The GA proposed in this paper is mainly used to calculate the similarity between two directed graphs, which is consistent with the classical GED method. Therefore, our comparative experiment selects the edit distance method and its two variant methods. GED is the sum of the minimum editing operation costs required to edit a source graph into a target graph. The cost of each step is obtained by defining the corresponding cost function [30]. In this paper, the cost of replacing, deleting and inserting the nodes and edges of the directed graph in the edit-distance method is all set to 1.

4.2.2 Edit Distance Based on Binary Linear Programming Formula

In 2015, Julien Lerouge proposed a new binary linear programming formulations for computing the exact GED between two graphs (GED_linear) [11], the advantage of which is the universality of the formula, the similarity between digraphs and undirected graphs can be calculated. But when the number of nodes is too large, the method does not work, and the time cost of this method is larger than that of other methods.

4.2.3 Edit Distance Based on Hausdorff Matching

In 2015, Andreas Fischer proposed a quadratic time approximation of GED based on Hausdorff matching (GED_distance) [6]. The advantage of the method is that the time performance is greatly improved compared with other methods. However, the disadvantage is that some accuracy is lost, that is, a small loss of precision.

4.3 Experimental Results and Analysis

4.3.1 Comparison of CFG Generated by Angr and IDA

In order to verify the integrity and simplification of the CFG generated by Angr, we compare and analyze the CFG generated by Angr and IDA. For example, the CFG of the CADET_0001 file given by the DARPA Challenge, Figure 5 is a CFG generated using Angr, and Figure 6 by IDA. The number of basic blocks in the CFG generated by Angr is more than that of IDA, and the basic block of CFG generated by Angr also contains the basic block of CFG generated by IDA. After merging overlapped nodes and eliminating cyclic and unreachable edges and nodes, Angr can obtain more accurate CFG, which can provide a more concise CFG for subsequent analysis.

Functions

Name	Address	Binary	Size	Blocks
sub_8048080	8048080	CADET_00001	273	14
sub_80481a0	80481a0	CADET_00001	435	23
sub_8048360	8048360	CADET_00001	204	12
sub_804842c	804842c	CADET_00001	11	2
sub_8048437	8048437	CADET_00001	12	1
sub_8048443	8048443	CADET_00001	2	1
sub_8048445	8048445	CADET_00001	32	2
sub_8048465	8048465	CADET_00001	32	2
sub_8048485	8048485	CADET_00001	38	2
sub_80484ab	80484ab	CADET_00001	26	2
sub_80484c5	80484c5	CADET_00001	20	2
sub_80484d9	80484d9	CADET_00001	26	2
sub_80484f3	80484f3	CADET_00001	27	1
sub_804850e	804850e	CADET_00001	34	3
_terminate	a000000	cle##kernel	0	1
transmit	a000001	cle##kernel	0	1
receive	a000002	cle##kernel	0	1
fdwait	a000003	cle##kernel	0	1
allocate	a000004	cle##kernel	0	1
deallocate	a000005	cle##kernel	0	1
random	a000006	cle##kernel	0	1

Figure 5: CFG based blocks generated by Angr

Function name	Segment	Star
sub_80	seg000	0000
sub_1A0	seg000	0000
sub_360	seg000	0000
sub_445	seg000	0000
sub_465	seg000	0000

Figure 6: CFG based block generated by IDA

4.3.2 Comparison of GA with Other Experimental Methods

First, we get the CFGs of the control group files, and the nodes and edges of the CFGs are shown in Table 2. Then take one of them and compare it with the CFGs of the control group. The experimental results are demonstrated

by similarity trend diagram, in which the solid line refers to the left Y axis and the dashed line refers to the right Y axis. In this paper, the smaller the result, the greater the similarity between the two graphs.

(2.1) Similarity between *loadimagefile* and contrast files.

CFG of the *loadimagefile* consists 8 nodes and 12 edges, and the similarity trends are shown in Figure 7. Results of GA and three comparative experimental methods show that the descending order of similarity with *loadimagefile* is $1 = 2 > 4 > 3 > 5 > 6 > 8 > 7$. However, for GED_linear and GED_distance, the edit distance between *loadimagefile* and itself is not zero, while GED and GA can get that there is no difference between *loadimagefile* and itself, and the two CFGs are exactly the same. It further shows that *loadimagefile* of kernel32.dll in Windows 10 is not modified, just the same as in Windows 7.

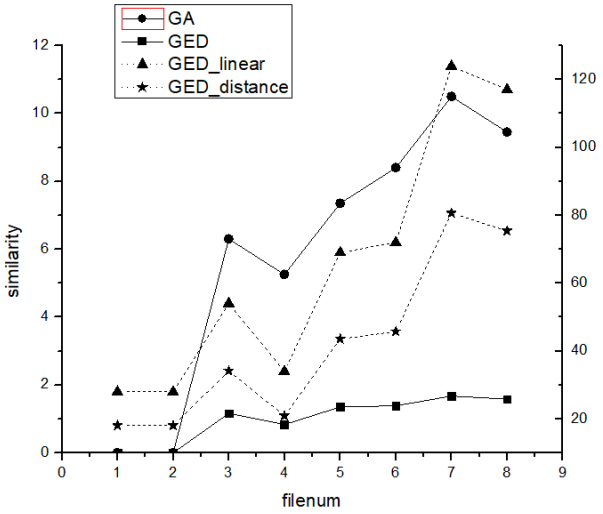


Figure 7: Similarity between *loadimagefile* and contrast files

The time performance of the four methods is shown as Figure 8. From Figure 8, it shows that the runtime of GA is relatively stable. But with the number of nodes increasing, the other three methods need more time. When the *loadimagefile* file similarity calculation with the file #8, the GA method needs the least running time, and both GED and GED_linear methods cost a long runtime. And on average, the running time of GED_linear is 9 times that of GA, GED 3 times, and GED_distance 0.6 times. But comparing with GED_distance, GA is more accurate and reasonable for identifying identical CFGs.

(2.2) Analysis of similarity between *loadappinitdlls* and contrast files.

The CFG of *loadappinitdlls* contains 17 nodes and 29 edges. As mentioned in Section 4.2, when the number of nodes is large, GED_linear cannot calculate the similarity (distance value) of two graphs. Just as in Table 3, we use NA to describe. Therefore, we mainly concern the other 3 methods and the similarity results are shown in Figure 9.

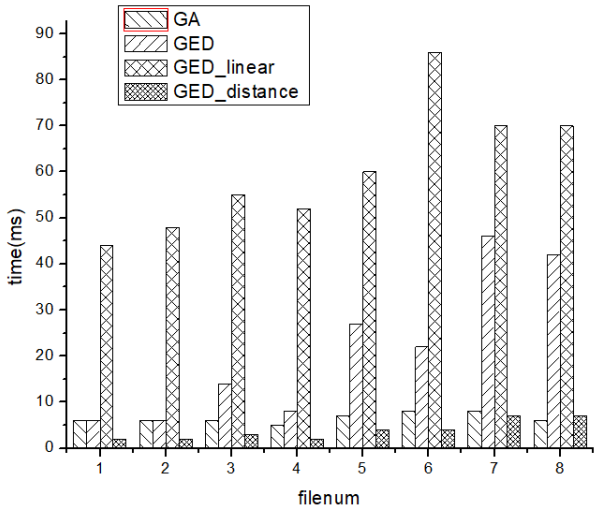


Figure 8: Time performances of four different methods for computing *loadimagefile*

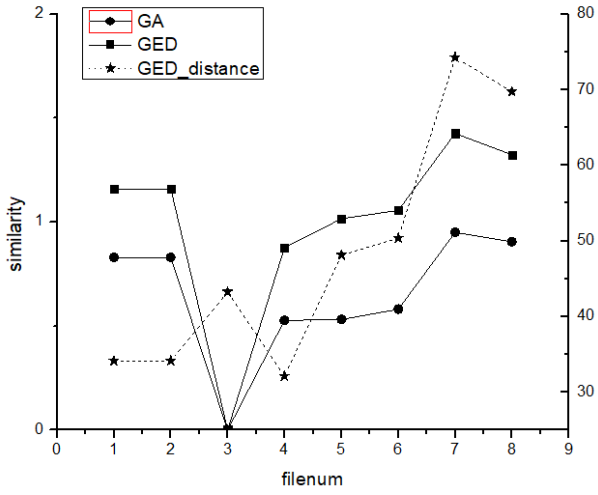


Figure 9: Similarity comparison between *loadappinitdlls* and contrast files

From Figure 9, it says that, for *loadappinitdlls*, the similarity order calculated by GED_distance is $4 > 1 = 2 > 3 > 5 > 6 > 8 > 7$, and the distance between *loadappinitdlls* and itself is not 0. When using GED, the similarity order is $3 > 4 > 5 > 6 > 1 = 2 > 8 > 7$, and *loadappinitdlls* is exactly the same as itself. For GA method, the result is consistent with the GED method.

Therefore, the results obtained, the results obtained by GED_distance and GED are inconsistent, and GA is consistent with that of GED. And when the two programs are exactly the same, the distance value should be 0, for instance, *loadappinitdlls* and file #3. Therefore, the results

Table 2: Nodes and edges of experimental files CFG

filenum(#)	1	2	3	4	5	6	7	8
Node number	8	8	17	10	24	25	48	45
Edge number	12	12	29	16	36	39	68	64

Table 3: Distance calculation by the GED_linear method

filenum (#)	filename	distance value
1	loadimagefile	46
2	Loadimagefile_7	46
3	loadappinitdlls	NA
4	Loadappinitdlls_7	50
5	basepappinitdlls	NA
6	Basepappinitdlls_7	NA
7	Write.exe	NA
8	Write_7.exe	NA

obtained by GA and GED are more reasonable. Further, the experiment shows that *loadappinitdlls* of kernel32.dll in Windows 10 is slightly modified from Windows 7.

When calculating the similarity between the *loadappinitdlls* and contrast files, the time performance of the three methods is shown in Figure 10. It can be seen that GA is less than that of both GED and GED_linear methods, and when the number of nodes is too many, the GED_linear cannot get result in a reasonable time. When calculating similarity between *loadappinitdlls* and the file #3, GA is significantly longer than that of GED_distance. And the main reason is file #3 is the *loadappinitdlls* file itself, and GA will evolve to a perfect match. When a chromosome fitness value is 0, the evolution stops. But for GED_distance, it is not 0. On average, running time for GED is 3 times that of the GA, and GED_distance 0.6 times.

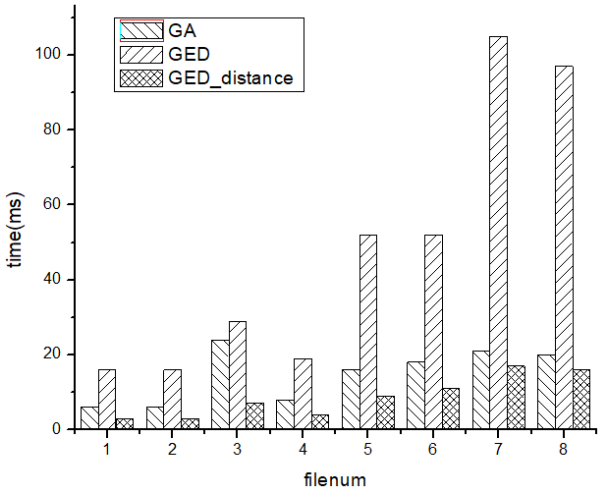


Figure 10: Time performance of 4 different methods for computing *loadappinitdlls*

(2.3) Similarity comparison between *loadappinitdlls_7* and contrast files.

The CFG of *loadappinitdlls_7* contains 10 nodes and 16 edges. The similarity diagram is as Figure 11. According to Figure 11, the results calculated by the two variations of editing distance show that the order of similarity with *loadappinitdlls_7* is $1 = 2 > 4 > 3 > 5 > 6 > 8 > 7$, and *loadappinitdlls_7* is not exactly the same as itself. For GED and GA method, the order of similarity with *loadappinitdlls_7* is $4 > 1 = 2 > 3 > 6 > 5 > 8 > 7$. So, for GA and GED, there is no difference between *loadappinitdlls_7* and itself, and two CFGs are identical.

Based on the results, the *loadappinitdlls_7* should be the most similar to itself, so the most similar file number should be 4. The results of the four methods show that the file with the least similarity to *loadappinitdlls_7* is the file #7. Therefore, it can be proved that the method proposed in this paper is reasonable to some extent.

For *loadappinitdlls_7*, the time performance of the four methods are shown in Figure 12. It can be seen that the time cost of the GED and GED_linear methods is gradually increasing with the increase of the number of nodes, but the trend of GA is almost the same as that of GED_distance. On average, the running time of GED_linear is 7 times that of GA, GED is 3 times, and GED_distance is 0.5 times.

5 Conclusions and Future work

In this paper, we proposed a genetic algorithm method to calculate the similarity between two binary files. First, the binary file is converted into CFG, and then the simi-

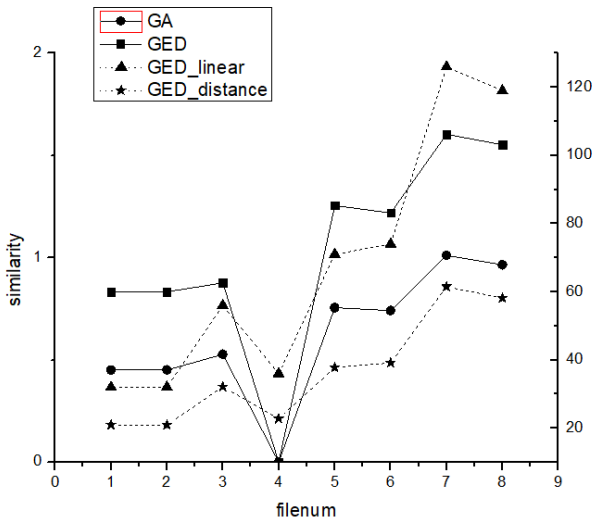


Figure 11: Similarity comparison between *loadappinitdlls_7* and contrast files

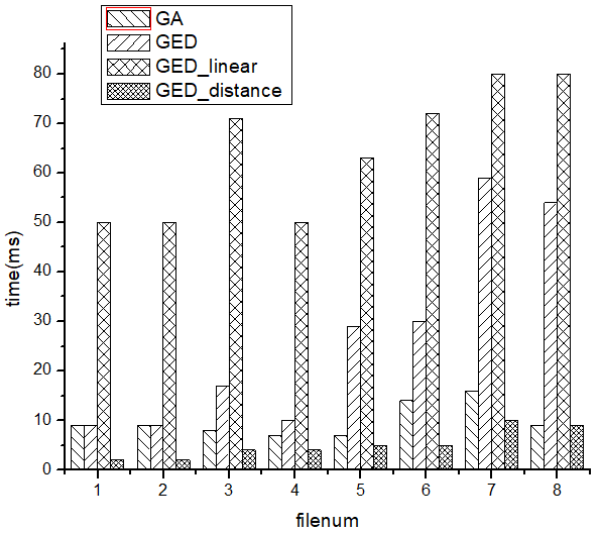


Figure 12: Time performance of 4 different methods for computing *loadappinitdlls_7*

larity between CFGs are calculated by GA. For similarity calculation, the proposed method is consistent with the GED method and can be used to accurately identify the isomorphism subgraph relationship and the exact identical CFGs, which is the basis of software homology detection.

GED is a very classical graphic similarity calculation algorithm. Compared with other methods, GED has the advantages of high accuracy and simple operation. The distance can be calculated simply by inputting 2 pairs of graphs. The similarity trend of GA in experiment (2.1) is almost the same as the three comparison algorithms, and the results of GA in experiment (2.2) and (2.3) are only the same as those of GED algorithm. Since the difference between the target file and itself should be equal to 0, we can infer that the results of GA and GED are more reasonable. Moreover, the proposed method is high efficient, especially for graphs with massive nodes. Also, according to the fitness function designed in this paper, GA can effectively identify whether the two graphs are isomorphic sub-graphs or exactly the same. About time performance, on average, GA is 0.3 times that of the GED, 0.1 times the GED_linear and 1.7 times the GED_distance.

About future work, some directions should be augmented. First of all, CFG is the analysis basis. How to enrich the CFG is a direction worth further study. After that, for GA based software homology detection, it should be reinforced in more real and wide areas applications. Meanwhile, some hyper-parameters of GA should be optimized automatically and adaptively according to different applications.

Acknowledgment

This work is partially sponsored by National Key Research and Development Program of China (2018YFB0704400, 2016YFB0700504, 2017YFB0701601), Research and Development Program in Key Areas of Guangdong Province (2018B010113001). The authors gratefully appreciate the anonymous reviewers for their valuable comments.

References

[1] T. Avgerinos, A. Rebert, K. C. Sang, and D. Brumley, "Enhancing symbolic execution with veritesting," in *International Conference on Software Engineering*, pp. 1083–1094, 2014.

[2] D. K. Chae, J. Ha, S. W. Kim, B. J. Kang, and E. G. Im, "Software plagiarism detection: A graph-based approach," in *ACM International Conference on Conference on Information Knowledge Management*, pp. 1577–1580, 2013.

[3] V. Chipounov, V. Georgescu, C. Zamfir, and G. Candea, "Selective symbolic execution," in *The Workshop on Hot Topics in System Dependability*, pp. 1286–1299, 2009.

[4] H. G. Choi, J. H. Kim, and B. R. Moon, "A hybrid incremental genetic algorithm for subgraph isomorphism problem," in *Conference on Genetic and Evolutionary Computation*, pp. 445–452, 2014.

[5] J. Choi, Y. Yoon, and B. R. Moon, "An efficient genetic algorithm for subgraph isomorphism," in *Conference on Genetic and Evolutionary Computation*, pp. 361–368, 2012.

[6] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen, and H. Bunke, "Approximation of graph edit distance based on hausdorff matching," *Pattern Recognition*, vol. 48, no. 2, pp. 331–343, 2015.

[7] J. Kim, H. G. Choi, H. Yun, and B. R. Moon, "Measuring source code similarity by finding similar sub-

graph with an incremental genetic algorithm," in *Genetic and Evolutionary Computation Conference*, pp. 925–932, 2016.

[8] K. Kim and B. R. Moon, "Malware detection based on dependency graph using hybrid genetic algorithm," in *Conference on Genetic and Evolutionary Computation*, pp. 211–1218, 2010.

[9] V. Kononov and V. A. Rusakov, "On the problems of developing klee based symbolic interpreter of binary files," *Procedia Computer Science*, vol. 145, pp. 275–281, 2018.

[10] R. Koschke, "Large-scale inter-system clone detection using suffix trees," in *European Conference on Software Maintenance and Reengineering*, pp. 309–318, 2012.

[11] J. Lerouge, Z. Abu-Aisheh, R. Raveaux, P. Héroux, and S. Adam, "Graph edit distance : A new binary linear programming formulation," *Computer Science*, vol. 72, pp. 254–265, 2015.

[12] H. I. Lim, "Comparing control flow graphs of binary programs through match propagation," in *IEEE Computer Software and Applications Conference*, pp. 598–599, 2014.

[13] Mudpo, "Analysis of computer virus epidemic situation in March 2018," *Information Network Security*, no. 5, 2018.

[14] L. Nan, H. Lifang, and X. Kunfeng, "An improved software source code matching algorithm based on abstract syntax tree," *Information Network Security*, no. 1, pp. 38–42, 2014.

[15] G. Negi, E. Elias, R. Kohli, and V. Bibhu, "Reliability analysis of test cases for program slicing," in *The 1st International Conference on Innovation and Challenges in Cyber Security (ICICCS'16)*, pp. 36–40, 2016.

[16] H. Qin, *Study on Software Homology Detection Technology Based on AST Structure Optimization and CFG Comparison (in Chinese)*, Master Thesis, Beijing University of Posts and Telecommunications, 2011.

[17] Q. W. Qin, "Reverse analysis of software based on ida-Pro," *Computer Engineering*, vol. 34, no. 22, pp. 86–88, 2008.

[18] K. Sen, P. Godefroid, N. Klarlund, "DART: Directed automated random testing," *ACM Sigplan Notices*, vol. 40, no. 6, pp. 213–223, 2005.

[19] K. Sen, D. Marinov, and G. Agha, "Cute: A concolic unit testing engine for C," *ACM Sigsoft Software Engineering Notes*, vol. 30, no. 5, pp. 263–272, 2005.

[20] M. H. R. A. Shaikhly, H. M. El-Bakry, and A. A. Saleh, "Cloud security using markov chain and genetic algorithm," *International Journal of Electronics and Information Engineering*, vol. 8, no. 2, pp. 96–106, 2018.

[21] H. Shi, J. Mirkovic, and A. Alwabel, "Handling anti-virtual machine techniques in malicious software," *ACM Transaction on Privacy and Security*, vol. 21, no. 1, 2018.

[22] G. Tao, G. Dong, Q. Hu, and B. Cui, "Improved plagiarism detection algorithm based on abstract syntax tree," in *International Conference on Emerging Intelligent Data Web Technologies*, pp. 714–719, 2013.

[23] M. Weixuan, C. Zhongmin, and T. Li, "A malicious code detection method based on active learning," *Software Journal*, vol. 28, no. 2, pp. 384–397, 2017.

[24] P. Wu, J. Wang, and B. Tian, "Software homology detection with software motifs based on function-call graph," *IEEE Access*, no. 99, pp. 1–1, 2018.

[25] Y. Xiang, J. Han, H. Xu, and X. Guo, "An improved heuristic method for subgraph isomorphism problem," *IOP Conference Series: Materials Science and Engineering*, vol. 231, no. 1, pp. 012050, 2017.

[26] L. Xu, F. Sun, and Z. Su, "Constructing precise control flow graphs from binaries," *University of California*, 2012. (`https://pdfs.semanticscholar.org/8a80/f0d173ec7420478e4b96a8264e21e0dafac0.pdf`)

[27] S. Yan, R. Wang, C. Salls, N. Stephens, M. Polino, A. Dutcher, J. Grosen, S. Feng, C. Hauser, and C. Kruegel, "Sok: (State of) the art of war: Offensive techniques in binary analysis," in *Security and Privacy*, pp. 138–157, 2016.

[28] J. Yang, X. Song, Y. Xiong, and Y. Meng, "An open source software defect detection technique based on homology detection and pre-identification vulnerabilitys," *Advances in Intelligent Systems and Computing*, vol. 773, pp. 932–940, 2019.

[29] Y. Zhibin, J. Xin, and S. Dawei, "A binary-oriented mixed recovery method for control flow graphs," *Computer Application Research*, no. 7, 2018.

[30] X. Zhoubo, Z. Li, Ninglihua, and A. Tianlong, "Graphic editing distance summary," *Computer Science*, vol. 45, no. 4, pp. 11–18, 2018.

Jinyue Bian is a master degree student in the school of computer science, Shanghai University. His research interests include big data analysis, machine learning, and computer and network security especially in host behaviour analysis.

Quan Qian is a full Professor in Shanghai University, China. His main research interests concerns computer network and network security, especially in cloud computing, big data analysis and wide scale distributed network environments. He received his computer science Ph.D. degree from University of Science and Technology of China (USTC) in 2003 and conducted postdoc research in USTC from 2003 to 2005. After that, he joined Shanghai University and now he is the dean of department of machine intelligence and technology, and the lab director of materials informatics and data science.

A New Diffusion and Substitution-based Cryptosystem for Securing Medical Image Applications

L. Mancy and S. Maria Celestin Vigila
(Corresponding author: S. Maria Celestin Vigila)

Department of Information Technology, Noorul Islam University
Kumaracoil, Tamilnadu, India
(Email: mancy1989@gmail.com)

(Received Mar. 21, 2017; Revised and Accepted June 26, 2018; First Online July 11, 2019)

Abstract

In recent periods, individual security is becoming increasingly endangered. Several methods of safeguarding person are private, economic or health data are developed by entities, fabrications, and managements. Thus the patient data are generally produced in the place of the health images for observing. Thus, it is effortlessly available to each one. The patient data obviously shown may be interrupted by other parties during automated broadcast. For analysis purposes, these health informatics desires to be voluntarily nearby to the physicians. Unique feature of this scheme is to make a revision on the Digital Imaging and Communications in medicine (DICOM) standard, which specifies the form that all numerical health images will be well-suited. Several private data, such as patient's name, date of birth, gender, and patient identity with details about where the image was taken. So it is essential from the patient's idea of sight is to keep the above evidence in private. Therefore the security of medical images can be achieved through confidentiality, availability, reliability and authentication.

Keywords: Diffusion; Encryption; Histogram; Steganography; Substitution

1 Introduction

Cryptography has developed a communal, to afford an extraordinary defense against various attacks. It diminishes with the progression of systems for altering facts among reasonable and worthless practices. In endangered dispatches the cryptographic methods are focused by one or more keys. When the keys are same, are convoked as private key cryptography. Consecutively, when keys are different, then the cryptographic methods are known as public key cryptography. There are two vital assets which all encryption schemes must fulfill. The first is the confusion assets which involves, that cipher texts should have random advent. The second is the diffusion assets, which requires that alike keys should yield entirely contradictory cipher texts for the similar plaintext.

Digital imageries like hypnotic quality images, X-rays images, etc. are generally used in remedial solicitations. It compact with patient proceedings that are remote and should only obtainable to official folks. Though certain patients are unworried about rupture of privacy could root risky awkwardness and disgrace. Therefore, there is a necessity to defend and sustain privacy of patient information. In this dispatch, an endangered image encryption for DICOM images, based on the substitution and diffusion transformations is suggested. A secret key of 128-bits size is generated by an image a histogram. Initially, the visual feature of DICOM image is decomposed by the mixing process. The subsequent image is divided into key reliant blocks and further, these blocks are passed through diffusion and substitution processes. Total five rounds are used in the encryption method. Finally the generated secret key is embedded within the encrypted image in the process of steganography. At the receiver side the secret key was recovered from the embedded image and decryption operation was performed in inverse format. Here the Steganography is the art of secreting data by means that avoid the recognition of secret communications. It contains a huge amount of approaches to coat a communication from being grasped. The main aim of steganography is to distort the presence of any secreted statement.

The above introductory section explains the medical image security and provides some security measures to overcome the problem. The second section presents a review of literature and relevant research associated with the problem addressed in this study. Then the third section explains the methodology and the steps of the proposed cryptographic algorithm. Finally the fourth section explains the analysis of the data and the presentation of the results. Then the final section explains the conclusion

of medical image security applications.

2 Related Works

In the appraisal of works several researches have raised the safety of medicinal images in their own idea. Certain appropriate mechanism is highlighted in this section.

Zhou *et al.* introduced a technique certainty and reliability of digital mammography to encounter the necessities of authenticity and integrity of images [36]. Zhang *et al.* (2007) an image scrambling technique was established on queue transformation. Zhang *et al.* (2007) implemented a secured digital communication protocols under various conditions. Zhicheng *et al.* (2008) suggested a novel lossless data embedding technique. Michal Voss berg *et al.* (2008) acclimate the procedure to the globes grid safekeeping administration. Gouenou Coatrieux *et al.* (2009) familiarized to make the image more serviceable, by watermarking it with a precis data. Yong Feng *et al.* (2009) presented an invertible map, called Line map, for image crypto system.

Luiz Octavio Massato Kobayashi *et al.* (2009) a method using cryptography to improve conviction of medical images. Weihai Li *et al.* (2009) Peizhenwang *et al.* (2010) a better image encryption technique, which is based on hyper chaotic categorization is anticipated. Vinod Patidar *et al.* (2010) a diffusion-substitution system, based on chaotic map, for the image encryption is recommended. Sanfu Wangl *et al.* (2010) offered a image scrambling technique. Yi Wan *et al.* (2010) In this paper, a histogram based vigorous estimator for the noise mixing prospect was projected. Guiliang Zhul *et al.* (2010) recommended the three levels of multilayer scramble. Stallings (2010) has documented about security issues in his book. Sathish kumar *et al.*(2011) suggested anew algorithm for the image cryptographic system.

Thomas Neuberger *et al.* (2011) safeguards the medical accounts from prohibited access in which the patient as material container to resolve, who are the certified persons. Mustafa Ultras *et al.* (2011) authorize a secret scattering system, which segments the health images among a health team of 'n' doctors. Maria Celestin Vigila and Muneeswaran (2012) projected an Elliptic curvature based key generation for stream cipher. Dalel Bouslimi *et al.* (2012) advocate a mutual encryption watermarking method to combine the Quantization index modulation and an encryption algorithm. Abir Awad *et al.* (2012) a novel chaotic replacement method based on the complementary rule.

Musheer Ahmad and Tanvir Ahmed (2012) recommended a framework to provide visual protection to withstand statistical attacks to medical images. Li Chin Huangc *et al.* (2013) put forward a histogram fluctuating method to reach high bit depth health images. Tsang IngRen *et al.* (2013) estimated a motion recompense method to be applied in both encryption and decryption process. Maria Celestin Vigila and Muneeswaran (2013) the Elliptic curve cryptosystem for a text message was implemented. Chong Fu *et al.* (2013) offer a chaos medical image encryption pattern, through a bit-level shuffling process. Viswanathan *et al.* (2014) advocated by giving admission to the results of medical images with secrecy, convenience and veracity. Abhilasha Sharma *et al.* (2015) direct the watermarking method based on DWT to embed multiple watermark into the cover image. Jani Anbarasi *et al.* (2015) anticipated a multi top-secret image sharing scheme that shares the multiple disruption polynomial.

Mancy and Maria Celestin Vigila (2015) reviewed on image encryption technique and their functionalities are analyzed. Maria Celestin Vigila and Muneeswaran (2015) proposed the implementation of reversible information hiding in spatial domain images rooted in neighbor mean image interruption without impairing the image eminence. ZeinabFawaz *et al.* (2016) a novel image encryption scheme based on two rounds of substitution–diffusion is proposed. Ritu Agrawal *et al.* (2016) uses a lossless medical image watermarking method using modulation variation was implemented, which offers high healthiness and low entropy distance for safeguard. Akram Belazi *et al.* (2016) a novel image encryption approach based on permutation substitution network and chaotic systems are proposed. BalaKrishnan Ramalingam *et al.* (2017) present permutation and diffusion based hybrid image crypto system in transform domain using combined chaotic maps and Haar Integer Wavelet Transform. Weijia Cao *et al.* (2017) presents a medical image encryption algorithm using edge maps derived from a source image. Shahryar Toughi *et al.* (2017) utilize the Elliptic curve generator to generate a sequence of arbitrary numbers based on curves.

3 The Proposed Methodology

The medical image safety has become a significant problem during the storage and broadcast of data. So to preserve the conveyed proof beside undesirable description, the secret key used in encryption process is generated from the image itself. By mixing process, the distinctive pixel is extended by associating the present pixel with its former pixel and its session key. Here the size of the block is decided by session key, in which the block may be of any one of the ten different sizes. Then in diffusion process, the pixel of each block is reorganized within the block by a spiral path pattern. In the substitution process, the pixel of each block is changed with one of their nearby pixels. Finally the generated secret key is embedded into the cipher image using the process of steganography.

To secure the medical DICOM image, the secret key of 128 bit size is generated by an image histogram. Then the medicinal image is encoded by using diffusion and substitution procedure. Total five rounds are used in the encryption method. At last the secret key is fixed within the encrypted image by the method of steganography. At the receiver side the key was recovered from the embedded

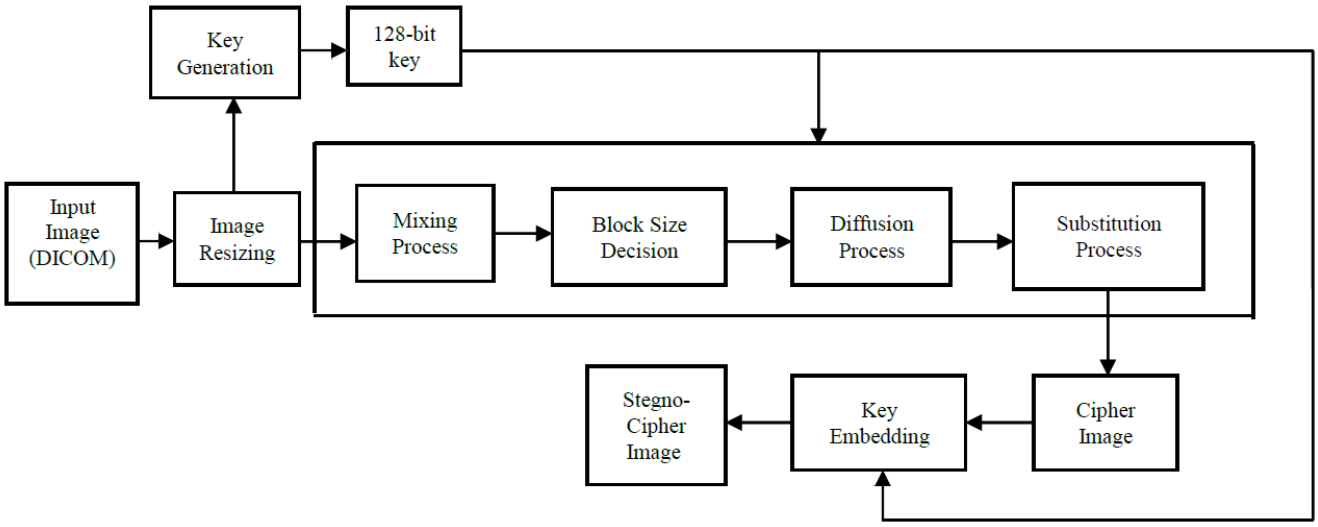


Figure 1: Block diagram of the proposed methodology

image and then decryption operation is performed in the inverse setup. Fig 1 shows the block diagram of proposed approach.

The steps of this proposed cryptographic algorithm is explained as follows.

1) Key generation:
The private key assets in the encryption process are generated from the image itself.

Step 1. At first the histogram of the image is calculated by means of histogram counts.

Step 2. Then histogram counts of 255 values are obtained by dividing the i^{th} count by $i+1^{th}$ count.

Step 3. The obtained result is rounded to the nearest integer and modulo10 operation is performed on the values.

Step 4. The resulting values will be in the range of [0-9].

Step 5. From the 255 values, the first 128 values are extracted and formed as a secret key with a size of 128 bits.

Step 6. Then this key is divided into block of 8 bits to form the session keys $k = k_1 k_2 \cdots k_{32}$ (in hexadecimal) given as $K = K_1 K_2 \cdots K_{16}$ (in ASCII).

Here, k_i's referred to as sub-keys are hexadecimal digits (0-9 and A-F) and K_i represents the session keys.

2) Mixing Process:
In mixing process every pixel of the image is interchanged with its earlier pixels and the session key by XOR operation. The algorithm related to mixing process is given in Algorithm 1. In this algorithm, 'H' and 'W' represent the Highness and Extensiveness of the image respectively; $P_{x,0} = \{0$ when $x = 1$ & $P_{x-1,W}$ when $x > 1\}$.

Algorithm 1 The mixing process

```
i ⟵ position of first session key, i.e. 1
for x = 1: H do do
  for y = 1: W do do
    P_{x,y} ⟵ (P_{x,y} ⊕ P_{x,y−1} ⊕ K_i)
    i ⟵ position of next session key, i.e. ((i mod 16) + 1)
  end for
end for
```

3) Block Size Decision:
The resulting image from the mixing process is divided into non-coinciding blocks B 1, B2....BN. Here the size of the block is decided by session key Ki .The block may be of any one of the ten different sizes. So, total five rounds are used to complete the encryption process. In each round unique session key Ki is used. The table1 shows the block size decision criteria.

4) Diffusion Process:
Here the scrambling of the image is based on spiral scanning pattern. Here the scrambling is done in a key dependent manner. The Pixel of each block is replaced within the same block by a spiral path of size 8×8 matrix. Location of the starting pixel for navigating in a block is made key reliant on exclusively. For this purpose pairs of neighboring sub keys are formed. The Key pairs are given as (k_1, k_2), (k_3, k_4),(k_5, k_6), (k_7, k_8) and (k_1, k_2). When all sub-key pairs are firmed it is commenced again from the first sub-key pair (k_1, k_2). At the end of diffusion processes, not only all pixels get altered, but also their neighboring pixels are rearranged broadly within the

Table 1: Block size decision table

K_i mod 10	0	1	2	3	4	5	6	7	8	9
Block Size (B_i)	16	24	32	40	48	56	64	72	80	96

image block. The 8×8 matrix diagram is given in Figure 2.

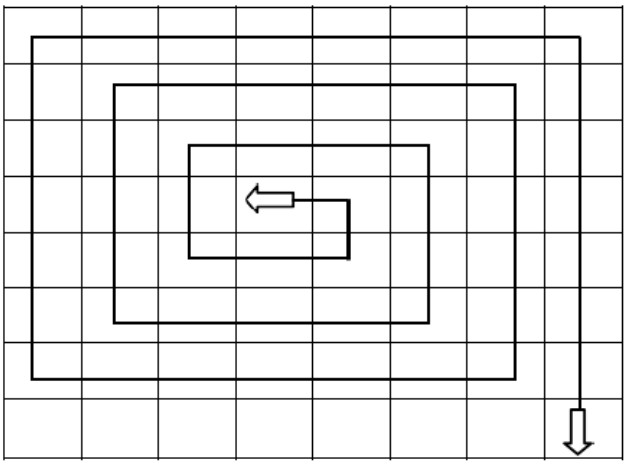


Figure 2: Spiral path Scrambling

5) Substitution Process:
In substitution process the property of the selected pixel is altered by the one of the neighboring pixels in the eight directions (i.e.) the current pixel is XOR-ed with the neighboring pixel. Therefore the eight neighboring pixels are North, North West, North East, South, South West, South East, East (E) and West (W). Here, we use neighboring as $P_{x,y}$ which is XOR-ed with current pixel $P_{i,j}$. When all the sub-keys are shattered, begin the procedure from the first sub-key k_1 again. In this step, some of the pixels lying on the boundary of a block may be remain unaffected. The pixel location table is given in Table 2.

6) Key Embedding:
The generated secret key is embedded into the cipher image using DWT transform, which is applied to the image to form four sub bands such as LL, LH, HL and HH. The LL level is obtained to the size of 128×128. Then the transformed coefficients in all the bands are converted into the binary form and along the diagonal elements the 3 LSB is replaced with each bit of the secret key after converting each digit in binary form with three digits. Then after implanting the secret key is transformed into decimal format and again, it is converted into original format by IDWT. The ultimate output is the embedded image and the secret key is mended at the receiver side through extraction process and the image is decrypted in the reverse order as done in the encryption stage.

Table 2: Pixel location table

Sub Key value k_i	Directions of adjacent surrounding pixel to a pixel	Location Surrounding pixel $P_{x,y}$ w.r.t current pixel
0/F	E	$P(i, j+1)$
1/E	NE	$P(i-1, j+1)$
2/D	N	$P(i-1, j)$
3/C	NW	$P(i-1, j-1)$
4/B	W	$P(i+1, j-1)$
5/A	SW	$P(i+1, j)$
6/9	S	$P(i+1, j)$
7/8	SE	$P(i+1, j+1)$

4 Performance Analysis

The current scheming power is capable of breaking encryption patterns in a real time, if the scheme is not designed to look into these problems. Hence, a good encryption system would preserve away from the possible attacks. Hence, analysis of encryption systems such as histogram analysis, Entropy, Correlation analysis, etc., certifies the precise growth of the security scheme.

1) Statistical Analysis:
The statistical analysis involves collecting every data sample in a set of items from which samples can be drawn. The different types of statistical analysis are given as follows.

   a. Histogram Analysis:
   A histogram is a graphical design of numerical data. An image histogram is a chart that shows the dispersal of intensities in a grayscale image. To prevent the outflow of data to an opponent, it is substantial to guarantee that the cipher image does not have any statistical resemblance to the input image. The histogram of the input image has massive sting. But, the histogram of the cipher image is nearly even and constant, signifying the nearly same probability of existence of each intensity level. Figure 3 shows the histogram of original, cipher and stegno image.

   b. Information Entropy:

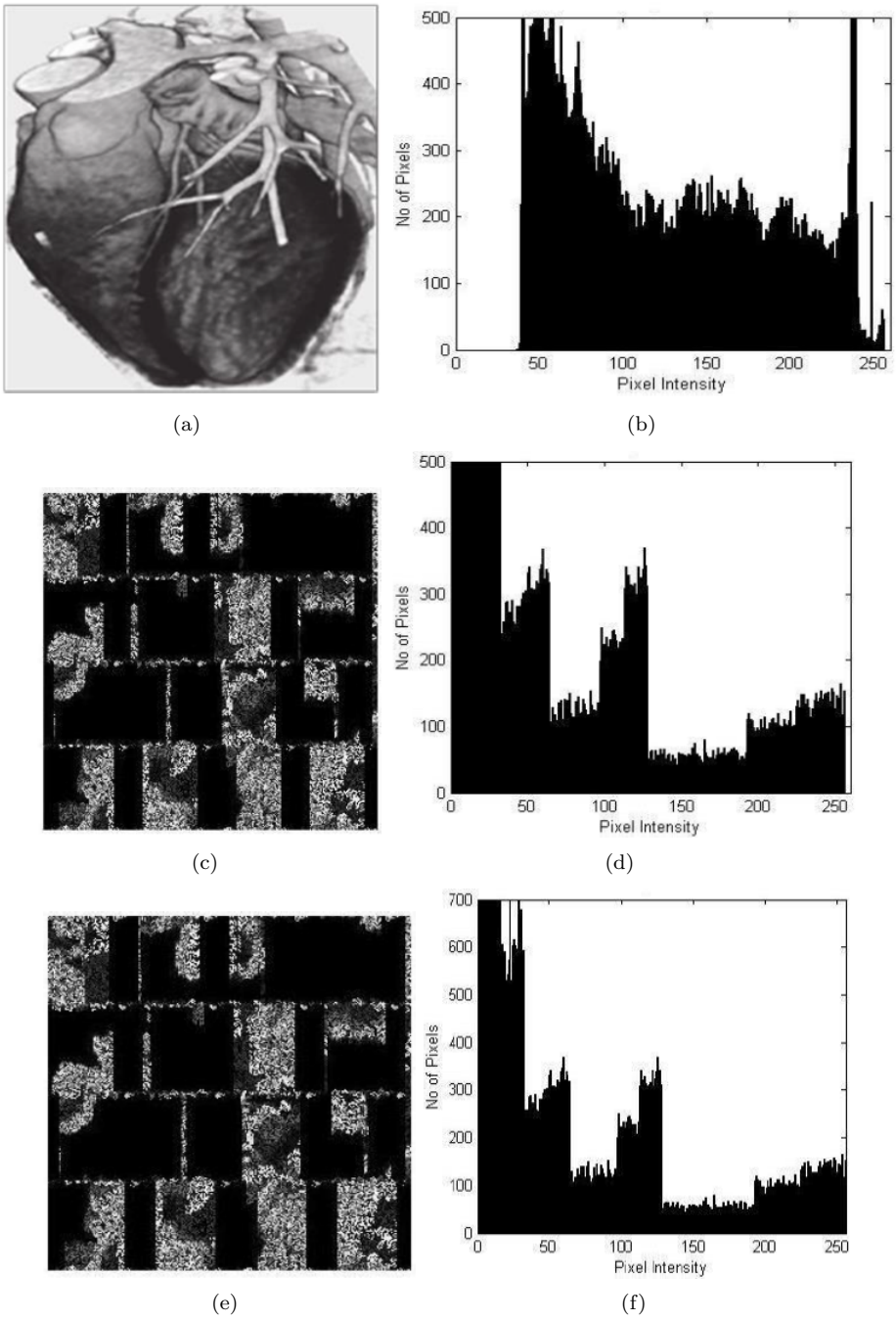


Figure 3: Histogram of original, encrypted, and stegno-cipher Images. (a)Original Image, (b) Histogram of Original Image, (c) Encrypted Image , (d) Histogram of Encrypted Image, (e) Stegno-Image, (f) Histogram of Stegno Image.

Information entropy is a concept from information theory. It tells how much information there is at an event. In general, the more uncertain or random the event is, the more information it will contain. It has applications in many areas, including lossless data compression, statistical inference, cryptography, etc. The Entropy calculation formula gives as follows,

$$H(s) = \sum P(S_i) \log 2 \times (1/P(S_i)). \quad (1)$$

Where $P(S_i)$ is the probability of an i th image. Table 3 shows the entropy of original, cipher and stegno image. To enterprise a good image encryption pattern, the entropy of encrypted image should be closer to the ultimate expected value 8. Therefore the information outflow in the proposed cipher is insignificant, and it is safe upon the entropy outbreak.

2) Correlation Analysis:
The correlation is an arithmetic method that shows

Table 3: Entropy of original, cipher and plain image

Image	Entropy
Original Image	7.2551
Cipher Image	7.5564
Stegno Image	7.6463

whether and how powerfully the couples of variables are connected. The correlation between pairs of original and its corresponding encrypted image produced using the proposed image encryption algorithm by computing the correlation coefficients. The formula related to correlation coefficients are given as follows.

$$\begin{aligned} C_{x,y} &= cov(x,y)/\sqrt{D(x)}\sqrt{D(y)} \\ E(x) &= 1/N \times \sum x_i \\ D(x) &= 1/N \times \sum (x_i - E(x))^2 \\ Cov(x,y) &= 1/N \times \sum (x_i - E(x))(y_i - E(y)). \end{aligned}$$

In the given formula, there are N pairs of neighboring pixels and x, y represent the intensity values of two neighboring pixels from an image. Table 4 shows the correlation of original, cipher and stegno images.

3) Key Sensitivity Analysis:
Even a variation in a single bit of the key will make an entirely different cipher image for the attackers to identify the key. This makes the encryption procedure sensitive enough to the secret key. To test the sensitivity of the proposed image cipher with respect to the key, encrypted image corresponding to plain image is decrypted with a slightly different key than the original one. Here the two cipher images are associated, in which it was not easy to compare the cipher images by simply observing these images. Thus, for comparison, the correlation between the identical pixels of the two cipher images is intended. Table 5 shows the entropy and correlation between two cipher images.

Here Figure 4 shows the Key Sensitivity Analysis of original and two cipher images. Therefore a perfect diffusion and substitution method should resist against all kinds of attacks. Hence some analysis techniques such as statistical, correlation and key sensitivity analysis are discussed in the above session to prove that the proposed algorithm is secretive against most common attacks. Therefore a new diffusion and substitution based cryptosystem for securing the medical image applications is implemented and performance analysis designates that the proposed cipher is more secure.

5 Conclusion

To secure the medical image, the secret key of 128 bit size is generated by means of image histogram and the medical image is encrypted by using diffusion and substitution process. Finally the secret key is embedded within the encrypted image in the process of steganography. This also enriched the security of medical image. At the receiver side the key was recovered from the embedded image and then decryption is performed in the reverse format. Due to this, the security of medical image is enriched. Here we deliberate the security analysis of medical image encryption pattern such as statistical analysis, correlation analysis and key sensitivity analysis, which prove that the proposed cipher is safe against the most common attacks.

References

[1] R. Agrawala, M. Sharma, "Medical image watermarking technique in the application of diagnosis using M-ary modulation," in *International Conference on Computational Modeling and Security*, 2016.

[2] M. Ahmed, T. Ahmed, "A framework to protect patient digital imagery for secure telediognosis," *Procedia Engineering*, vol. 38 , pp. 1055-–1066, 2012.

[3] J. S. Anbarasi, A. G. S. Malab, M. Narendrac, " DNA based multi-secret image sharing," in *International Conference on Information and Communication Technologies*, 2015.

[4] A. Awad and A. Miri, "A new image encryption algorithm based on a chaotic dna substitution method," in *IEEE International Conference on Communications (ICC'12)* , 2012.

[5] A. Belazi, A. A. Abd El-Latif, S. Belghith, "A novel Image Encryption Scheme based on Substitution-Permutation network and chaos," *Signal Processing*, vol. 128, pp. 155–170, 2016.

[6] D. Bouslimi, G. Coatrieux, M. Cozic, C. Roux, "A joint encryption/watermarking system for verifying the reliability of medical images," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, pp. 5, pp. 891–899, 2012.

[7] W. Cao, Y. Zhou, C. L. P. Chen, L. Xia, "Medical image encryption using edge maps," *Signal Processing*, vol. 132, pp. 96–109, 2017.

[8] G. Coatrieux, C. Le Guillou, J. M. Cauvin, "Reversible watermarking for knowledge digest embedding and reliability control in medical images," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 2, pp. 158—165, 2009.

[9] Z. Fawaz, H. Noura and A. Mostefaoui, "An efficient and secure cipher scheme for images confidentiality preservation," *Signal Processing*, vol. 42, pp. 90–108, 2016.

[10] Y. Feng, X. Yu, "A novel symmetric image encryption approach based on an invertible two-dimensional map," in *35th Annual Conference of IEEE Industrial Electronics*, pp. 4244–4649, 2009.

Table 4: Correlation of original, cipher and stegno images

Image	Vertical Correlation	Horizontal Correlation	Diagonal Correlation
Original Image	-0.0053	-0.0075	-0.0068
Cipher Image	-0.0063	-0.0042	-0.0130
Stegno Image	-0.0089	-0.0032	-0.0073

(a) (b) (c)

Figure 4: Key sensitivity analysis of original, cipher image 1 and cipher image 2. (a)Original Image, (b) Cipher Image 1, (c) Cipher Image 2.

Table 5: Entropy and Correlation between two cipher images

Image	Entropy	Vertical Correlation	Horizontal Correlation	Diagonal Correlation
Cipher Image 1	7.5564	-0.0063	-0.0042	-0.0130
Cipher Image 2	7.6463	-0.0089	-0.0032	-0.0073

[11] C. Fu, W. H. Meng, Y. F. Zhan, *et al.*, "An efficient and secure medical image protection scheme based on chaotic maps," *Computers in Biology and Medicine*, vol. 43, pp. 1000–1010, 2013.

[12] L. C. Huang, L. Y. Tseng, M. S. Hwang, "A reversible data hiding method by histogram shifting in high quality medical images," *The Journals of Systems and Software*, vol. 86, pp. 716–727, 2013.

[13] L. O. M. Kobayashi, S. S. Furuie, P. S. L. M. Barreto, "Providing integrity and authenticity in DICOM images: A novel approach," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 582–589, 2009.

[14] W. Li, Y. Yuan, " Improving security of an image encryption algorithm based on chaotic circular shift," in *IEEE International Conference on Systems and Cybernetics*, 2009.

[15] L. Mancy, S. M. C. Vigila, "A survey on protection of medical images," in *International Conference on Instumentation, Communication and Computational Technologies*, 2015.

[16] T. Neubauera, J. Heurixb, "A methodology for the pseudonymization of medical data," *International Journal of Medical Informatics*, vol. 80, pp. 190–204, 2011.

[17] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, X. Lin, "Robust lossless image data hiding designed for semi fragile image authentication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 4, pp. 497—509, 2008.

[18] V. Patidar, G. Purohit, K. K. Sud, N. K. Pareek, "Image encryption through a novel permutation-substitution scheme based on chaotic standard map," in *International Workshop on Chaos-Fractal Theory and Its Applications*, 2010.

[19] B. Ramalingam, A. Rngarajan, J. B. B. Rayappan, " Hybrid image crypto system for secure image communication - A VLSI approach," *Microprocessor and Micro Systems*, vol. 50, pp. 1–13, 2017.

[20] C. C. Sabino, L. S. Andrade, T. I. Ren, G. D. C. Cavalcanti, T. I. Jyh, J. Sijbers, "Motion compensation techniques in permutation-based video encryption," in *IEEE International Conference on Systems and Cybernetics*, 2013.

[21] G. A. Sathishkumar, K. B. Bagan, N. Sriraam, "Image encryption based on diffusion and multiple chaotic maps," *International Journal of Network Security & Its Applications*, vol. 3, pp. 2, pp. 181—194, 2011.

[22] A. Sharma, A. K. Singh, S. P. Ghrera, "Secure hybrid robust watermarking technique for medical images," in *International Conference on Eco-friendly Computing and Communication Systems*, 2015.

[23] W. Stallings, *Cryptography and Network Security*, Fifth Edition, 2010.

[24] S. Toughi, M. H. Fathi, Y. A. Sekhavat, " An image encryption scheme based on elliptic curve pseudo random and advanced encryption system," *Signal Processing*, vol. 141, pp. 217–227, 2017.

[25] M. Ulutas, G. Ulutas, V. Nabiyev, "Medical image security and EPR hiding using Shamir's secret sharing scheme," *The Journals of Systems and Software*, vol. 84, pp. 341–353, 2011.

[26] S. M. C. Vigila, K. Muneeswaran, "Key generation based on elliptic curve over finite prime field," *International Journal on Electronic Security and Digital Forensics*, vol. 4, pp. 1, pp.65—81, 2012.

[27] S. M. C. Vigila, K. Muneeswaran, "A new elliptic curve cryptosystem for securing sensitive data applications," *International Journal on Electronic Security and Digital Forensics*, vol. 5, pp. 1, 2013.

[28] S. M. C. Vigila, K. Muneeswaran, "Hiding of confidential data in spatial domain images using image interpolation," *International Journal of Network Security*, vol. 17, No. 6, pp. 722–727, 2015.

[29] P. Viswanathan, V. P. Krishna, "A joint FED watermarking system using spatial fusion for verifying the security issues of teleradiology," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, 2014.

[30] M. Vossberg, T. Tolxdorff, "DICOM Image Communication in Globus-Based Medical Grids," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, pp. 2, pp. 145—153, 2008.

[31] Y. Wan, Q. Chen, Y. Yang, "Robust impulse noise variance estimation based on image histogram," *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 485—488, 2010.

[32] P. Wang, H. Gao, M. Cheng, X. Ma, "A new image encryption algorithm based on hyperchaotic mapping," in *International Conference on Computer Application and System Modeling*, 2010.

[33] S. Wang, Y. Zheng, Z. Gao, " A new image scrambling method through folding transform," in *International Conference on Computer Application and System Modeling*, 2010.

[34] H. Y. Zhang, "A new image scrambling algorithm based on queue transformation," in *International Conference on Machine Learning and Cybernetics*, pp. 19–22, 2007.

[35] J. Zhang, F. Yu, J. Sun, Y. Yang, C. Liang, "DICOM image secure communications with internet protocols IPv6 and IPv4 ," *IEEE Transactions on Information Technology in Bio medicine*, vol. 11, no. 1, pp. 70—80, 2007.

[36] X. Q. Zhou, H. K. Huang, and S. L. Lou, "Authenticity and integrity of digital mammography images," *IEEE Transaction on Medical Imaging*, vol. 20, pp. 8, pp. 784—791, 2001.

[37] G. Zhu, W. Wang, X. Zhang, M. Wang, " ZGW-1 digital image encryption algorithm based on three levels and multilayer scramble," in *2nd IEEE International Conference on Network Infrastructure and Digital Content*, 2010.

Biography

S. Maria Celestin Vigila completed her B.E. in Computer Science and Engineering in 1996 and M.E. in Computer Science and Engineering in 1999. She completed her Ph.D. in the area of data security from Anna University, Chennai. She is currently Associate Professor in the Department of Information Technology, Noorul Islam University, Kumaracoil and member of ISTE and IET. She is the reviewer for quite a few peer reviewed international journals. Her research interest includes cryptography and network security, wireless networks and information hiding.

A LWE-based Oblivious Transfer Protocol from Indistinguishability Obfuscation

Shanshan Zhang[1,2]

(Corresponding author: Shanshan Zhang)

State Key Laboratory of Integrated Services Networks, Xidian University[1]
No. 2, Taibai South Road, Xi'an 710071, Shaanxi Province, China
School of Mathematics and Information Science, Baoji University of Arts and Sciences[2]
No. 44, Baoguang Road, Baoji 721013, Shaanxi Province, China
(Email: sszhang0801@163.com)

(Received Mar. 18, 2019; Revised and Accepted Aug. 4, 2019; First Online Jan. 29, 2020)

Abstract

Oblivious transfer is an important cryptographic primitive and served as a powerful tool in secure computation. Most existing oblivious transfer protocols are built upon the hardness of factoring or computing discrete logarithm problem. However, threatened by quantum computing, these protocols will be broken down directly in the presence of quantum computer. Therefore, it is essential to construct OT protocol based on post-quantum cryptography. As a subarea of post-quantum cryptography, lattice-based cryptography has some attractive features. Specifically, the learning with errors (LWE) problem has been used as an amazingly versatile basic tool to design cryptographic schemes. We are inspired by a result which proposed an oblivious transfer protocol using the decisional Diffie-Hellman assumption and indistinguishable obfuscation. Therefore, we propose a new secure LWE-based oblivious transfer protocol from indistinguishability obfuscation. The main tools consist of LWE-based dual-mode cryptosystem and a secure indistinguishability obfuscation which guarantee the security of our oblivious transfer protocol.

Keywords: Dual-mode Cryptosystem; Indistinguishability Obfuscation; Lattice-based; Oblivious Transfer

1 Introduction

Oblivious transfer (OT) is a fundamental cryptographic primitive, first proposed by Rabin [13] in 1981. It contains two participants, a sender (denoted by **S**), and a receiver (denoted by **R**), and requires that **S** sends a message to **R** with probability 1/2, while **S** is oblivious to whether or not the message was received by **R**. A well-known flavor of OT is called 1-out-of-2 OT (denoted by OT_1^2), where **S** has two inputs m_0 and m_1, and **R** has a chosen bit $b \in \{0,1\}$, and **R** wishes to obtain m_b, without **S** learning b, while **S** wants to ensure that **R** receives only one of the two messages. Due to the simple functionality of OT, it has been widely exploited to construct cryptographic schemes, such as contract signing, secure multi-party computation, the exchange of secrets, and key agreement. Therefore, it is of great significance to design efficient OT protocols.

1.1 Our Motivation

As far as we know, most existing OT protocols are built upon number theoretical problems mainly consist of the hardness of factoring and computing discrete logarithm problem. However, threatened by quantum computing [16], these protocols will be broken down directly in the presence of quantum computer. Therefore, it is essential to construct OT protocol based on post-quantum cryptography, such as lattice-based protocol and code-based protocol. Lattice-based cryptography has some attractive properties when compared with other post-quantum research fields, for instance, strong security guarantees from worst-case hardness and algorithmic simplicity. Among lattice-based hard problems, the learning with errors (LWE) problem [14] has been used as an amazingly versatile basic tool to design cryptographic schemes. Specifically, Peikert *et al.* [11] proposed an efficient and universally composable OT protocol which is extracted from a dual-mode cryptosystem and can be instantiated with the decisional Diffie-Hellman assumption, the quadratic residuosity assumption and the worst-case lattice assumption, respectively.

Zheng *et al.* [17] researched the framework for composable oblivious transfer and proposed a secure OT protocol from indistinguishability obfuscation (iO), with a dual-mode cryptosystem and an iO as main technical tools. Their work mainly has the following contributions. First, a k-out-of-n OT protocol was presented. Second, it explored the applications of iO. iO is a weaker notation of obfuscation that was first formally defined by [2]. They

suggested a definition of virtual black box obfuscation, and proved that this notion is impossible to realize. In order to avoid the impossibility result, they presented the notion of iO, which only requires that if two circuites compute the same functionality, then their obfuscation should be computational indistinguishable from each other. iO is both very useful and potentially achievable. Garg *et al.* [7] proposed the first candidate GGH13-based [6] iO for general circuits. Subsequently, many applications of iO were described in [15], such as public encryption, injective trapdoor function, deniable encryption, and so on.

The OT protocol of Zheng *et al.* has a main tool that is based on the hardness of the decisional Diffie-Hellman (DDH) problem. However, DDH assumption does not guarantee against quantum attack, and the selection of iO is the first candidate that have been attacked by [4]. For these reasons, we aim to remedy the insufficiency and try to design another new OT protocol that is security in quantum setting. Therefore, we take advantage of the LWE problem and a secure iO. Furthermore, if the security of OT protocol is proved only according to an ideal world simulator that is shown only for a cheating receiver, then they are not necessarily secure when integrated into a lager protocol. Thus, our protocol needs to satisfy the property of universally composable simultaneously.

1.2 Our Contribution

In this work, we combine LWE-based dual-mode cryptosystem [11] and a secure iO to design a new OT protocol. The key technique is an obfuscator of the dual-mode cryptosystem based on the hardness of LWE, versus based on DDH assumption in [17]. It is important that we choose GGH13-based obfuscator which against quantum attack when combined with the technique of [5] to prevent input partitioning. By utilizing these tools, we realize the oblivious transform functionality, and guarantee security of our OT protocol.

1.3 Organization

The rest of this paper is organized as follows. In Section 2, we introduce two useful definitions of LWE and iO. Then two corresponding building blocks are given in Section 3. In Section 4, we construct an LWE-based oblivious transfer protocol from iO, and the security proof of our OT protocol is presented in Section 5. Finally, conclusions are drawn in Section 6.

2 Preliminares

In this section, we introduce some notations and fundamental definitions.

2.1 Notation

We let $\mathbb{N}$ denote the set of natural numbers, for $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \ldots, n\}$. For an integer $q \geq 1$, $\mathbb{Z}_q$ denotes the quotient ring $\mathbb{Z}/q\mathbb{Z}$. Let " $\leftarrow$ " denote sampling an element from some distribution uniformly at random. We use bold lower-case letters to denote vectors in column form, and bold upper-case letters to denote matrices. Let $n \in \mathbb{N}$ denote the security parameter throughout this paper, and all other quantities are functions of n. We use standard notation o to classify the growth of functions, the function $\mathrm{negl}(n)$ denotes an unspecified function $f(n) = o(n^{-c})$ for some constant $c > 0$, calling $\mathrm{negl}(n)$ is negligible, and we say a probability is overwhelming if it is 1-$\mathrm{negl}(n)$. We use the definition of computational indistinguishability, denoted by $\overset{c}{\approx}$.

2.2 Learning with Errors

The LWE problem was proposed by Regev [14], the hardness of it can be reduced by a quantum algorithm to some standard problems on lattices in the worst case.

For an integer $q = q(n) \geq 2$ and some probability distribution χ over $\mathbb{Z}_q$, we define $A_{s,\chi}$ as the distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ of the tuples $(\mathbf{a}, c) = (\mathbf{a}, \mathbf{a}^T\mathbf{s} + e)$ where $\mathbf{s}, \mathbf{a} \leftarrow \mathbb{Z}_q^n$ is uniform and $e \leftarrow \chi$, and all operations are performed in $\mathbb{Z}_q$. There are two versions of the LWE problem, search-LWE and decision-LWE, respectively.

Definition 1 (Search-LWE and decision-LWE)**.** *For an integer $q = q(n)$ and a distribution χ on $\mathbb{Z}_q$, for any $\mathbf{s} \in \mathbb{Z}_q^n$, search-LWE finds $\mathbf{s}$ given any independent samples $(\mathbf{a}, c)$ from $A_{s,\chi}$. The goal of decision-LWE is to distinguish between an oracle that returns independent samples from $A_{s,\chi}$ for some uniform $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, and an oracle that returns independent samples from the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$.*

Regev showed that these two versions are polynomially equivalent for $q = poly(n)$. He proved that for certain choices of q and χ, the decision-LWE problem is as hard as solving the shortest independent vectors problem (SIVP) using a quantum algorithm.

Theorem 1. *Let $q = q(n)$ be a prime and let $\alpha = \alpha(n) \in (0, 1)$ such that $\alpha q > 2\sqrt{n}$. If there exists an efficient algorithm that solves the decision-LWE problem, then there exists an efficient quantum algorithm for the SIVP within $\widetilde{O}(n/\alpha)$ in the worst case.*

Due to the hardness of SIVP, we choose the decision-LWE problem as underlying hardness in this paper.

2.3 Indistinguishability Obfuscation

Program obfuscation aims to make computer programs "unintelligible" while preserving their functionality. The systematic study of program obfuscation was initiated by Barak *et al.* in 2001. In their work, they gave a potentially realizable notion of iO. iO requires that, given any two equivalent circuits of the same size, the obfuscation of these two circuits should be computationally indistinguishable. The specific definition is as follows.

Definition 2 (Indistinguishiability obfuscation)**.** *A PPT algorithm iO is said to be an indistinguishability obfuscator for a class of circuits $\mathbb{C}$, if it satisfies:*

Functionality: For any $C \in \mathbb{C}$ and security parameter n,

$$Pr[\forall x : iO(C, 1^n)(x) = C(x)] = 1.$$

Indistinguishability: For any PPT distinguisher D, there exists a negligible function $negl(\cdot)$, such that for any two circuits $C_0, C_1 \in \mathbb{C}$ that compute the same function are of the same size:

$$|Pr[D(iO(C_0, 1^n)) = 1] - Pr[D(iO(C_1, 1^n)) = 1]|$$

$$\leq negl(n).$$

Starting with the work of [7] constructing the first iO candidate for the polynominal-size circuit. Several iO candidates have appeared in literatures [1,3,9,10]. Unfortunately, many constructions are attacked by [10]. So we choose a secure iO [12] based on GGH13 multilinear map that haven't found classical attack and quantum attack.

3 Building Blocks

In order to construct an LWE-based oblivious transfer protocol from iO, we need two building blocks, including LWE-based dual-mode cryptosystem and a secure iO.

3.1 LWE-based Dual-mode Cryptosystem

The dual-mode cryptosystem is a simple and general framework, proposed by Peikert *et al.* [11]. Actually it is an encryption scheme that can operate in two modes, which are called *messy mode* and *decryption mode*. The trusted setup phase produces a common reference string (denoted by crs) and the corresponding trapdoor information according to one of two chosen modes. The crs may be uniformly random or be some specified distribution.

The dual-mode cryptosystem has four security properties:

1) It suffices decryption completeness with overwhelming probability over the randomness of the entire experiment;

2) Given crs, the first outputs of SetupMessy and SetupDec are computationally indistinguishable;

3) In Messy mode, for every pk, at least one of the derived public keys can statistically hide its encrypted message;

4) In decryption mode, the honest receiver's chosen bit σ is statistically hidden by its choice or the base key pk.

These security properties make the dual-mode cryptosystem able to derive a UC-secure OT protocol.

The instantiation of the dual-mode cryptosytem based on the hardness of LWE relies on existing techniques, including an LWE-based encryption and an efficient securely embedded a trapdoor algorithm. So, we first introduce an optimized version of the LWE-based encryption, then instancing the dual-mode cryptosystem, where the message space is $\mathbb{Z}_2 = \{0, 1\}$. Let the modulus $q = \text{poly}(n)$ be a prime, all operations are performed over $\mathbb{Z}_q$. For every message $M \in \mathbb{Z}_2$, the "center" of M is defined as $t(M) = M \cdot \lfloor q/2 \rfloor \in \mathbb{Z}_q$. Let χ denote an error distribution over $\mathbb{Z}_q$.

LWEKeyGen (1^n): Choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a secret key $\mathbf{s} \leftarrow \mathbb{Z}_q^{n \times 1}$ are both uniformly at random. To generate the public key, choose an error vector $\mathbf{x} \leftarrow \mathbb{Z}_q^{1 \times m}$ where each entry $x_i \in \chi$ is chosen independently for all $i \in [m]$. Then compute $\mathbf{p} = \mathbf{s}^T\mathbf{A} + \mathbf{x}$, the public key is $(\mathbf{A}, \mathbf{p})$.

LWEEnc $((pk = (\mathbf{A}, \mathbf{p}), M)$: To encrypt a message $M \in \mathbb{Z}_2$, choose a vector $\mathbf{e} \in \mathbb{Z}_2^m$ uniformly at random. The ciphertext is the pair $(\mathbf{u}, c) = (\mathbf{Ae}, \mathbf{pe} + M \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

LEWDec $((sk = \mathbf{s}, (\mathbf{u}, c)))$: Compute $d = c - \mathbf{s}^T\mathbf{u} \in \mathbb{Z}_q$, output 0 if d is closer to 0 than $\lfloor q/2 \rfloor$ modulo q, otherwise output 1.

We verify the completeness of the encryption scheme based on the LWE. The decryption algorithm needs to compute

$$d = c - \mathbf{s}^T\mathbf{u} = (\mathbf{s}^T\mathbf{A} + \mathbf{x})\mathbf{e} + M \cdot \lfloor q/2 \rfloor - \mathbf{s}^T\mathbf{Ae}$$

$$= \mathbf{xe} + M \cdot \lfloor q/2 \rfloor \in \mathbb{Z}_q.$$

If $\mathbf{xe} + M \cdot \lfloor q/2 \rfloor$ is closer to 0 than $\lfloor q/2 \rfloor$ modulo q, then output 0, otherwise output 1.

The encryption scheme based on the LWE is secure under chosen plaintext attack, unless SIVP and GapSVP are easy for quantum algorithms.

We now give the construction of the LWE-based dual-mode cryptosystem using LWE-based encryption. It consists six probabilistic algorithms, and the last two algorithms are only used in the security proof.

SetupMessy (1^n): Choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ uniformly at random, together with a trapdoor $t = \{\mathbf{S}, \mathbf{A}\}$ as in [8]. Choose a row vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1 \times m}$. For each $b \in \{0, 1\}$, choose an independent row vector $\tau_b \in \mathbb{Z}_q^{1 \times m}$ uniformly at random. Let $crs = (\mathbf{A}, \tau_1, \tau_2)$ and output (crs, t).

SetupDec (1^n): Choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a row vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1 \times n}$ are both at random. For each $b \in \{0, 1\}$, choose a secret $\mathbf{s}_b \leftarrow \mathbb{Z}_q^n$ and an error row vector $\mathbf{x}_b \leftarrow \chi$ are both uniformly at random. Let $\tau_b = s_b^T A + x_b - \mathbf{w}$, $crs = (\mathbf{A}, \tau_1, \tau_2)$, $t = (\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$ and output (crs, t).

KeyGen (σ): Choose a secret $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ uniformly at random and a row vector $\mathbf{x} \leftarrow \chi^{1\times m}$. Let $pk = \mathbf{s}^T\mathbf{A} + \mathbf{x} - \tau_\sigma$, $sk = \mathbf{s}$, and output (pk, sk).

Enc(pk, b, M)**:** Output $y \leftarrow \text{LWEEnc}(\mathbf{A}, pk + \tau_b, M)$, where y is the pair $(\mathbf{u}, c)$.

Dec(sk, y)**:** Output $M \leftarrow \text{LWEDec}(sk, (\mathbf{u}, c))$.

Findmessy(t, pk)**:** Parse t as $(\mathbf{S}, \mathbf{A})$, run ISMessy $(\mathbf{S}, \mathbf{A}, pk + \tau_b)$ for each $b \in \{0,1\}$, and output a b such that IsMessy can output messy on at least one branch correctly with overwhelming probability.

TrapKeyGen(t)**:** Parse t as $(\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$, and output $(pk, sk_0, sk_1) = (\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$.

According to [8], we know an efficient and UC-secure OT protocol based on LWE hardness can be directly derived when the LWE-based dual-mode cryptosystem built well. Although the LWE-based dual-mode cryptoystem is a relaxed version, it can still derive a UC-secure OT protocol based on the LWE hardness.

3.2 Secure Indistinguishability Obfuscation

In this section, we introduce a secure iO as another tool in our scheme. Indistinguishability obfuscation is a powerful notion, which holds for every pair functionally equivalent circuits C_0, C_1 that $iO(C_0)$ and $iO(C_1)$ are computationally indistinguishable. Almost all known candidate constructions of iO are based on multilinear-maps, which have been the subjects of various attacks. The first candidate branching program obfuscator can be attacked when the branching program has input partitioning. So we combine it with the prevent input partitioning technique to against cryptanalytic attacks.

In this section, we introduce the secure iO for all circuits. Firstly, we need to construct iO for NC^1 circuit. More specifically, an NC^1 circuit can be computed by branching programs. Let $f : \{0,1\}^n \to \{0,1\}$ be a function to be obfuscated. Fernando *et al.* [5] give a model which takes partitionable f as input and produces a function g with the same functionality, where g has no input partitions exist. In this way, GGH13 - based iO can defence the extension of annihilation attacks by [4]. Secondly, using iO for NC^1 circuit together with Fully Homomorphic Encryption (FHE) to achieve iO for all circuits. The process contains an obfuscation algorithm and an evaluation algorithm. To obfuscate a circuit C, we choose and publish two FHE keys PK_0 and PK_1. Obfuscate$(1^\lambda, C \in \mathbb{C}_\lambda)$, then output $\tau = (P, PK^1_{FHE}, PK^2_{FHE}, g_1, g_2)$, where $P = iO_{NC^1}(P1^{SK^1_{FHE}}, g_1, g_2)$, $g_1 = \text{Encrypt}_{FHE}(P1^{SK^1_{FHE}}, C)$ and $g_2 = \text{Encrypt}_{FHE}(P1^{SK^2_{FHE}}, C)$. We describe the two program classes in Figure 1 and Figure 2. The evaluate algorithm takes in the obfuscation output τ and program input M, denoted by Evaluate(τ, M). Compute the following procedure, where U_λ is a poly-sized universal circuit.

$P1$

Given input (M, e_1, e_2, ϕ), $(P1^{SK^1_{FHE}, g_1, g_2})$ proceeds as follows:
1. Check if ϕ is a valid low-depth proof for the NP-statement:

$$e_1 = \text{Eval}_{FHE}(PK^1_{FHE}, U_\lambda(\cdot, M), g_1),$$
$$e_2 = \text{Eval}_{FHE}(PK^2_{FHE}, U_\lambda(\cdot, M), g_2).$$

2. If the check fails output 0; otherwise, output

$$\text{Decrypt}_{FHE}(e_1, \text{SK}^1_{FHE}).$$

Figure 1: Program P1

$P2$

Given input (M, e_1, e_2, ϕ), $(P2^{SK^2_{FHE}, g_1, g_2})$ proceeds as follows:
1. Check if ϕ is a valid low-depth proof for the NP-statement:

$$e_1 = \text{Eval}_{FHE}(PK^1_{FHE}, U_\lambda(\cdot, M), g_1),$$
$$e_2 = \text{Eval}_{FHE}(PK^2_{FHE}, U_\lambda(\cdot, M), g_2).$$

2. If the check fails output 0; otherwise, output

$$\text{Decrypt}_{FHE}(e_2, \text{SK}^2_{FHE}).$$

Figure 2: Program P2

1) Compute

$$e_1 = \text{Eval}_{FHE}(PK^1_{FHE}, U_\lambda(\cdot, M), g_1),$$
$$e_2 = \text{Eval}_{FHE}(PK^2_{FHE}, U_\lambda(\cdot, M), g_2).$$

2) Compute a low depth proof ϕ that e_1 and e_2 were computed correctly.

3) Run $P(M, e_1, e_2, \phi)$ and output the result.

4 LWE-based Oblivious Transfer Protocol from Indistinguishability Obfuscation

4.1 Obfuscator for LWE-based Dual-mode Cryptosystem

The setup of a dual-mode cryptosystem has messy mode and decrytion mode, and they are computationally indistinguishable. Therefore, their obfuscating results are still

indistinguishable. We know SetupMessy and SetupDec have two choices respectively, denoted as four circuits C_n that describe in Figure 3, 4, 5, 6, where $n = 1, 2, 3, 4$. We obfuscate these circuits to substitute the two modes in LWE-based dual-mode cryptosystem. The key is to describe the four circuits C_n, that are based on LWE encryption scheme.

The circuit C_1 and C_2 output truly random vectors, and the circuit C_3 and C_4 output LWE instantiations. Through obfuscating these circuits C_1, C_2, C_3 and C_4, we obtain the result that their outputs are computationally indistinguishable. The specific setup of the LWE-based dual-mode cryptosystem are called two obfuscation branches as follows.

Messy-obf-branch (1^n): Choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ uniformly at random, together with a trapdoor $t = \{\mathbf{S}, \mathbf{A}\}$ as in [8]. Choose a row vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1\times m}$. For each $b \in \{0,1\}$, choose a secret $\mathbf{s}_b \leftarrow \mathbb{Z}_q^n$ and an error row vector $\mathbf{x}_b \leftarrow \chi$ are both uniformly at random. Let $\tau_1 = \text{Obfuscate}(1^\lambda, C_1)$, $\tau_2 = \text{Obfuscate}(1^\lambda, C_2)$, $crs = (\mathbf{A}, \tau_1, \tau_2)$ and output (crs, t).

Dec-obf-branch (1^n): Choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a row vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1\times n}$ are both uniformly at random. For each $b \in \{0,1\}$, choose a secret $\mathbf{s}_b \leftarrow \mathbb{Z}_q^n$ and an error row vector $\mathbf{x}_b \leftarrow \chi$ are both uniformly at random. Let $\tau_1 = \text{Obfuscate}(1^\lambda, C_3)$, $\tau_2 = \text{Obfuscate}(1^\lambda, C_4)$, $crs = (\mathbf{A}, \tau_1, \tau_2)$, $t = (\mathbf{w}, \mathbf{s}_0, \mathbf{s}_1)$ and output (crs, t).

Next, we invoke the evaluate algorithm in KeyGen process, where we let $V_0 = \text{Obfuscate}(\tau_1, \mathbf{A})$ and $V_1 = \text{Obfuscate}(\tau_2, \mathbf{A})$. Comparing to the above LWE-based dual-mode cryptosystem, the rest of the steps are identical except that $\mathbf{V}_\sigma$ is substituted for $\mathbf{v}_\sigma$.

4.2 Oblivious Transfer Protocol from Indistinguishability Obfuscation

OT_1^2 is a two-party protocol, involving a sender $\mathbf{S}$ inputs M_0, M_1 and a receiver $\mathbf{R}$ inputs a choice bit $\sigma \in \{0,1\}$. The result is that $\mathbf{R}$ learns M_σ and nothing about another message, while $\mathbf{S}$ learns nothing at all. Our OT protocol operate in the common reference string model, denoted by F_{crs}^D, where D denotes a PPT algorithm. F_{crs}^D runs with two parties and there is a trusted party which can produce crs for two parties before interacting. Once the obfuscator for LWE-based dual-mode cryptosystem is constructed well, our LWE-based OT protocol from iO denoted by $iO\text{dm}^{\text{branch}}$ can be derived directly, we describe the protocol in Table 1. It can realize the exact definition of the ideal OT functionality in the F_{crs}^D. $iO\text{dm}^{\text{branch}}$ operates in two branches, when D=Messsy-obf-branch, D runs in the Messsy-obf-branch; when D=Dec-obf-branch, D runs in the Dec-obf-branch.

Circuit C_1

Input: choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a row vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1\times m}$ are both uniformly at random. Choose a secret $\mathbf{s}_0 \leftarrow \mathbb{Z}_q^n$ and an error row vector $\mathbf{x}_0 \leftarrow \chi^{1\times m}$ are all uniformly at random.
Output: a row vector $\mathbf{v}_0 \leftarrow \mathbb{Z}_q^{1\times m}$ uniformly at random.

Figure 3: Circuit C_1

Circuit C_2

Input: choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a row vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1\times m}$ are both uniformly at random. Choose a secret $\mathbf{s}_1 \leftarrow \mathbb{Z}_q^n$ and an error row vector $\mathbf{x}_1 \leftarrow \chi^{1\times m}$ are both uniformly at random.
Output: a row vector $\mathbf{v}_1 \leftarrow \mathbb{Z}_q^{1\times m}$ uniformly at random.

Figure 4: Circuit C_2

Circuit C_3

Input: choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a row vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1\times m}$ are both uniformly at random. Choose a secret $\mathbf{s}_0 \leftarrow \mathbb{Z}_q^n$ and an error row vector $\mathbf{x}_0 \leftarrow \chi^{1\times m}$ are both uniformly at random.
Output: $\mathbf{v}_0 = \mathbf{s}_0^T\mathbf{A} + \mathbf{x}_0 - \mathbf{w}$.

Figure 5: Circuit C_3

Circuit C_4

Input: choose a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a row vector $\mathbf{w} \leftarrow \mathbb{Z}_q^{1\times m}$ are both uniformly at random. Choose a secret $\mathbf{s}_1 \leftarrow \mathbb{Z}_q^n$ and an error row vector $\mathbf{x}_1 \leftarrow \chi^{1\times m}$ are both uniformly at random.
Output: $\mathbf{v}_1 = \mathbf{s}_1^T\mathbf{A} + \mathbf{x}_1 - \mathbf{w}$.

Figure 6: Circuit C_4

Table 1: Protocol $i\mathsf{Odm}^{\mathsf{branch}}$ for oblivious transfer

Sender $(\mathsf{sid}, \mathsf{ssid}, M_0, M_1)$		**Receiver** $(\mathsf{sid}, \mathsf{ssid}, \sigma)$
Setup:		
	$(\mathsf{sid}, \mathbf{S}, \mathbf{R})$ ⇉ F_{crs}^{D} ⇇ $(\mathsf{sid}, \mathbf{S}, \mathbf{R})$ (sid, crs) (sid, crs)	
Multi-session OT:		
	$(\mathsf{sid}, \mathsf{ssid}, pk)$ ←	$(pk, sk) \leftarrow$ KeyGen-obf-branch(crs, σ)
$y_b \leftarrow \mathsf{Enc}(pk, b, M_b)$ for each $b \in \{0,1\}$	$(\mathsf{sid}, \mathsf{ssid}, y_0, y_1)$ →	outputs $(\mathsf{sid}, \mathsf{ssid}, \mathsf{Dec}(sk, y_\sigma))$

5 Security Proof

Obfuscator for LWE-based dual-mode cryptosystem includs two obf-branches, the trapdoor generation of keys in Dec-obf-branch, and the guaranteed existence and identification of messy branches in Messy-obf-branch. Its properties take the form of theorem as follows.

Theorem 2. *In the obfuscator for LWE-based encryption scheme construction, the Messy-obf-branch and Dec-obf-branch are indistinguishability, assuming LWE is hard.*

Proof. The output of Dec-obf-branch is of form $(\mathbf{A}, (\mathbf{s}_0^T\mathbf{A}+\mathbf{x}_0)-\mathbf{w}, (\mathbf{s}_1^T\mathbf{A}+\mathbf{x}_1)-\mathbf{w})$. Because of the hardness of LWE, we have $(\mathbf{A}, \mathbf{s}_1^T\mathbf{A}+\mathbf{x}_1) \overset{c}{\approx} (\mathbf{A}, \mathbf{w}_1)$, where $\mathbf{w}_1 \leftarrow Z_q^{1\times m}$ is uniformly random and independent. So we have $(\mathbf{A}, (\mathbf{s}_0^T\mathbf{A}+\mathbf{x}_0)-\mathbf{w}, (\mathbf{s}_1^T\mathbf{A}+\mathbf{x}_1)-\mathbf{w}) \overset{c}{\approx} (\mathbf{A}, (\mathbf{s}_0^T\mathbf{A}+\mathbf{x}_0)-\mathbf{w}, \mathbf{w}_1-\mathbf{w})$. The right side of the vector equation is totally uniform, because $\mathbf{w}$ and $\mathbf{w}_1$ are uniform and independent. By the output of Messy-obf-branch is entirely uniform, thus $(\mathbf{A}, (\mathbf{s}_0^T\mathbf{A}+\mathbf{x}_0)-\mathbf{w}, (\mathbf{s}_1^T\mathbf{A}+\mathbf{x}_1)-\mathbf{w}) \overset{c}{\approx} (\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1)$. □

Theorem 3. *In the obfuscator for LWE-based encryption scheme construction satisfying for every* $(crs, t) \leftarrow Dec-obf-branch(1^n)$, *TrapKenGen(t) outputs* (pk, sk_0, sk_1) *such that for every* $\sigma \in \{0,1\}$, $(pk, sk_\sigma) \approx KeyGen(\sigma)$, *assuming LWE is hard.*

Proof. The case $\sigma = 0$ and $\sigma = 1$ are symmetrically, so we consider only one of them. Given the case $\sigma = 0$, we will prove that

$$(Dec-obf-branch(1^n), KeyGen(0)) \overset{c}{\approx} (crs, (pk, sk_0)),$$

where $(crs, t) \leftarrow Dec-obf-branch(1^n)$ and $(pk, sk_0, sk_1) \leftarrow TrapKeyGen(t)$. We get the result using a sequence of hybrid games. □

By the outputs of these two branches are indistinguishable, we have the first hybrid game expands as

$$(\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1, \mathbf{s}^T\mathbf{A}+\mathbf{x}-\mathbf{v}_0, \mathbf{s}),$$

where $\mathbf{A}, \mathbf{v}_0, \mathbf{v}_1$ and $\mathbf{s}$ are uniform and $\mathbf{x} \leftarrow \chi^{1\times m}$.

Through defining $\mathbf{w} = \mathbf{s}^T\mathbf{A}+\mathbf{x}-\mathbf{v}_0$ and using $\mathbf{s}_0$ and $\mathbf{x}_0$ replace $\mathbf{s}$ and $\mathbf{x}$, the second game outputs

$$(\mathbf{A}, \mathbf{s}_0^T\mathbf{A}+\mathbf{x}_0-\mathbf{w}, \mathbf{v}_1, \mathbf{w}, \mathbf{s}_0),$$

where $\mathbf{w}$ is uniform. Because $\mathbf{v}_1$ is uniform and independent of the other variables, the third game outputs

$$(\mathbf{A}, \mathbf{s}_0^T\mathbf{A}+\mathbf{x}_0-\mathbf{w}, \mathbf{v}_1-\mathbf{w}, \mathbf{w}, \mathbf{s}_0).$$

The above three games are equivalent to each other.

The hardness of LWE implies that $(\mathbf{A}, \mathbf{v}_1)$ is distinguishable from $(\mathbf{A}, \mathbf{s}_1^T\mathbf{A}+\mathbf{x}_1)$, where $\mathbf{s}_1 \leftarrow Z_q^n$ and $\mathbf{x}_1 \leftarrow \chi^{1\times m}$. So the prior games are indistinguishable from the one that outputs

$$(\mathbf{A}, \mathbf{s}_0^T\mathbf{A}+\mathbf{x}_0-\mathbf{w}, \mathbf{s}_1^T\mathbf{A}+\mathbf{x}_1-\mathbf{w}, \mathbf{w}, \mathbf{s}_0).$$

This is the whole process, the final output is equivalent to $(crs, (pk, sk_0))$ by definition.

Theorem 4. *In the obfuscator for LWE-based encryption scheme construction using the parameters* $m \geq 2(n+1)logq$ *and* $t \geq \sqrt{qm} \cdot log^2 m$, *for* $(crs, t) \leftarrow$ *Messy-obf-branch*(1^n) *and every key pk, FindMessy*(t, pk) *outputs a messy branch with overwhelming probability.*

Proof. In [8], the facts are as follows. Let $m \geq 2(n+1)logq$ and $t \geq \sqrt{qm} \cdot log^2 m$, there is a negligible function $negl(m)$ such that with overwhelming probability over the choice of $\mathbf{A}, \mathbf{S}$, for all but an at most $(1/2\sqrt{q})^m$ fraction of vectors $\mathbf{p} \in Z_q^{1\times m}$, Ismessy$(\mathbf{S}, \mathbf{A}, \mathbf{p})$ outputs messy with overwhelming probability. Define $D \subseteq Z_q^{1\times m}$ to be the set of vectors $\mathbf{p}$, then we have

$$Pr[\mathbf{v} \notin D] \leq (1/2\sqrt{q})^m, \text{where } \mathbf{v} \in Z_q^{1\times m}.$$

□

For every $pk \in Z_q^{1\times m}$, there is a branch $pk + \mathbf{v}_b \in M$, where $b \in \{0,1\}$ and $\mathbf{v}_0, \mathbf{v}_1 \in Z_q^{1\times m}$ in the crs. For any fixed pk, we have

$$Pr[pk+\mathbf{v}_0 \notin M \text{ and } pk+\mathbf{v}_1 \notin D]$$

$$= (Pr[\mathbf{v} \notin D])^2 \leq (1/4q)^m.$$

For $(crs, t) \leftarrow$ Messy-obf-branch(1^n) and every key pk, FindMessy(t, pk) outputs a messy branch with overwhelming probability, because of both branches lie outside D is at most $(1/4)^m$=negl(n).

From the above, we draw a conclusion that the obfuscator for LWE-based encryption scheme is a slightly relaxed dual-mode cryptosystem. On the basis of it, we obtain an OT protocol. The protocol operates in either obf-branches, which are obfuscation of the dual-mode encryption branches. The OT protocol based on dual-mode securely realizes the functionality F_{OT}. Therefore, our LWE-based oblivious transfer protocol from indistinguishability obfuscation is secure.

6 Conclusions

We can see that most existing OT protocols are based on the hardness of number theoretical problems. In this paper, we propose a secure LWE-based oblivious transfer protocol from iO, and give the proof of security. In addition to iO, our protocol is based on LWE-based dual-mode encryption, which is a framework for efficient and composable oblivious transfer. Thus, our protocol can realize the oblivious transform functionality. Compared with the protocol of Zheng *et al.*, our protocol is secure in the quantum environment. At present, using punctured programs technique to carry out some applications of iO gradually become a central primitive for cryptography, so we would like to use this technique to build another secure, efficient, and succinct oblivious transfer protocol.

Acknowledgments

This study was supported by the National Key R&D Program of China under Grant No.2017YFB0802000, the National Natural Science Foundations of China under Grant Nos.61972457, 61672412, 61402015, U1736111, the National Cryptography Development Fund under grant No.MMJJ20170104, the MOE Layout Foundation of Humanities and Social Sciences under Grant 19YJA790007, and the Research Program of Baoji University of Arts and Sciences under Grant No.ZK2018093. I acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. Badrinarayanan, E. Miles, A. Sahai, and M. Zhandry, "Post-zeroizing obfuscation: New mathematical tools, and the case of evasive circuits," in *Advances in Cryptology*, pp. 764–791, 2016.

[2] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs," in *Advances in Cryptology*, pp. 1–18, 2001.

[3] N. Bitansky and V. Vaikuntanathan, "Indistinguishability obfuscation from functional encryption," in *Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS'15)*, pp. 171–190, 2015.

[4] Y. L. Chen, C. Gentry, and S. Halevi, "Cryptanalyses of candidate branching program obfuscators," in *Advances in Cryptology*, pp. 278–307, 2017.

[5] R. Fernando, P. M. R. Rasmussen, and A. Sahai, "Preventing CLT attacks on obfuscation with linear overhead," in *Advances in Cryptology*, pp. 242–271, 2017.

[6] S. Garg, C. Gentry, and S. Halevi, "Candidate multilinear maps from ideal lattices," in *Advances in Cryptology*, pp. 1–17, 2013.

[7] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, "Candidate indistinguishability obfuscation and functional encryption for all circuits," in *IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 40–49, 2013.

[8] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of 40th Annual ACM Symposium on Theory of Computing*, pp. 197–206, 2008.

[9] H. Lin, "Indistinguishability obfuscation from constant-degree graded encoding schemes," in *Advances in Cryptology*, pp. 28–57, 2016.

[10] E. Miles, A. Sahai, and M. Zhandry, "Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13," in *Advances in Cryptology*, pp. 629–658, 2016.

[11] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," *LNCS*, vol. 5444, no. 72, pp. 554–571, 2009.

[12] A. Pellet-Mary, "Quantum attacks against indistinguishablility obfuscators proved secure in the weak multilinear map model," in *Advances in Cryptology – CRYPTO 2018*, pp. 153–183, 2018.

[13] M. O. Rabin, "How to exchange secrets by oblivious transfer," *Technical Report*, 1981. (`https://eprint.iacr.org/2005/187.pdf`)

[14] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of 37th Annual ACM Symposium on Theory of Computing*, pp. 84–93, 2005.

[15] A. Sahai and B. Waters, "How to use indistinguishability obfuscation: Deniable encryption, and more," *Proceedings of the Annual ACM Symposium on Theory of Computing*, pp. 475–484, 2014.

[16] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[17] Y. Zheng, W. Mei, and F. Xiao, "Secure oblivious transfer protocol from indistinguishability obfuscation," *The Journal of China Universities of Posts and Telecommunications*, vol. 23, no. 3, pp. 1–10, 2016.

Biography

Shanshan Zhang received her B.S. degree in 2004 from Baoji University of Arts and Sciences, and received her M.S. degree in 2007 from Huaibei Normal University. Now she is a PhD student in Xidian University. Her main research interests include public key cryptography and indistinguishable obfuscation.

Verifiable Secret Sharing Based On Micali-Rabin's Random Vector Representations Technique

Haiou Yang and Youliang Tian
(Corresponding author: Youliang Tian)

School of Computer Science and Technology, Guizhou University
Guizhou Province, Guiyang, China
(Email: youliangtian@163.com)

(Received Mar. 24, 2019; Revised and Accepted Nov. 16, 2019; First Online Jan. 29, 2020)

Abstract

Verifiable secret sharing is the core basic protocol of many cryptographic systems, which is widely used in secure communication in network environment. By now, there are many researches on verifiable secret sharing for threshold structure, which lacks generality and flexibility compared with general access structure. However it is difficult to realize verifiable secret sharing scheme for general access structures. Existing generalized verifiable secret sharing schemes are few and have low efficiency. In this paper, we propose a new verifiable secret sharing scheme of general access structure. We use knowledge commitment scheme based on bilinear pairing to ensure the security and concealment of public information, and adopt the Micali-Rabin's random vector representations technique to improve the the efficiency of verification process. Our security and performance analysis shows that the new scheme is more efficient and practical compared to existing similar schemes.

Keywords: Bilinear Pairing; General Access Structure; Micali-Rabin's Random Vector Representations Technique; Verifiable Secret Sharing

1 Introduction

Secret sharing is the basic protocol for constructing cryptographic schemes such as secure multiparty computation and digital signature, which is mainly used for the distribution, preservation and reconstruction of secret and key (or other secret information), to prevent loss, damage or been tampered of information. The fundamental idea of secret sharing is that secret is divided into multiple parts by one dealer and shared among different participants, and some subsets of participants can be used to reconstruct the secret, while others cannot reconstruct it and get no information about secret. Shamir [10] and Blakley [2] first proposed the (t,n) threshold secret sharing scheme based on Lagrange interpolation polynomial and mapping geometry theory respectively. Asmuth [1] proposed the threshold secret sharing scheme based on Chinese Remainder Theorem (CRT). Halper and Teague [6] combined game theory with secret sharing and proposed the concept of rational secret sharing for the first time, that is, all participants are rational rather than honest or malicious. TIAN [12] analyzed the distribution mechanism and reconstruction mechanism of secret sharing under the framework of game theory, and studied the problem of one secret sharing based on Bayesian game, which solved the cooperation of this kind of rational secret sharing system. But none of these schemes can properly detect and prevent the malicious behavior of dealers and participants. To address possible dishonesty among participants, Chor *et al.* [3] proposed the concept of verifiable secret sharing (VSS) based on large integer factor decomposition problem for the first time. Stadler [11] improved the Chor's scheme, and proposed the Publicly Verifiable Secret Sharing schemes (PVSS) based on discrete logarithm. TIAN [13] constructed a non-interactive public verifiable secret sharing using bilinear pairs on elliptic curves, and its information rate reached 2/3. Jhanwar [4] proposed a PVSS scheme and provided a formal proof for the IND-secrecy of his scheme, based on the (t, n)-multi-sequence of exponents Diffie-Hellman assumption. After that, a lot of achievements have been made in the research on secret sharing schemes. However, almost all of the above schemes are designed for threshold structure, the threshold secret sharing is only a special case of generalized secret sharing.

Threshold structure lack flexibility and are not applicable in some specific scenario compared with general access structure. For instance, the dealer share secret among participants U_1, U_2, U_3, U_4 and specify the two subsets of participants $\{U_1, U_2\}, \{U_2, U_3, U_4\}$ can reconstruct the secret. So (t,n) threshold secret sharing scheme is no longer applicable under this circumstance. In order to study the

secret sharing of general access structures with wider applicability, Ito *et al.* [8] first proposed a secret sharing scheme based on general access structure, that is, the cooperation of participants with any authorized subset can reconstruct the secret. Harn *et al.* [7] applied integer programming to the generalized secret sharing scheme. They stipulate authorized subsets and non-authorized subsets to try to find a reasonable share allocation scheme so as to achieve the general access structure with the traditional (t,n) threshold scheme, but the scheme was expensive and has low efficiency in calculation. In the existing secret sharing scheme of general access structure, either the correctness of secret shares cannot be verified, or the computational overhead is increased to achieve the verifiability of secret shares.

Therefore, we propose an efficient generalized verifiable secret sharing scheme based on Micali-Rabin's random vector representations technique. In view of the complexity of verification process and the low probability of verifying the correctness of computing result, we adopt Micali-Rabins random vector representation technique, based on Zero-Knowledge Proof(ZKPs), proposed by Micali and Rabin [9]. Zero Knowledge Proofs, proposed by Golddwasser [5], are one of the most remarkable innovations in information security, which refers to the ability of a prover to convince a verifier that a statement is true without providing any useful information to the verifier, has been widely used in the field of information security. Rabin [9] developed a novel secure and highly efficient way for verifying correctness of the output of a transaction while keeping input values secret, based on the ZKPs. Xin [15] proposed a new fair and rational delegation computation. Aiming at the complexity of the verification problem, they adopted the Micali-Rabin's random vector representation technique. Consequently, ZKPs is used to prove the correctness of the computing results, which provides a new direction for our research.

In this paper, a new generalized verifiable secret sharing (GVSS) scheme is proposed. The contributions of our proposed GVSS are as follows: First, we proposed a GVSS scheme based on the difficulty of Diffie-Hellman problem of bilinear pairings. In this scheme, secret shares are chosen by the participants themselves, effectively avoiding deception by the dealer. Also, compared with threshold schemes, this scheme can specify any authorized subsets to reconstruct secret, which greatly increases the flexibility of secret sharing and expands the application scenarios of the schemes. Second, on account of the comlexity of verification phase, we adopt Micali-Rabin's random vector representation technique, that is, the secret shares are represented by knowledge commitment scheme for bilinear pairing. When needs to verify the correctness of secret shares, it only needs to execute an efficiently process according to the public information on the bulletin board.

The rest of this paper is organized as follows. In Section 2 we give the relevant backgrounds. Our proposed GVSS is described in Section 3. In Section 4, we give security analysis of our proposed GVSS and the comparison of performance between our proposed GVSS and the VSS proposed by ZHANG [16] and Tsu-Yang [14]. Conclusion is given in Section 5.

2 Preliminaries

2.1 Bilinear Pairing

Let G_1 and G_2 be additive cyclic groups and multiplicative cyclic groups of order q, where q is a big prime number. Assuming that discrete logarithm problems on group G_1 and G_2 are difficult. A map: $e : G_1 \times G_1 \to G_2$ with the following properties is called a bilinear pairing:

1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$ and $a, b \in Z_q^*$.

2) Non-degenerate: There exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$.

3) Computable: For all $P, Q \in G_1$, there exists an efficient algorithm to compute $e(P, Q) = 1$.

2.2 Knowledge Commitment Scheme Based on Bilinear Pairing

Let $P, Q \in G_1$ be two generators of group G_1. Nobody knows the the discrete log of P, Q (anybody does not know the $n \in Z_q^*$ such that $Q = nP$). When making commitment to $s \in Z_q^*$, we just have to compute the commitment $COM(s) = e(P, Q)^s$. When revealing the commitment, only s needs to be disclosed, and the verifier can verify whether the commitments revealed by dealer are correct according to $COM(s) = e(P, Q)^s$.

2.3 Micali-Rabin's Random Vector Representations

We adopt the knowledge commitment scheme based on bilinear pairing by Tian [**?**], the equality is proved by zero knowledge proofs. The properties of the bilinear pairing satisfy the above assumption, and F_q is a finite field and q is a large prime number of 128bits.

Definition 1. *A random vector representation of x is a vector $X = (u, v)$, where $u, v \in Z_q^*$, u was randomly chosen, and $v = (x - u) mod q$. The value of the vector X is $val(X) = (u + v) mod q$.*

Definition 2. *Commitment to vector $X = (u, v)$ is $COM(X) = (COM(u), COM(v))$, where $COM(u) = e(P, Q)^u$, $COM(v) = e(P, Q)^v$.*

Definition 3. *A list of commitments $COM(X^{(j)}), 1 \leq j \leq m$ are called value consistent if $val(X^{(j)}) = val(X^{(j+1)})$ for any $1 \leq j \leq m$.*

When needs to prove $COM(X), COM(Y)$ value consistent, where $X = (u_1, v_1), Y = (u_2, v_2)$, we will prove $val(X) = val(Y)$. Note that $val(X) = val(Y)$ if and only

if there exists $w \in Z_q^*$ such that $X = Y + (w, -w)$(If such an w does not exist, the value is inconsistent). The prover randomly chooses $c \leftarrow \{1, 2\}$. Assume that $c = 1$, the prover reveals to verifier $u_1, u_2, -w$. The verifier computes $COM(u_1), COM(u_2)$, and compares to the posted first coordinates of $COM(X), COM(Y)$. The verifier next checks that u_1=$u_2 - w$ is true. Vice versa, assume that $c = 2$, the prover reveals to verifier v_1, v_2, w. The verifier computes $COM(v_1), COM(v_2)$ and compares to the posted second coordinates of $COM(X), COM(Y)$. The verifier next checks that $v_1 = v_2 + w$ is true. Apparently, the prover accepts an false formula with a probability of $\frac{1}{2}$.

Lemma 1. ***If more than*** k ***commitments are false, then the probability that the verifier accepts is*** $(\frac{1}{2})^k$.

Proof. The probability that any $val(X^{(j)}) \neq val(X^{(j+1)})$ is not found to be wrong is at most $\frac{1}{2}$. So the probability that at least k formulas are not found wrong is $(\frac{1}{2})^k$ with a randomly chosen value $c \leftarrow \{1, 2\}$. □

3 Scheme

This scheme assumes that the Dealer D needs to share the secret s between n participants. Also, the scheme assumes the existence of a secure bulletin board(SBB), which is used to publish data and cannot be deleted or modified as soon as it is published. At the same time, the published data is visible to dealer and all participants. The scheme includes Distribution Phase, Verification Phase and Secret Reconstruction.

Initialization: Assume that F_q is a finite field and q is a large prime number, G_1 and G_2 are respectively additive cyclic groups and multiplicative cyclic groups of order q, $P, Q \in G_1$ are two generators of group G_1. The properties of the bilinear pairing satisfy the above assumption, and there are efficient algorithms for mapping $e : G_1 \times G_1 \rightarrow G_2$ on groups G_1 and G_2. $H : G_2 \rightarrow Z_q^*$ is the anti-collision hash function.

The secret distributor specifies the authorized subset. Assume that the participants set is $P = \{P_1, P_2, \cdots, P_n\}$, $\Gamma_0 = \{\delta_1, \delta_2, \cdots, \delta_n\}$ is the minimum access structure, $\delta_j = \{P_{1j}, P_{2j}, \cdots, P_{|\delta_j|j}\}$ is the authorized subset, which $|\delta_j|$ is the number of members in δ_j. The secret distributor is SD, the secret reconstructor is SR, the shared secret is s.

3.1 Distribution Phase

Step 1. Each participant P_{ij} randomly chooses $s_{ij} \in Z_q^*$, and computes $R_{ij} = e(P, Q)^{s_{ij}}$. And keeps s_{ij} secretly, delivers R_{ij} to SD.

Step 2. SD randomly selects s_0, and computes $R_0 = e(P, Q)^{s_0}$. Then, chooses $a \in Z_q^*$ randomly and construct a $1st$ degree polynomial $f(x) = (s + ax) mod q$. Simultaneously, chooses t different random numbers $d_1, d_2, \cdots, d_t$ to represent these t authorized subsets in Γ_0 respectively. In succession, SD computes $f(1)$, and for each authorized subset $\delta_j = \{P_{1j}, P_{2j}, \cdots, P_{|\delta_j|j}\}$ in Γ_0 computes $H_j = f(d_j) \oplus H(R_{1j}{}^{s_0}) \oplus H(R_{2j}{}^{s_0}) \oplus \cdots \oplus H(R_{|\delta_j|j}{}^{s_0})$. Finally, publish $R_0, f(1), H_1, H_2, \cdots, H_t, d_1, d_2, \cdots, d_t$ on the SBB.

3.2 Verification Phase

All participants of any authorized subset δ_j can cooperate to reconstruct the secret s. Assume that the participants of $\delta_j = \{P_{1j}, P_{2j}, \cdots, P_{|\delta_j|j}\}$ reconstruct the secret s.

Step 3. Each participant P_{ij} computes $R_{ij'} = R_0^{s_{ij}}$ based on the public information R_0 on the SBB. And each participant P_{ij} posts on the SBB $3k$ rows of $R_{ij}{}'$: $COM(R_{1j'}^{(h)})$, $\cdots$, $COM(R_{|\delta_j|j'}^{(h)})$, $1 \leq h \leq 3k$. The $3k$ rows of SBB use the Micali-Rabin's random vector representations technique $COM(R_{1j'}^{(h)}) = (COM(u_{ij'}^{(h)}),\ COM(v_{ij'}^{(h)}))$, where $R_{1j'}^{(h)} = (u_{ij'}^{(h)}, v_{ij'}^{(h)})$, $val(R_{1j'}^{(h)}) = (u_{ij'}^{(h)} + v_{ij'}^{(h)}) \bmod q$, $1 \leq h \leq 3k$.

Step 4. To begin with, P_{ij} randomly chooses half of the commitments from the $3k$ rows of $R_{ij'}$, P_{ij} secretly reveals $R_{ij'}$ and commitment values $R_{1j'}^{(h)} = (u_{ij'}^{(h)}, v_{ij'}^{(h)})$ to SR.

Next, determines the value of $c \leftarrow \{1, 2\}$ by flipping a coin, opening a part of the remaining commitments value. Assume that $c = 1$, P_{ij} secretly reveals the commitments $COM(u_{ij'}^{(h)}), COM(u_{ij'}^{(h+1)})$ and $-w$ to SR, where $w = (u_{ij'}^{(h+1)} - u_{ij'}^{(h)}) \bmod q$. Assume that $c = 2$, P_{ij} secretly reveals the commitments $COM(v_{ij'}^{(h)}), com(v_{ij'}^{(h+1)})$ and w to SR, where $w = (v_{ij'}^{(h+1)} - v_{ij'}^{(h)}) \bmod q$.

Step 5. At first, SR privately received commitment values $R_{1j'}^{(h)} = (u_{ij'}^{(h)}, v_{ij'}^{(h)})$ and $R_{ij'}$ sent by P_{ij}. SR first verify that the equation $val(R_{ij'}) = (u_{ij'}^{(h)} + v_{ij'}^{(h)}) \bmod q$ is correct.

Next, SR performs a value consistent check on the remaining commitment values. Assume that received $c = 1$, SR opens the commitment value $COM(u_{ij'}^{(h)}), COM(u_{ij'}^{(h+1)})$ and $-w$, then verifies that the equation $COM(u_{ij'}^{(h)}) = (COM(u_{ij'}^{(h+1)}) + (-w)) \bmod q$ is correct. If received $c = 2$, SR opens the commitment value $COM(v_{ij'}^{(h)}), COM(v_{ij'}^{(h+1)})$ and w, then verifies that the equation $v_{ij'}^{(h)} = (u_{ij'}^{(h+1)} + w) \bmod q$ is correct. Apparently, only opened half of the commitment values at one time, from lemma 1, it can be seen that the probability of the participants accepting an false equation is $\frac{1}{2}$, if more than k commitments are false, then the probability that the participants accept is $(\frac{1}{2})^k$.

3.3 Secret Reconstruction

Step 6. And then SR received the verified $R_{ij'}$. With these values, SR can compute $H_{j'} = H_j \oplus H(R_{1j'}) \oplus H(R_{2j'}) \oplus \cdots \oplus H(R_{|\delta_j|j'})$. With the two coordinate points $(1, f(1)), (d_j, H_{j'})$, SR can reconstruct $f(x) = xf(1) - xH_{j'} - d_j f(1) + H_{j'}(1 - d_j)^{-1}$. At last, the shared secret can be recovered by computing $s = f(0) \bmod q$.

4 Scheme Analysis

4.1 Security Analysis

Theorem 1. ***If the $R_{ij'}$ received by the secret recuperator SR are verified by the Micali-Rabin's random vector representations technique, then $R_{ij'}$ is the correct and has not been modified. Then this new scheme is verifiable.***

Proof. In the verification phase, by adopting the Micali-Rabin random vector representations technique, each participant P_{ij} committed $3K$ rows: $COM(R_{1j'}^{(h)}), \cdots, COM(R_{|\delta_j|j'}^{(h)}), 1 \le h \le 3k$ to the $R_{ij'}$ on the SBB. In the verification phase, SR verify half of the commitments $u_{ij'}^{(h)}, v_{ij'}^{(h)}$ and $R_{ij'}$ sent by P_{ij}(That is to verify that $val(R_{ij'}^{(h)}) = (u_{ij'}^{(h)} + v_{ij'}^{(h)})$modq is equal). If verification fails, SR reject $R_{ij'}$ sent by the dealer. If it's verified, SR performed a value consistent check on the remaining commitment values, that is verified $COM(u_{ij'}^{(h)}) = (COM(u_{ij'}^{(h+1)}) + (-w)) \bmod q$ or $v_{ij'}^{(h)} = (u_{ij'}^{(h+1)} + w) \bmod q$. According to Lemma 1, if more than k commitment values are wrong, the probability of SR accepting the wrong results is $(\frac{1}{2})^k$. To sum up, the $R_{ij'}$ verified by Micali-Rabin's random vector representations technique is the correct and has not been modified, this scheme is verifiable. □

Theorem 2. ***The knowledge commitment scheme based on bilinear pairing meets the requirements of complete hiding and computational binding.***

Proof. Assume that there exists $s' \in Z_q^*$ and $s' \neq s$ such that $COM(s') = COM(s)$(that is, the dealer can open the commitment in two ways). Assume that $s = s' + t, 0 < t < q$, that is $e(P,Q)^{s'} = e(P,Q)^s$. Because $P, Q \in G_1$ are two generators of group G_1, and q is the big prime order on group G_1, so $qP = 0, qQ = 0$(0 is the point at infinity of the group G_1). We get $e(s'P, Q) = e(sP, Q)$ from $e(P,Q)^{s'} = e(P,Q)^s = e(sP,Q) = e(s'P,Q)$, so we have $sP = s'P$. So there exists $sP - s'P = tP = 0$ with $s = s' + t, 0 < t < q$. But, $0 < t < q$, this contradicts P with order q, so it has to be $s = s'$, that is, the dealer only can open the commitment in one way. So the scheme meets the requirement of computational binding. □

Also $COM(s) = e(sP, Q) = e(P, sQ)$, it is not computationally feasible for an attacker to try to get the specifics of the commitment, since the calculation of Diffie-Hellman problem (CDHP) of bilinear pairings are hard to work out. In conclusion, the knowledge commitment scheme based on bilinear pairing meets the requirements of complete hiding and computational binding.

Theorem 3. ***It is assumed that the calculation of Diffie-Hellman problem (CDHP) of bilinear pairings are difficult to be solved, then the proposed scheme is of security.***

Proof. In the verification phase, the attackers try to get the s_0 and s_{ij} from the commitments R_0 and the commitments of $3k$ rows $COM(R_{1j'}^{(h)}), \cdots, COM(R_{|\delta_j|j'}^{(h)}), 1 \le h \le 3k$ posted on the SBB, they have to solve $R_0 = e(P,Q)^{s_0}$, $COM(u_{ij'}^{(h)}) = e(P,Q)^{u_{ij'}^{(h)}}$ and $COM(v_{ij'}^{(h)}) = e(P,Q)^{v_{ij'}^{(h)}}$. However, the discrete logarithm problem(DLP) on the elliptic curve and the calculation of Diffie-Hellman problem (CDHP) of bilinear pairings are difficult to be solved. So it's not computationally feasible to get the specifics of the commitments. □

At the same time, the participant P_{ij} tries to distribute false $R_{ij'}$ to SR in the verification phase, that is $COM(R_{ij'}{}') = COM(R_{ij'})$, where $R_{ij'}{}' \neq R_{ij'}, R_{ij'}R_{ij'}{}' \in Z_q^*$. However, according to theorem 2, the knowledge commitment scheme based on bilinear pairing meets the requirement of computational binding, the dealer only can open the commitment in one way. So P_{ij} cannot send a false $R_{ij'}$ to SR. Also, the secret shares s_{ij} of each participant P_{ij} in this scheme are chosen by the participants themselves, avoiding the distributor's deception.

4.2 Performance Analysis

This section briefly analyzes the performance of the proposed scheme by comparing it with the existing scheme. T_e denotes the time of executing a bilinear pairing, T_m denotes the time of executing a scalar of multiplication in G_1, T_exp denotes the time of executing an exponentiation in G_2, T_p denotes the time of computing the polynomial value. The time of executing a modular addition operation in Z_q^* and one-way hash function are negligible compared with T_e and T_m. Therefore, we just consider those time-consuming operations T_e, T_m, T_exp, T_p, other computational overhead is ignored.the computational efficiency as shown in Table 1.

The above mentioned, $|\delta_j|$ denotes the number of members in authorized subset, hence $|\delta_j| \ll n$. Therefore, performance analysis shows that in our scheme, with the adoption of Micali-Rabin's random vector representations technique, the verification process is much less computationally intensive. Compared with the above two schemes, the computational costs in the distribution phase has also been significantly improved.

Table 1: Comparison of computational costs

Schemes	Distribution Phase	Verification Phase	Reconstruction Phase
TIAN's scheme[8]	$(3n+t)G_1$	$tT_{\exp}+G_1+(t+1)T_e$	$3tT_m+2tT_e$
ZHANG's scheme[14]	$(n+1)T_m+2tT_{\exp}$	$tT_e+n(t+1)T_{\exp}$	tT_m
Tsu-Yang's scheme[15]	$nT_e+(4n+t)T_m$ $+nT_{\exp}+nT_p$	$(n+3)T_e+n(t+1)T_m$ $+nT_{\exp}+ntT_p$	tT_m
TIAN's scheme[8]	$(n+1)T_e+nT_{\exp}$	$6k\lvert\delta_j\rvert T_e+\lvert\delta_j\rvert T_{\exp}$	T_m

4.3 Simulation Analysis

For the generalized verifiable secret sharing based on Micali-Rabin's random vector representation technique proposed in this paper, simulation analysis was carried out in combination with the actual scenario. All data were the average of the experimental results for 10 times. The execution performance of the secret distribution process is shown in Figure 1. It can be seen that the execution time is linear with the change of the number of people. Because as the number of people increases, the number of operations of bilinear pairing and exponentiation increases. As can be seen from Figure 2, as the sub-secret is verified by Micali-Rabin's random vector representation technique in the verification phase, the calculation time of the secret verification process is less affected by the number of participants, and the calculation time of the secret reconstruction process does not change much with the number of people. In general, the scheme has good application value in practical application scenarios.

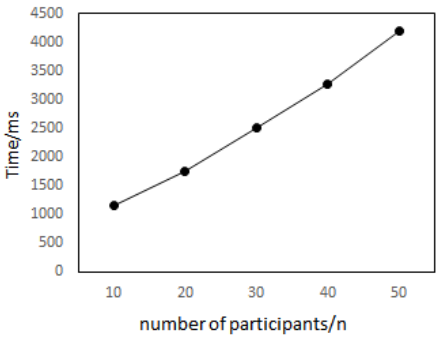


Figure 1: The curve of secret distribution calculations as the number of people changes

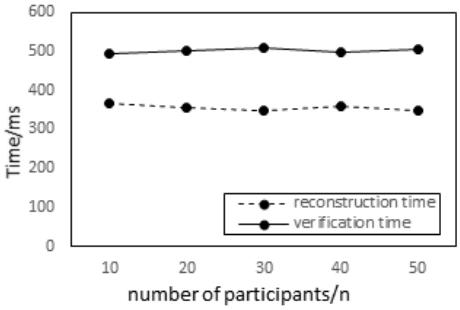


Figure 2: The curve of calculation costs as the number of people changes in the verification phase, and the curve of secret reconstruction calculations as the number of people changes

5 Conclusions

The (t,n) threshold secret sharing scheme has certain limitations in practical application, so it is of great application value to study the secret sharing of general access structure. This paper proposes a verifiable secret sharing scheme based on general access structure. Firstly, our scheme adopts the knowledge commitment scheme based on the bilinear pairing to guarantee the concealment and security of public information. Secondly, our scheme adopts Micali-Rabin's random vector representations technique, which greatly improves the efficiency of the verification phase. Finally, by analyzing the security and performance of the scheme, the scheme satisfies the feature of verifiable secret sharing and is more efficient than the existing schemes. The next work is to design a efficient secret sharing scheme with general access structure that can be publicly verified and applied to suitable application scenarios.

Acknowledgments

This work is supported by Key Projects of the Union Fund of the National Natural Science Foundation of China under Grant No. U1836205; The National Natural Science Foundation of China under Grant No. 61772008; The Science and Technology Top-notch Talent Support Project in Guizhou Province Department of Education under Grant No.Qian Education Combined KY word [2016]060; The Guizhou Province Science and Technology Major Special Plan No. 30183001; The Guizhou Provincial Science and Technology Plan Project under Grant No. [2017]5788; The Ministry of Education-China Mobile Research Fund Project under Grant No. MCM20170401; The Guizhou University Fostering Project No. [2017]5788; Research on Block Data Fusion Analysis Theory and Security Management Model of Data Sharing Application (No. U1836205); Research on Key Technologies of Blockchain for Big Data Applications (Grant No. [2019]1098).

References

[1] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 208-210, 1983.

[2] G. R. Blakley, "Safeguarding cryptographic keys," *IEEE Computer Society Digital Library*, vol. 22, no. 11, pp. 612-613, 1979.

[3] B. Chor and S. Goldwasser and S. Micali and B. Awerbuch, " Verifiable secret sharing and achieving simultaneity in the presence of faults," in *Foundations of Computer Science*, pp. 383-395, 1985.

[4] Y. Gan and L. Wang and P. Pan and Y. Yang, "Publicly verifiable secret sharing scheme with provable security against chosen secret attacks," *International Journal of Distributed Sensor Networks*, pp. 1-9, 2013.

[5] S. Goldwasser and S. Micali and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 186-208, 1989.

[6] J. Halpern and V. Teague, "Rational secret sharing and multiparty computation: Extended abstract," *The 36th Acm Symposium on Theory of Computing*, pp. 623-632, 2004.

[7] L. Harn and C. Hsu and M. Zhang and T. He and M. Zhang, "Realizing secret sharing with general access structure," *Information Sciences*, vol. 367-368, pp. 209-220, 2016.

[8] M. Ito and A. Saito and T. Nishizeki, "Secret sharing scheme realizing general access structure," *Electronics and Communications in Japan Part Iii-fundamental Electronic Science*, vol. 72, no. 9, pp. 56-64, 1989.

[9] M. O. Rabin and Y. Mansour and S. Muthukrishnan and M. Yung, "Strictly-black-box zero-knowledge and efficient validation of financial transactions," in *International Colloquium Conference on Automata*, pp. 738-749, 2012.

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

[11] M. Stadler, "Publicly verifiable secret sharing," in *Theory and Application of Cryptographic Techniques*, pp. 190-199, 1996.

[12] Y. L. Tian and J. Katz J. F. Ma and C. G. Peng, "One-time rational secret sharing scheme based on Bayesiangame," *Wuhan University Journal of Nature Science*, vol. 16, pp. 430-434, 2011.

[13] Y. Tian and C. Peng, "publicly verifiable secret sharing schemes using bilinear pairings," *International Journal Network Security*, vol. 14, no. 3, pp. 142-148, 2012.

[14] T. Wu and Y. Tseng, "A pairing-based publicly verifiable secret sharing scheme," *Journal of Systems Science and Complexity*, vol. 24, no. 1, pp. 186-194, 2011.

[15] Y. Xin and M. O. Rabin, "Fair and rational delegation computation protocol," *Journal of Software*, vol. 29, no. 7, pp. 1953-1962, 2018.

[16] F. Zhang, "Efficient and information-theoretical secure verifiable secret sharing over bilinear groups," *Chinese Journal of Electronics*, vol. 23, no. 1, pp. 13-17, 2014.

Biography

Haiou Yang biography. He received the B.Sc. degree in Information Management and System from Dalian Commuication University in 2017. He is now a postgraduate student at Guizhou University. His research interests include Security, Cloud computing and Cryptographic protocols.

Youliang Tian biography. He received the B.Sc. degree in Mathematics and Applied Mathematics in 2004 and the M.Sc. degree in Applied Mathematics from Guizhou University in 2009. He received the Ph.D. degree in cryptography from Xidian University in 2012. In the years 2012 to 2015, he was a Postdoctoral Associate at the State Key Laboratory for Chinese Academy of Sciences. He is currently a professor and Ph.D. supervisor at College Of Computer Science and Technology, GuiZhou University. His research interests include algorithm game theory, cryptography and security protocol.

A Privacy-Preserving Data Sharing System with Decentralized Attribute-based Encryption Scheme

Li Kang and Leyou Zhang
(Corresponding author: Li Kang)

School of Mathematics and Statistics, Xidian University
Xi'an, Shaanxi 710071, China
(Email: li_kkang@126.com)

(Received Mar. 26, 2019; Revised and Accepted Nov. 16, 2019; First Online Jan. 29, 2020)

Abstract

The huge storage space and powerful computing power make cloud servers be the best place for people to store data. However, convenience comes with the risk of data leakage as the cloud servers are not completely reliable. To ensure data confidentiality and user privacy, decentralized attribute-based encryption (ABE) can provide a solution. Nevertheless, most of existing works cannot provide a complete method since there are some vulnerabilities in users' collusion resilience and privacy protection. In this paper, a decentralized ciphertext-policy (CP) ABE scheme is proposed for the secure sharing of data. In the proposed scheme, without requiring any cooperation and knowing users' global identifiers (GID), authorities can issue private keys for users independently through a privacy-preserving key generation protocol. Additionally, this scheme supports policy anonymity. The security of the proposed scheme is reduced to the decisional bilinear Diffie-Hellman (DBDH) assumption. Theoretical analysis and performance evaluations illustrate the efficiency of the scheme.

Keywords: Cloud Storage; Data Sharing; Decentralized CP-ABE; Privacy-preserving

1 Introduction

1.1 Background

The emergence of the big data era makes cloud storage technology become the one of the hottest technologies. The cloud becomes the best choice for data users to store data as its storage space is much larger than the local one. The so-called cloud storage is a new technology that puts data on the cloud for human access, by which users can easily access data at any time and anywhere through any connected device. However, it also brings some challenges, such as data confidentiality.

In many cases, the data owners are reluctant to expose their data stored in the cloud server to all users as the data may be sensitive, such as the personal health record (PHR). PHR is a new summarized electronic record of an individual's medical data and information, which are often exchanged through cloud servers. However, the record may contain sensitive information about consumers, such as allergies, diseases, etc. Placing the raw PHR data directly on an unreliable third-party server will threaten the owner's privacy. Thus, to protect the data confidentiality, it is an effective way for the owner to execute encryption operation before uploading it to the cloud server.

In 2005, Sahai and Waters [30] proposed the concept of attribute-based encryption (ABE) to support fine-grained access control, in which only when the attributes of ciphertext match the decryption keys can the user successfully recover the received ciphertext. Since then, many studies [4,22,25] have been proposed. The multi-authority ABE (MA-ABE) [5] was introduced to reduce the burden on central authority (CA) and to divide its powers. In 2011, the decentralized ABE scheme [18] as a new MA-ABE scheme was proposed by Lewko and Waters. Neither CA nor authority cooperation exists in this construction, thus it is more in accord with the actual requirements for the independence of authorities. The basic properties [17,21] that all ABE schemes should satisfy are: fine-grained access control, scalability, data confidentiality, and collusion resistance.

In addition, the privacy protection of users cannot be ignored. Although the rapid development of cloud computing technology enables users to deal with a large amount of digital information, the privacy of personal data is also facing unprecedented challenges. After witnessing some information leakage incidents, people gradually realize the importance of privacy protection. They want to keep their data private while gaining legal access. Therefore, there is an urgent need for an encryption

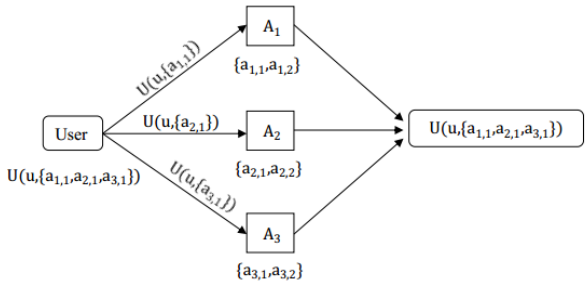


Figure 1: Colluding authorities gather information about the user

scheme that can protect users' privacy.

In the MA-ABE schemes, in order to protect data confidentiality against collusion attacks, the global identifier (GID) is usually tied to the decryption keys. Although the introduction of global identifier in secret keys enhances their uniqueness and can resist the collusion attack of unauthorized users to a certain extent, the user's privacy will inevitably be compromised if he/she directly sends the undisguised identifier to each authority for private key request. As shown in Figure 1, suppose that the authority $A_i (i = 1, 2, 3)$ manages the attribute set $\tilde{A}_i = \{a_{i,j}\}_{j=\{1,2\}}$, the user U who owns attributes $\{a_{1,1}, a_{2,1}, a_{3,1}\}$ and identifier u directly submits $(u, a_{i,1})$ to A_i for secret key request. By jointly tracking the same u [6], the colluding authorities can easily gather all the attributes attached to it. As a result, users' personal information $U(u, \{a_{1,1}, a_{2,1}, a_{3,1}\})$ is thoroughly exposed to them. Therefore, the hiding of GID is the primary task of constructing a privacy-preserving encryption scheme. Besides, the access policy also needs to be hidden because it indicates the legal recipient. Malicious users can infer the attributes of the recipient based on the access policy. This also compromises user privacy. Based on this, a secure decentralized scheme dedicated to privacy protection is proposed.

1.2 Related Work

Along with the development, many expansion schemes [1, 29, 35] were derived from the original ABE scheme [30]. In general, ABE comes in two categories: key-policy ABE (KP-ABE) [11] and ciphertext-policy ABE (CP-ABE) [2]. The CP-ABE scheme enables data owners to achieve absolute control over their data and decide who can and cannot access the data, since the access policy is determined by them. In the data encryption system, considering the computational and storage costs of the authorized organization, single-authority ABE scheme like [36] is no longer applicable. Since there is only a trusted central authority (CA) to manage all users, thus the computational and communication costs of CA have undoubtedly increased. More notably, the excessive power of CA will be a potential risk because it has all users' information and decryption keys, so that it can access all encrypted files. Once the CA is corrupted, the confidentiality of the data and the user privacy will be compromised. In addition, there are many categories of attributes in the real world, so it is impractical to have only one authority to manage them.

Chase [5] put forward a multi-authority ABE (MA-ABE) scheme to weaken the power of central authority (CA), in 2007. Instead of a single authority in the scheme, there are many authorities here to issue private keys for users. While sharing the pressure of single-authority, multiple authorities have differentiated its rights and provided a more secure and effective management mode. However, there is still a CA to manage other attribute authorities, and the multiple authorities need to cooperate to initialize the system. Chase and Chow [6] first introduced the concept of privacy protection and proposed a privacy-preserving MA-ABE scheme, which employed an anonymous key issuing protocol to achieve the goal of hiding the user's GID and employed a distributed pseudorandom functions (PRF) to get rid of the CA. Later, some MA-ABE schemes with privacy-preserving in PHR system have been proposed [19,27,32]. These schemes employed the distributed PRF technique in [6] to protect the GID information. In these schemes, there are multiple authorities that manage a set of disjoint attributes, though CA is not required, there is still a partnership among the multiple authorities namely every two authorities (A_k, A_j) must perform a 2-party key exchange to share the PRF seed $s_{k,j} = s_{j,k}$, which increases communication and computing costs. In 2011, Lewko and Waters [18] introduced a novel MA-ABE, namely decentralized attribute-based encryption scheme. Unlike the previous schemes, there is no CA in this scheme, and the authorities are independent of each other without any cooperation.

The first decentralized KP-ABE scheme considering user GID privacy was proposed by Han *et al.* [12], in 2012. Unfortunately, it turned out to be unsafe [10]. Then a series of improvement schemes [28, 34] were proposed. In scheme [34], Zhang *et al.* modified the original scheme and came up with a more secure scheme against user collusion. All the schemes mentioned above exploited the technique of hiding GID in [6] to produce the secret keys for users.

In 2015, Huang *et al.* [15] put forward a revocable MA-ABE scheme. In their scheme, a CA is needed to send the seed s_k to each authority A_k and generate the secret key component to users. To protect user GID and attributes from being leaked, Han *et al.* [13] proposed a PPDCP-ABE scheme. However, Wang *et al.* [31] found that it could neither resist the collusion attack nor achieve attributes hiding. In 2018, the scheme [14] gave a new idea. In their proposal, there exists an identity management (IDM) that actually acts as a fully trusted CA and multiple attribute authorities. IDM is responsible for generating a pseudonym $Pid_{U,j}$ for the user U corresponding to the authority A_j, and generating an anonymous identity certificate $(AID_{Cred,j})$ for the user by signing the pseudonym and attributes information. The $AID_{Cred,j}$ is

Table 1: Comparisons of the proposed scheme with other MA-ABE schemes

Schemes	Central Authority	*AA* Cooperation	GID Anonymous	Policy Anonymous	Decryption Outsourced	Collusion Resistance	
						User	Authority
Huang [15]	Yes	No	No	No	No	Yes	N-1
Feng [9]	Yes	Yes	Yes	Yes	No	Yes	N-2
Fan [7]	Yes	No	No	Yes	No	Yes	N-1
Hu [14]	Yes	No	Yes	No	No	Yes	N-1
Chase [6]	No	Yes	Yes	No	No	–	N-2
Qian [27]	No	Yes	Yes	No	No	–	N-1
Zhang [34]	No	No	Yes	No	No	Yes	N-1
Qian [26]	No	No	Yes	No	No	No	N-1
Feng [8]	No	No	No	Yes	No	Yes	–
Han [13]	No	No	Yes	No	No	No	N-1
Lyu [23]	No	No	Yes	No	Yes	No	N-1
Ours	No	No	Yes	Yes	Yes	Yes	N-1

sent to A_j for verification, and if the verification passes, the authority will generate partial secret keys for the user. During the whole process, the authority cannot know the user's GID and attributes information, so the user's privacy is protected. This scheme is resistant to users collusion and authorities collusion, however the existence of IDM and collaboration between IDM and multiple authorities suggests that further improvement is needed. In 2017, Lyu *et al.* [23] proposed a decentralized scheme to achieve GID anonymity and improve the efficiency by using the online/offline encryption and the verifiable outsource decryption. This scheme can tolerant $N-1$ compromised authorities. However, the authors did not mention the privacy of the access policy.

In consideration of GID and policy privacy, Qian *et al.* [26] employed the AND,OR gates on multi-valued attributes and put forward a PPDCP-ABE with fully hidden policy scheme in 2013. But a malicious user can guess the access structure with a simple test: $e(C_{i,j,1}, g) \stackrel{?}{=} e(\Pi_{k\in I_c} T_{i,j}^k, C_{i,j,2})$. Fan *et al.* [7] came up with a MA-ABE scheme for a hidden policy under the q-BDHE assumption. In 2018, a new access control system [8] was proposed by Feng *et al.*, where they divided attributes into attribute names and attribute values, and realized policy hiding by hiding attribute values. In this scheme, GID is bounded with the time period, namely $H_1(GID||Time)$, to resist the collusion attack of illegal users. In 2019, Feng *et al.* [9] improved the scheme [16] and proposed a privacy-preserving searchable CP-ABE scheme, which achieved attributes anonymity and collusion resistance. However, the CA is needed in this scheme to issue the random identity (RID) and user key (UK) for each user, where RID is used to replace the user's real identity to achieve identity anonymity.

1.3 Contributions

As shown in Table 1, some existing solutions either have weaknesses in privacy protection and collusion resistance, or require the CA or cooperation among multiple authorities to generate private keys for users. To improve the user privacy and data confidentiality, a privacy-preserving decentralized CP-ABE scheme is proposed, in which the CA is no longer needed and each authority can independently issue partial secret key for the user. The proposed scheme supports both user anonymity and collusion resistant, and the proxy server is introduced here to undertake partial decryption computation. The main contributions of the proposed paper are listed below.

- A decentralized CP-ABE scheme is presented to cater to the requirements of a distributed system where there is no central authority and multiple attribute authority can independently manage users' attributes and generate secret keys for them.
- By using the privacy-preserving key generation protocol (PPKeyGen) and composite order bilinear groups, we hide both GID and access policy, thus providing a higher privacy security.
- The proposed construction can resist collusion attacks of the unauthorized users and tolerant $N-1$ (N is the number of attribute authorities involved in encryption) corrupted authorities, which provides a guarantee for data confidentiality.
- Theoretical analysis and experimental simulations are given to enhance the credibility of the proposed scheme in terms of security and efficiency.

1.4 Organization

The organization of the rest paper is as follows. In Section 2, some preliminaries involved in this paper are listed. The system model and security model are described in Section 3. In Section 4, the proposed scheme is presented in detail. Sections 5 and 6 show the security analysis and performance analysis, respectively. A brief conclusion is given in Section 7.

2 Preliminaries

2.1 Composite Order Bilinear Groups

Assume that $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups with same order $N = pp_1$, where p and p_1 are two different primes, $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$ is a bilinear map. $\mathbb{G}_p$ and $\mathbb{G}_{p_1}$ are subgroups of the group $\mathbb{G}$ having order p and p_1 respectively. The map e satisfies the following features:

1) Bilinearity: $\forall f, h \in \mathbb{G}, \forall u, v \in \mathbb{Z}_N$, here is the equation $e(f^u, h^v) = e(f, h)^{uv}$.

2) Non-degenerate: $\exists f \in \mathbb{G}$ such that $e(f,h)$ in group $\mathbb{G}_T$ has order N.

3) Orthogonality: Let g_p and g_{p_1} be the generator of group $\mathbb{G}_p$ and $\mathbb{G}_{p_1}$, respectively. Notice that e is efficiently computable, and it actually satisfies another property: $e(g_p, g_{p_1}) = 1$.

2.2 Decisional Bilinear Diffie-Hellman (DBDH) Assumption

Suppose g is the generator of group $\mathbb{G}$, and a, b, c, z are random numbers selected from $\mathbb{Z}_p$. The DBDH assumption holds if given a tuple $(A, B, C, Z) = (g^a, g^b, g^c, Z)$, there is no algorithm $\mathcal{B}$ can distinguish $Z = e(g,g)^{abc}$ or $Z = e(g,g)^z$ in polynomial-time with non-negligible advantage. $\mathcal{B}$'s advantage is defined as:

$$Adv_{\mathcal{B}}^{DBDH} = |\Pr[\mathcal{B}(A, B, C, e(g,g)^{abc}) = 1] - \Pr[\mathcal{B}(A, B, C, e(g,g)^z) = 1]|.$$

2.3 Access Structure

The AND-gate on multi-valued attributes is applied in this scheme as an access policy.

Suppose $\mathcal{U} = [a_1, a_2, \cdots, a_n]$ is an attribute set, each attribute $a_i \in \mathcal{U}$ has n_i possible values $V_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,n_i}\}$. Let $S = [S_1, S_2, \cdots, S_n]$ be a user's attribute set where $S_i \in V_i$, and $W = [W_1, W_2, \cdots, W_n]$ be an access structure where $W_i \in V_i$. The symbol $S \models W$ indicates that the attribute set S satisfies the access structure W (namely, $S_i = W_i$), and $S \not\models W$ indicates the opposite.

2.4 Commitment

The commitment scheme adopted in this construction is the Pedersen commitment scheme [24], which is a perfectly hiding scheme. This scheme is stated as follows. Assume $\mathbb{G}$ is a prime-order group with generators $g_0, g_1, \cdots, g_k$. To commit messages $(m_1, m_2, \cdots, m_k)$, the user picks $r \in_R \mathbb{Z}_p$, and calculates $T = g_0^r \prod_{j=1}^{k} g_j^{m_k}$. The number r is used by the user to decommit the commitment T when needed.

2.5 Zero-Knowledge Proof

Zero knowledge proof (ZKP) is a kind of interactive proof that the prover proves some knowledge to the verifier without leaking them. The ZKP scheme proposed by Camenisch and Stadler [3] is briefly described as follows. Take the following formula as an example,

$$PoK\{(\alpha, \beta, \gamma) : y = g^\alpha h^\beta \wedge y_0 = g_0^\alpha h_0^\gamma\},$$

where $\{g, h\}$ and $\{g_0, h_0\}$ are generators of group $\mathbb{G}$ and $\mathbb{G}_0$, respectively. The values α, β and γ are knowledge that need to be proven, the remaining values are used by the checker to verify the equations.

3 Definition and Security Model

3.1 System Model

The system architecture is shown in Figure 2, where there are 5 entities: Data Owner, N Attribute Authorities, Cloud Storage Server, Proxy Server and Data User.

Data Owner: The owner encrypts data file under his/her own specified access policy, then upload the ciphertext to the cloud server.

Attribute Authorities: Authorities are responsible for generating secret keys for users, which are independent of each other and manage disjoint attribute sets. The authorities are not entirely credible as they will try to collude with other entities to gather useful information in order to obtain illegal profits.

Cloud Storage Server: The server is assumed to be an honest-but-curious entity that stores the encrypted data file uploaded by data owners and provides access channel for the users. It acts normally in most of time but may attempt to collect as much information as possible.

Proxy Server: The proxy server owns strong computing power and it only provides partial decryption computing service for users.

Data User: The user can issue secret key queries to the authorities and download any ciphertext on the cloud server. However, only when the user's attributes satisfy the access structure can the data be recovered. Users are assumed to be distrusted, and they have a tendency to conspire with other entities to illegally access the data file.

3.2 Outline of Decentralized ABE Scheme

Let $A_1, A_2, \cdots, A_N$ represent N attribute authorities, and each authority A_k monitors a disjoint attributes set $\tilde{A}_k$, namely, $\tilde{A}_k \cap \tilde{A}_j = \emptyset$ $(k, j \in [1, N] \wedge k \neq j)$. $\tilde{L}$ represents a list of attributes for the user U.

A decentralized ABE scheme has five algorithms as follows:

Global Setup(1^λ): Input λ as a security parameter, the algorithm outputs the system parameters PP.

Authority Setup(PP): Taking system parameters PP as input, each authority A_k executes this algorithm to generate its public-secret key pair (PK_k, SK_k).

KeyGen($PP, SK_k, GID, \tilde{A}_u^k$): Input the system parameters PP, secret keys SK_k, user's GID and attributes set $\tilde{A}_u^k$, where $\tilde{A}_u^k = \tilde{A}_k \bigcap \tilde{L}$, authority A_k performs this algorithm and returns secret keys SK_U^k for the user.

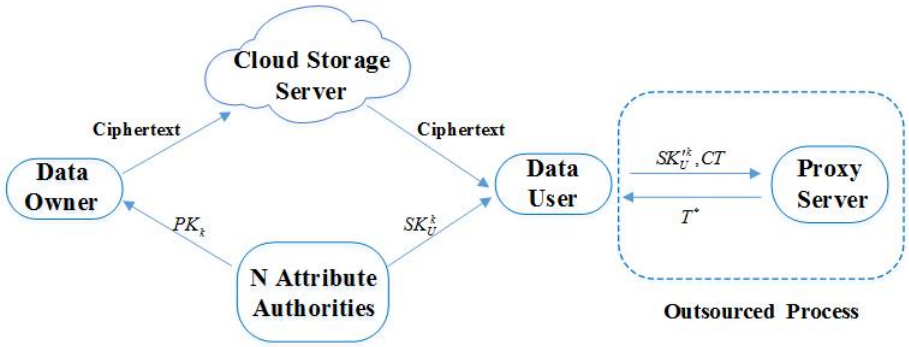


Figure 2: System model of the proposed scheme

Encrypt$(PP, PK_k, \mathcal{M}, W)$**:** With the system parameters PP, public keys PK_k, message $\mathcal{M}$ and access policy W as input, the encryption algorithm outputs CT as the corresponding ciphertext.

Decrypt(PP, GID, SK_U^k, CT)**:** This algorithm consists of two phases: *Pre-Decrypt* and *User-Decrypt*. The first phase is performed by the proxy server and the second is executed by the data user. The user first modifies the obtained secret keys SK_U^k to construct new keys $SK_U'^k$, and then sends the $(SK_U'^k, CT)$ to the proxy server for partial decryption calculations. Finally, the user performs the remaining decryption calculations with the results T^* returned by the proxy server.

3.3 Security Model

The security model is defined by a security game between adversary $\mathcal{A}$ and challenger $\mathcal{B}$.

Initialization: $\mathcal{A}$ provides an access policy $\mathcal{W}^*$ that he/she wants to challenge and a list of corrupted authorities $C_{\mathcal{A}}$ ($|C_{\mathcal{A}}| < N$) to $\mathcal{B}$.

Global Setup: $\mathcal{B}$ executes this algorithm and returns system parameters PP to $\mathcal{A}$.

Authority Setup: Two cases are discussed here:

1) For the corrupted authorities, namely $A_k \in C_{\mathcal{A}}$, $\mathcal{B}$ performs the authority setup algorithm and returns the public-secret key pair (PK_k, SK_k) to $\mathcal{A}$.
2) For the uncorrupted authorities, namely $A_k \notin C_{\mathcal{A}}$, $\mathcal{B}$ performs the authority setup algorithm and sends public keys PK_k to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ initiates a private key query to $\mathcal{B}$ by submitting a list of attributes $\tilde{L}$, and the only limitation is that the list of attributes submitted does not meet the access policy to be challenged, namely, $\tilde{L} \not\models \mathcal{W}^*$. Then $\mathcal{B}$ executes the algorithm KeyGen and outputs the secret keys to $\mathcal{A}$ correspondingly.

Challenge: The adversary $\mathcal{A}$ provides $\mathcal{M}_0$ and $\mathcal{M}_1$ to challenger $\mathcal{B}$, where $|\mathcal{M}_0| = |\mathcal{M}_1|$. Then $\mathcal{B}$ picks a random number $\xi \in \{0, 1\}$. The encryption algorithm is executed to encrypt the message $\mathcal{M}_\xi$ under the access policy $\mathcal{W}^*$ and outputs the ciphertext CT^* accordingly. Then $\mathcal{B}$ returns ciphertext CT^* to adversary $\mathcal{A}$.

Phase 2: Same as Phase 1.

Guess: $\mathcal{A}$ returns ξ' as a guess on ξ. If $\xi' = \xi$, $\mathcal{A}$ wins the game.

Definition 1. *If there is no t-time adversary who can make q secret key queries to break through the above game with a non-negligible advantage, then the scheme is* (t, q, ϵ) *secure in the selective model.*

4 Scheme Construction

4.1 Decentralized ABE Scheme

Global Setup(1^λ)**:** Taking the security parameter λ as input, this algorithm produces two cyclic groups $\mathbb{G}, \mathbb{G}_T$ with same order $N = pp_1$, where p and p_1 are two different primes, and a bilinear map, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Suppose that $\mathbb{G}_p$ and $\mathbb{G}_{p_1}$ are subgroups of $\mathbb{G}$ with order p and p_1, respectively. Let g_{p_1} be a generator of $\mathbb{G}_{p_1}$ and g_p, g_0, g_1 be generators of $\mathbb{G}_p$. $H : \{0,1\}^* \to \mathbb{Z}_N$ is a hash function. Public parameters are $PP = (e, p, p_1, g_p, g_0, g_1, g_{p_1}, H, \mathbb{G}, \mathbb{G}_T)$.

Authority Setup(PP)**:** Let $\tilde{A}_k$ be an attribute set monitored by authority A_k. Each A_k randomly selects $\alpha_k, \beta_k, a_{k,i,j} \in_R \mathbb{Z}_N$ and $R_k, R_{k,i,j} \in_R \mathbb{G}_{p_1}$ ($j = [1, n_i]$). Then A_k computes $Y_k = e(g_p, g_p)^{\alpha_k}, Z_k = g_p^{\beta_k} \cdot R_k$ and $A_{k,i,j} = g_p^{a_{k,i,j}} \cdot R_{k,i,j}$ as its public keys, namely $PK_k = (Y_k, Z_k, \{A_{k,i,j}\}_{j=[1,n_i]})$. The secret keys are $SK_k = (\alpha_k, \beta_k, \{a_{k,i,j}\}_{j=[1,n_i]})$.

KeyGen$(PP, SK_k, GID, \tilde{A}_u^k)$**:** Let $u = H(GID)$. Input PP, SK_k and user information $(u, \tilde{L})$, A_k performs the PPKeyGen algorithm to produce the decryption keys as follows. For $v_{k,i,j} \in \tilde{A}_u^k$, A_k picks $t_{k,i,j}^u \in_R \mathbb{Z}_N^*$ and sets $t_{k,u} = \sum t_{k,i,j}^u$. Then, A_k calculates:

$$D_{k,u} = g_p^{\alpha_k} g_0^{\frac{t_{k,u}}{\beta_k + u}} g_1^{u\beta_k}, D_{k,i,j} = g_0^{\frac{t_{k,i,j}^u}{(\beta_k + u)a_{k,i,j}}}$$

where $\tilde{A}_u^k = \tilde{A}_k \bigcap \tilde{L}$, $\tilde{L}$ represents an attribute list of the user.

Then, A_k outputs the secret keys

$$SK_U^k = (D_{k,u}, \{D_{k,i,j}\}_{v_{k,i,j} \in \tilde{A}_u^k}).$$

Encrypt$(PP, PK_k, \mathcal{M}, W)$**:** The owner employs the existing symmetric encryption algorithm to encrypt the data file M, and gets $CT' = Enc_{\mathcal{K}}(M)$, where $\mathcal{K}$ is the symmetric key. The decentralized ABE scheme is then applied to encrypt $\mathcal{K}$ under the access structure W defined by the encryptor. Finally, the owner uploads the ciphertext $\{CT, CT', CT''\}$ to the cloud server. Algorithm 1 gives the specific encryption steps. I_c represents an index set of relevant authorities A_k.

Algorithm 1 Encrypt

1: Begin
2: **Input** $\{PP, PK_k, M, \mathcal{K}, W\}$.
3: Select symmetric encryption algorithm Enc and key $\mathcal{K}$.
4: **for** data file M **do**
5: $CT' = Enc_{\mathcal{K}}(M)$.
6: **end for**
7: **for** symmetric key $\mathcal{K}$ **do**
8: $CT'' = g^{H(\mathcal{K})}$.
9: select $s \in_R \mathbb{Z}_N^*, R_1, R_2 \in_R \mathbb{G}_{p_1}$.
10: $C = \mathcal{K} \prod_{k \in I_c} Y_k^s, C_1 = g_p^s \cdot R_1, C_2 = (\prod_{k \in I_c} Z_k^s) \cdot R_2$.
11: **for** each attribute $v_{k,x,y} \in W$ **do**
12: select $R'_{k,x,y} \in_R \mathbb{G}_{p_1}$.
13: $C_{k,x,y} = A_{k,x,y}^s \cdot R'_{k,x,y}$.
14: **end for**
15: $CT = (C, C_1, C_2, \{C_{k.x,y}\}_{v_{k,x,y} \in W})$.
16: **end for**
17: **Output** $\{CT, CT', CT''\}$.
18: End

Decrypt(PP, GID, SK_U^k, CT)**:** Any user can download the ciphertext $\{CT, CT', CT''\}$ from the cloud server for decryption. To reduce the computational burden, the user can outsource some decryption operations to the proxy server. The decryption algorithm has two phases: *Pre-Decrypt* and *User-Decrypt*. The user first chooses $z \in_R \mathbb{Z}_N^*$, and then calculates the new keys NK. Then, the user sends (CT, NK) to the proxy server and keeps z secret.

Pre-Decrypt: This phase is executed by the proxy server. Input (CT, NK), the server performs partial decryption calculation and sends the result T^* to the user.

User-Decrypt: The user decrypts $\mathcal{K}$ through the secret value z and the result T^* returned by the server. Then, user uses the symmetric key $\mathcal{K}$ to get M.

The detail steps are shown in algorithm 2. In the check phase, the equation $g^{H(\mathcal{K})} \stackrel{?}{=} CT''$ is used to verify the correctness of the symmetric key $\mathcal{K}$. Only the user whose attributes satisfy the access policy can get the correct $\mathcal{K}$ and further run the symmetric decryption algorithm Dec to obtain the correct data file M. It can be known from the algorithm that user only needs to perform one exponential operation and one pairing operation to obtain the symmetric secret key $\mathcal{K}$, thus greatly reducing the computing cost of user.

Algorithm 2 Decrypt

1: Begin
2: **Input** $\{PP, SK_U^k, \{CT, CT', CT''\}\}$.
3: User selects a random number $z \in \mathbb{Z}_N^*$.
4: Calculate $SK_U'^k = (D'_{k,u}, D'_{k,i,j}) = (D_{k,u}^{1/z}, D_{k,i,j}^{1/z})$.
5: Set $NK = (SK_U'^k)$.
6: User sends (CT, NK) to the proxy server.
7: **Pre-Decrypt:**
8: Calculate $T^* = \frac{\prod_{k \in I_c} e(D'_{k,u}, C_1)}{\prod_{v_{k,x,y} \in W} e(D'_{k,i,j}, C_{k,x,y})}$.
9: Send T^* to the user.
10: **User-Decrypt:**
11: Calculate $B = e(C_2, g_1^u), \mathcal{K} = CB/(T^*)^z$.
12: Check $g^{H(\mathcal{K})} \stackrel{?}{=} CT''$
13: **if** $g^{H(\mathcal{K})} = CT''$ **then**
14: $M = Dec_{\mathcal{K}}(CT')$.
15: **else**
16: return $\perp$.
17: **end if**
18: **Output** M or $\perp$.
19: End

Correctness.

$$\begin{aligned} T^* &= \frac{\prod_{k \in I_c} e(D'_{k,u}, C_1)}{\prod_{v_{k,x,y} \in W} e(D'_{k,i,j}, C_{k,x,y})} \\ &= \frac{\prod_{k \in I_c} e((g_p^{\alpha_k} g_0^{\frac{t_{k,u}}{\beta_k + u}} g_1^{u\beta_k})^{1/z}, g_p^s \cdot R_1)}{\prod_{v_{k,x,y} \in W} e((g_0^{\frac{t_{k,i,j}^u}{(\beta_k+u)a_{k,i,j}}})^{1/z}, g_p^{sa_{k,x,y}} R_{k,x,y}^s R'_{k,x,y})} \\ &= \frac{\prod_{k \in I_c} (e(g_p, g_p)^{\alpha_k s} e(g_p, g_0)^{s \frac{t_{k,u}}{\beta_k + u}} e(g_p, g_1)^{su\beta_k})^{1/z}}{\prod_{k \in I_c} (e(g_p, g_0)^{s \frac{t_{k,u}}{\beta_k + u}})^{1/z}} \\ &= \prod_{k \in I_c} e(g_p, g_p)^{\alpha_k s/z} e(g_p, g_1)^{su\beta_k/z} \end{aligned}$$

$$B = e(C_2, g_1^u) = e(\prod_{k \in I_c} g_p^{\beta_k s} R_k^s \cdot R_2, g_1^u) = \prod_{k \in I_c} e(g_p, g_1)^{su\beta_k}$$

$$\mathcal{K} = BC/(T^*)^z = \frac{\mathcal{K} \cdot \prod_{k \in I_c} e(g_p, g_p)^{\alpha_k s} \cdot e(g_p, g_1)^{su\beta_k}}{(\prod_{k \in I_c} e(g_p, g_p)^{\alpha_k s/z} e(g_p, g_1)^{su\beta_k/z})^z}$$

Algorithm 3 PPKeyGen Protocol

Begin
U selects $u, \rho_0 \in_R \mathbb{Z}_N$, A_k selects $\rho_1, \beta_k \in_R \mathbb{Z}_N$
$U(u, \rho_0) \xleftrightarrow{2PC} A_k(\rho_1, \beta_k) : \eta = (u + \beta_k)\rho_0\rho_1 \bmod N$
U selects $z^*, z_1, z_2, z_3 \in_R \mathbb{Z}_N$, and computes $T = g_p^{z^*} g_1^u$, $P_0 = g_0^{\rho_0}, T' = g_p^{z_1} g_1^{z_2}$, $P_0' = g_0^{z_3}$
U returns (T, P_0, T', P_0') to A_k
A_k picks $c \in_R \mathbb{Z}_N$ and sends c to U
U sets $a_1 = z_1 - cz^*, a_2 = z_2 - cu, a_3 = z_3 - c\rho_0$ and sends (a_1, a_2, a_3) to A_k
A_k checks the zero-knowledge proof
if $T' = g_p^{a_1} g_1^{a_2} T^c$ and $P_0' = g_0^{a_3} P_0^c$ **then**
    $\forall v_{k,i,j} \in \tilde{A}_u^k$, A_k selects $t_{k,i,j}^u \in_R \mathbb{Z}_N$ and sets $t_{k,u} = \sum t_{k,i,j}^u$
    A_k selects $y_1, y_2, y_3, y_4, y_5 \in_R \mathbb{Z}_N$ and computes $P_1 = g_0^{\rho_1}$, $P_1' = g_0^{y_4}$, $Z_k' = g_p^{y_2}$, $\tilde{D}_{k,i,j} = P_1^{\frac{t_{k,i,j}^u}{\eta a_{k,i,j}}}$, $\tilde{D}_{k,u} = g_p^{\alpha_k} T^{\beta_k} P_0^{\frac{\rho_1 t_{k,u}}{\eta}}$, $\tilde{D}_{k,i,j}' = P_1^{y_5}$, $\tilde{D}_{k,u}' = g_p^{y_1} T^{y_2} P_0^{y_3}$
    A_k sends $(P_1, \tilde{D}_{k,u}, \tilde{D}_{k,i,j}, P_1', \tilde{D}_{k,u}', \tilde{D}_{k,i,j}', Z_k')$ to U
else
    Abort
end if
U chooses $c' \in_R \mathbb{Z}_N$, and sends c' to A_k
A_k sets $y_1' = y_1 - c'\alpha_k, y_2' = y_2 - c'\beta_k, y_3' = y_3 - c'\frac{t_{k,u}\rho_1}{\eta}, y_4' = y_4 - c'\rho_1, y_5' = y_5 - c'\frac{t_{k,i,j}^u}{\eta a_{k,i,j}}$
A_k sends $(y_1', y_2', y_3', y_4', y_5')$ to U.
U checks the proof
if $P_1' = g_0^{y_4'} P_1^{c'}$, $\tilde{D}_{k,u}' = g_p^{y_1'} T^{y_2'} P_0^{y_3'} (\tilde{D}_{k,u})^{c'}$, $\tilde{D}_{k,i,j}' = \tilde{D}_{k,i,j}^{c'} P_1^{y_5'}$ and $Z_k' = g_p^{y_2'} (Z_k)^{c'}$ **then**
    U computes $D_{k,u} = \frac{\tilde{D}_{k,u}}{Z_k^{z^*}}, D_{k,i,j} = \tilde{D}_{k,i,j}^{\rho_0}$, where $Z_k = g_p^{\beta_k}$
else
    Abort
end if
End

4.2 Privacy-Preserving Key Generation Protocol

Each user has a unique global identifier that distinguishes them from each other, if the GID is exposed to a malicious party, this will directly endanger the privacy. To achieve GID hidden, the knowledge of ZKP and commitment scheme can be utilized to conduct a privacy-preserving key generation protocol between the relevant authority and the user. Without revealing GID, user can prove his/her legitimacy to the authority and get the secret keys generated by the authority. The PPKeyGen algorithm is presented as follows.

1) The user U selects $u, \rho_0 \in \mathbb{Z}_N$, and the authority A_k selects $\rho_1, \beta_k \in \mathbb{Z}_N$. Then A_k executes the 2-party secure computing (2PC) protocol with U and gets $\eta = (u + \beta_k)\rho_0\rho_1 \bmod N$.

2) U picks $z^* \in_R \mathbb{Z}_N$ and runs the *Commit* algorithm on the GID u. Let T be the commitment value. U computes (T, P_0, T', P_0') and sends them to A_k. Moreover, U sends $PoK\{(u, \rho_0, z^*) : T = g_p^{z^*} g_1^u \wedge P_0 = g_0^{\rho_0}\}$ to A_k to anonymously prove that he/she has knowledge (u, ρ_0, z^*).

3) A_k checks the proof first. If the proof is correct, then A_k computes $(P_1, \tilde{D}_{k,u}, \tilde{D}_{k,i,j})$ and sends them to U. Otherwise, A_k will terminate the algorithm. Then, A_k sends $PoK\{(\alpha_k, \beta_k, \rho_1, t_{k,u}, a_{k,i,j}) : \tilde{D}_{k,u} \wedge \tilde{D}_{k,i,j} \wedge P_1\}$ to U to anonymously prove that it has knowledge $(\alpha_k, \beta_k, \rho_1, t_{k,u}, a_{k,i,j})$.

4) Similarly, A_k checks the zero-knowledge proof. If this proof works correctly, U can compute $D_{k,u}$ and $D_{k,i,j}$. Otherwise, U stops this algorithm.

Algorithm 3 illustrates the specific steps.

The PPKeyGen algorithm needs to meet two features: *leak-freeness* and *selective-failure blindness*. In the first one, a malicious user running the PPKeyGen algorithm with trusted authorities would not gain more information than executing the KeyGen algorithm. The second means that a malicious authority cannot get any information about the user's GID, nor can it fail the PPKeyGen algorithm based on the user's selection of GID. The definitions of the two features are shown below.

Definition 2. *(leak-freeness). A PPKeyGen algorithm is leak-free if for all efficient adversaries $\mathfrak{U}$, there exists a simulator $\mathfrak{U}'$ such that no efficient distinguisher $\mathfrak{D}$ can distinguish whether $\mathfrak{U}$ is executing in the real game or in the ideal game with non-negligible advantage. The two games are defined as follows:*

Real Game: The distinguisher $\mathfrak{D}$ runs the setup algorithm as many times as it wants, the malicious user $\mathfrak{U}$ selects a GID u and executes the PPKeyGen algorithm with the honest authority.

Ideal Game: The distinguisher $\mathfrak{D}$ runs the setup algorithm as many times as it wants, and the simulator $\mathfrak{U}'$ chooses a GID u' and executes the KeyGen algorithm with a trusted authority.

Definition 3. *(selective-failure blindness). A PPKeyGen algorithm is selective-failure blind if no probably polynomial time adversary $\mathcal{A}_k$ can win the following game with non-negligible advantage.*

1) The malicious authority $\mathcal{A}_k$ submits public key PK_k and two global identifiers u_0, u_1.

2) A random bit $b \in \{0, 1\}$ is selected.

3) $\mathcal{A}_k$ is given two comments com_b and com_{1-b}, then it can black-box access oracles $U(PP, u_b, PK_k, com_b)$ and $U(PP, u_{1-b}, PK_k, com_{1-b})$.

4) The algorithm U returns the secret keys $SK_{U_b}^k$ and $SK_{U_{1-b}}^k$, separately.

5) If $SK^k_{U_b} \neq \perp$ and $SK^k_{U_{1-b}} \neq \perp$, $\mathcal{A}_k$ is given $(SK^k_{U_b}, SK^k_{U_{1-b}})$; If $SK^k_{U_b} \neq \perp$ and $SK^k_{U_{1-b}} = \perp$, $\mathcal{A}_k$ is given $(\epsilon, \perp)$; If $SK^k_{U_b} = \perp$ and $SK^k_{U_{1-b}} \neq \perp$, $\mathcal{A}_k$ is given $(\perp, \epsilon)$; If $SK^k_{U_b} = \perp$ and $SK^k_{U_{1-b}} = \perp$, $\mathcal{A}_k$ is given $(\perp, \perp)$.

6) Finally, $\mathcal{A}_k$ returns its guess b' on b. $\mathcal{A}_k$ wins the game if $b' = b$.

5 Security Analysis

5.1 Scheme Analysis

Identity Privacy: The PPKeyGen algorithm introduced in this paper is an interactive process between the user and each authority, which ensures that the user can obtain the correct secret key from the authority without directly providing the unencapsulated u. In this case, the identifier u is confidential to all authorities, so they can no longer track it to aggregate user's information. **Theorem 2** in Section 5.2 shows that this protocol satisfies *leak-freeness* and *selective-failure blindness*.

Collusion Resistance: In this construction, the authors bind all the user's key components to u, making the user's secret key unique. The power settings of g_0 and g_1 (namely, $\frac{t_{k,u}}{\beta_k + u}$ and $u\beta_k$, respectively) make it impossible for different users to achieve effective attacks by combining their secret keys. What's more, the scheme can prevent the collusion attack of multiple authorities. By observing the ciphertext component C in CT (namely, $\mathcal{K}\prod_{k\in I_c} e(g_p, g_p)^{\alpha_k s}$), it can be known that only when all relevant α_k is obtained can the message $\mathcal{K}$ be restored. Therefore, as long as one authority is honest in the set of authority involved, the attack will not succeed.

Hidden Policy: The authors implement attributes anonymity in the access policy by applying some random elements $(R_1, R_2, R'_{k,x,y})$ in group G_{p_1} on some ciphertext components $(C'_1, C_2, C_{k,x,y})$ [20,33]. Such a setting is very necessary. The existence of these random elements makes it difficult for an attacker to easily guess the value of the attribute embedded by the encryptor in the access policy. Without the introduction of these random elements, the original ciphertext would be $C'_1 = g_p^s, C'_2 = (\prod_{k\in I_c} {Z'_k}^s), C'_{k,x,y} = A'^s_{k,x,y}$, where $Z'_k = g_p^{\beta_k}, A'_{k,i,j} = g_p^{a_{k,i,j}}$. An attacker needs only a simple test $e(A'_{k,i,j}, C'_1) \stackrel{?}{=} e(g_p, C'_{k,x,y})$ to be able to guess whether the encryptor has embedded the attribute value $v_{k,i,j}$ in the access policy or not. In fact,

$$X' = e(A'_{k,i,j}, C'_1) = e(g_p^{a_{k,i,j}}, g_p^s) = e(g_p, g_p)^{sa_{k,i,j}}$$

$$Y' = e(g_p, C'_{k,x,y}) = e(g_p, g_p^{sa_{k,x,y}}) = e(g_p, g_p)^{sa_{k,x,y}}$$

If $X' = Y'$, namely, $a_{k,i,j} = a_{k,x,y}$, means that $v_{k,i,j} \in W$; If $X' \neq Y'$, namely, $a_{k,i,j} \neq a_{k,x,y}$, means that $v_{k,i,j} \notin W$.

Do the same operations on the proposed scheme, then

$$\begin{aligned} X = e(A_{k,i,j}, C_1) &= e(g_p^{a_{k,i,j}} \cdot R_{k,i,j}, g_p^s \cdot R_1) \\ &= e(g_p, g_p)^{sa_{k,i,j}} e(R_{k,i,j}, R_1) \\ Y = e(g_p, C_{k,x,y}) &= e(g_p, g_p^{sa_{k,x,y}} \cdot R^s_{k,x,y} \cdot R'_{k,x,y}) \\ &= e(g_p, g_p)^{sa_{k,x,y}} \end{aligned}$$

An attacker cannot determine whether the attribute value $v_{k,i,j}$ belongs to W by comparing the values of X and Y. Therefore, the privacy of the recipient is protected to a certain extent.

5.2 Security Proof

Theorem 1. *The proposed scheme is secure in the selective access policy model, assuming that the DBDH assumption holds.*

Proof. Assuming that the proposed scheme can be broken by an adversary $\mathcal{A}$ with a non-negligible advantage ϵ, then, using the ability of $\mathcal{A}$, a simulator $\mathcal{B}$ can be constructed to solve the DBDH problem with the advantage $\epsilon/2$. □

Firstly, the challenger generates a bilinear group $(e, p, p_1, \mathbb{G}, \mathbb{G}_T)$, where $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_{p_1}$. Then he/she picks a random number $\varphi \in \{0, 1\}$, and sets:

$$\begin{cases} Z = e(g_p, g_p)^{abc}, \varphi = 0 \\ Z = e(g_p, g_p)^z, \varphi = 1 \end{cases} \tag{1}$$

where, z is a random number in group $\mathbb{G}$. The simulator gets the challenge tuple $(g_p, g_{p_1}, A, B, C, Z) = (g_p, g_{p_1}, g_p^a, g_p^b, g_p^c, Z)$ from the challenger and finally returns a guess φ' on φ.

Initialization: The adversary $\mathcal{A}$ provides a set of corrupted authorities C_A and an access structure $\mathcal{W}^*$ which he/she wants to challenge. Suppose there is at least one completely honest authority A^* in the security game.

Global Setup: $\mathcal{B}$ randomly chooses $\gamma, \theta \in_R \mathbb{Z}_N$ and sets $g_0 = A \cdot g_p^\gamma = g_p^{a+\gamma}, g_1 = g_p^\theta$. Then, $\mathcal{B}$ sends $PP = (e, p, p_1, g_p, g_0, g_1, g_{p_1}, H, \mathbb{G}, \mathbb{G}_T)$ to adversary $\mathcal{A}$.

Authority Setup: Three cases are discussed here:

1) For the corrupted authorities $A_k \in C_A$, $\mathcal{B}$ randomly selects $\alpha_k, \beta_k, a_{k,i,j} \in_R \mathbb{Z}_N$ and $R_k, R_{k,i,j} \in_R \mathbb{G}_{p_1}$, then calculates $Y_k = e(g_p, g_p)^{\alpha_k}, Z_k = g_p^{\beta_k} \cdot R_k$ and $A_{k,i,j} = g_p^{a_{k,i,j}} \cdot R_{k,i,j}$. Then, $\mathcal{B}$ sends the secret keys $SK_k = (\alpha_k, \beta_k, \{a_{k,i,j}\}_{j=[1,n_i]})$ and the public keys $PK_k = (Y_k, Z_k, \{A_{k,i,j}\}_{j=[1,n_i]})$ to adversary $\mathcal{A}$.

2) For the authority A_k not corrupted and not A^*, $\mathcal{B}$ randomly chooses $\alpha_k, \beta_k, a_{k,i,j} \in_R \mathbb{Z}_N$ and $R_k, R_{k,i,j} \in_R \mathbb{G}_{p_1}$, computes $Y_k = e(g_p, g_p)^{b\alpha_k}, Z_k = g_p^{\beta_k} \cdot R_k, A_{k,i,j} = g_p^{a_{k,i,j}} \cdot R_{k,i,j}$, when $v_{k,i,j} \in \mathcal{W}^*$, or $A_{k,i,j} = g_p^{ba_{k,i,j}} \cdot R_{k,i,j}$, when $v_{k,i,j} \notin \mathcal{W}^*$. $\mathcal{B}$ sends the public keys $PK_k = (Y_k, Z_k, \{A_{k,i,j}\}_{j=[1,n_i]})$ to adversary $\mathcal{A}$.

3) For the authority A^*, $\mathcal{B}$ randomly selects β^*, $a^*_{i,j} \in_R \mathbb{Z}_N$ and $R^*, R^*_{i,j} \in_R \mathbb{G}_{p_1}$, then computes $Y^* = e(g_p^a, g_p^b) \cdot \prod_{A_k \in C_A} e(g_p, g_p)^{-\alpha_k} \cdot \prod_{A_k \notin C_A \cup A^*} e(g_p, g_p)^{-b\alpha_k}$ and $Z^* = g_p^{\beta^*} \cdot R^*$. $A^*_{i,j} = g_p^{a^*_{i,j}} \cdot R^*_{i,j}$, if $v_{i,j} \in \mathcal{W}^*$; $A^*_{i,j} = g_p^{ba^*_{i,j}} \cdot R^*_{i,j}$, if $v_{i,j} \notin \mathcal{W}^*$. Then, $\mathcal{B}$ sends the public keys $PK^* = (Y^*, Z^*, \{A^*_{i,j}\}_{j=[1,n_i]})$ to adversary $\mathcal{A}$.

Phase 1: The authority $\mathcal{A}$ issues a secret keys query for global identifier u^* with a list of attributes L^*, where $L^* \not\models \mathcal{W}^*$.

1) For $A_k \in C_A$, $\mathcal{A}$ can generate the user secret keys directly by himself.

2) For $A_k \notin C_A \cup A^*$, $\forall v_{k,i,j} \in \tilde{A}^k_{u^*}$, $\mathcal{B}$ picks $t^{u^*}_{k,i,j} \in_R \mathbb{Z}_N$ and sets $t_{k,u^*} = \sum t^{u^*}_{k,i,j}$. Then, $\mathcal{B}$ computes $D_{k,u^*} = B^{\alpha_k} g_0^{\frac{t_{k,u^*}}{\beta_k + u^*}} g_1^{u^*\beta_k}, D_{k,i,j} = g_0^{\frac{t^{u^*}_{k,i,j}}{(\beta_k + u^*)a_{k,i,j}}}$.

3) For $A_k = A^*$, $\forall v_{i,j} \in \tilde{A}^*_{u^*}$, $\mathcal{B}$ chooses $t^{u^*}_{i,j} \in_R \mathbb{Z}_N$, sets $t_{u^*} = \sum t^{u^*}_{i,j}$, and computes $D^*_{u^*} = B^{-\gamma} g_0^{\frac{t_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*} \prod_{A_k \in C_A} g_p^{-\alpha_k} \prod_{A_k \notin C_A \cup A^*} B^{-\alpha_k}$ and $D^*_{i,j} = g_0^{\frac{t^{u^*}_{i,j}}{(\beta^* + u^*)a_{i,j}}}$. Where,

$$\begin{aligned} D^*_{u^*} &= B^{-\gamma} g_0^{\frac{t_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*} \prod_{A_k \in C_A} g_p^{-\alpha_k} \prod_{A_k \notin C_A \cup A^*} B^{-\alpha_k} \\ &= g_p^{-b\gamma} g_0^{\frac{t_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*} g_p^{-(\sum_{A_k \in C_A} \alpha_k + \sum_{A_k \notin C_A \cup A^*} b\alpha_k)} \\ &= g_p^{ab - (\sum_{A_k \in C_A} \alpha_k + \sum_{A_k \notin C_A \cup A^*} b\alpha_k) - b(a+\gamma)} g_0^{\frac{t_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*} \\ &= g_p^{ab - (\sum_{A_k \in C_A} \alpha_k + \sum_{A_k \notin C_A \cup A^*} b\alpha_k)} g_0^{\frac{t_{u^*}}{\beta^* + u^*} - b} g_1^{u^*\beta^*} \end{aligned}$$

Let $t'_{u^*} = t_{u^*} - b(\beta^* + u^*)$, then

$$D^*_{u^*} = g_p^{ab - (\sum_{A_k \in C_A} \alpha_k + \sum_{A_k \notin C_A \cup A^*} b\alpha_k)} g_0^{\frac{t'_{u^*}}{\beta^* + u^*}} g_1^{u^*\beta^*}$$

Therefore, $D^*_{u^*}$ is a valid secret key.

Challenge: The adversary $\mathcal{A}$ provides two messages of the same length $\mathcal{K}_0$ and $\mathcal{K}_1$. $\mathcal{B}$ chooses a random bit $\xi \in \{0,1\}$ and then encrypts the message $\mathcal{K}_\xi$: $CT^* = \{C^* = \mathcal{K}_\xi \cdot Z, C^*_1 = g^c \cdot R_1, C^*_2 = (\prod_{k \in I_c} g_p^{\beta_k c} \cdot R_k^c) \cdot R_2, \forall v_{k,x,y} \in \mathcal{W}^* : C^*_{k,x,y} = g_p^{ca_{k,x,y}} \cdot R^c_{k,x,y} \cdot R'_{k,x,y}\}$.

Phase 2: Same as Phase 1.

Guess: Adversary $\mathcal{A}$ returns the guess ξ' on ξ. If $\xi' = \xi$, simulator $\mathcal{B}$ returns $\varphi' = 0$ to the challenger; otherwise, simulator $\mathcal{B}$ returns $\varphi' = 1$.

If $\varphi = 1$, $Z = e(g_p, g_p)^z$ is a random value, adversary $\mathcal{A}$ cannot get any information about ξ, so $Pr[\xi' \neq \xi \mid \varphi = 1] = \frac{1}{2}$. Since $\mathcal{B}$ returns $\varphi' = 1$ when $\xi' \neq \xi$, thus $Pr[\varphi' = \varphi \mid \varphi = 1] = \frac{1}{2}$.

If $\varphi = 0$, then $Z = e(g_p, g_p)^{abc}$, the adversary $\mathcal{A}$ will get a valid ciphertext of message $\mathcal{K}_\xi$. The advantage of $\mathcal{A}$ in breaking the proposed scheme is ϵ (non-negligible) by definition, so $Pr[\xi' = \xi \mid \varphi = 0] = \frac{1}{2} + \epsilon$. Since $\mathcal{B}$ returns $\varphi' = 0$ when $\xi' = \xi$, thus $Pr[\varphi' = \varphi \mid \varphi = 0] = \frac{1}{2} + \epsilon$.

Therefore, the advantage of $\mathcal{B}$ to break the DBDH assumption is $| \frac{1}{2} Pr[\varphi' = \varphi \mid \varphi = 0] + \frac{1}{2} Pr[\varphi' = \varphi \mid \varphi = 1] - \frac{1}{2} | = \epsilon/2$ (non-negligible).

Theorem 2. *The proposed PPKeyGen algorithm is leak-free and selective-failure blind.*

Proof. (*Leak-freeness*) Suppose that there is a malicious user $\mathfrak{U}$ runs the PPKeyGen algorithm with an honest A_k in the real game. There should also exist a simulator $\tilde{\mathfrak{U}}$ runs the KeyGen algorithm with a trusted authority in the ideal game such that no distinguisher $\mathfrak{D}$ can effectively distinguish the two games. The simulator $\tilde{\mathfrak{U}}$ can simulate the communication between $\mathfrak{D}$ and $\mathfrak{U}$. $\tilde{\mathfrak{U}}$ works as follows:

1) $\tilde{\mathfrak{U}}$ sends PP and the public-key PK_k of authority A_k to the malicious user $\mathfrak{U}$.

2) The $\mathfrak{U}$ needs to prove to $\tilde{\mathfrak{U}}$ that he/she owns u in zero-knowledge by submitting two values (T, P_0). If the proof succeeds, then $\tilde{\mathfrak{U}}$ will gets (u, ρ_0, z^*) using rewind technique.

3) $\tilde{\mathfrak{U}}$ sends u to the trusted party and gets secret keys $(D_{k,u}, D_{k,i,j})$.

4) $\tilde{\mathfrak{U}}$ chooses $\rho \in_R \mathbb{Z}_p$, and calculates $\rho_1 = \rho/\rho_0, P_1 = g_0^{\rho_1}, \tilde{D}_{k,u} = D_{k,u} \cdot Z_k^{z^*}, \tilde{D}_{k,i,j} = D_{k,i,j}^{1/\rho_0}$. Then $\tilde{\mathfrak{U}}$ returns $(P_1, \tilde{D}_{k,u}, \tilde{D}_{k,i,j})$ to $\mathfrak{U}$.

□

If $(D_{k,u}, D_{k,i,j})$ are correct keys from the trusted AA in the ideal game, then $(\tilde{D}_{k,u}, \tilde{D}_{k,i,j})$ are correct keys from A_k in the real game. The distinguisher $\mathfrak{D}$ cannot distinguish the real game with the ideal game.

Proof. (*Selective-failure blindness*) The malicious authority A_k provides PK_k and two global identifiers (u_0, u_1). A random bit $b \in \{0,1\}$ is picked. A_k can

Table 2: The comparison of computing cost

Scheme	Setup	KeyGen	Encryption	Decryption
Huang [15]	$P+(N+2)E$	$(2\|A_U\|+N+1)E$	$(2\|A_c\|+2)E$	$(2\|A_c\|+N+1)P+\|A_c\|E$
Qian [26]	$P+(2N+m\cdot\|U\|)E$	$(2\|A_U\|+4N)E$	$(2\|A_c\|+2)E$	$(\|A_c\|+3N+1)P$
Qian [27]	$P+(3N+\|U\|)E$	$(\|A_U\|+3N(N-1))E$	$(\|A_c\|+2)E$	$(N\|A_c\|+1)P+(\|A_c\|+1)E$
Ours	$P+(2N+m\cdot\|U\|)E$	$(\|A_U\|+3N)E$	$(\|A_c\|+N+2)E$	$(\|A_c\|+N+1)P+E$

[1] P: the bilinear pairing operation. E: the exponential operation. $|*|$: the number of elements in $*$.
[2] A_U: the attribute set of U. A_c: the attribute set in ciphertext. m: the number of values for each attribute.

have a black box access to $U(u_b, com_b, PK_k, PP)$ and $U(u_{1-b}, com_{1-b}, PK_k, PP)$. Then, U runs PPKeyGen algorithm with A_k and returns secret keys $SK^k_{U_b}$ and $SK^k_{U_{1-b}}$: If $SK^k_{U_b} \neq \perp$ and $SK^k_{U_{1-b}} \neq \perp$, A_k is given $(SK^k_{U_b}, SK^k_{U_{1-b}})$; If $SK^k_{U_b} \neq \perp$ and $SK^k_{U_{1-b}} = \perp$, A_k is given $(\epsilon, \perp)$; If $SK^k_{U_b} = \perp$ and $SK^k_{U_{1-b}} \neq \perp$, A_k is given $(\perp, \epsilon)$; If $SK^k_{U_b} = \perp$ and $SK^k_{U_{1-b}} = \perp$, A_k is given $(\perp, \perp)$. Finally, A_k outputs a guess b' on b. □

In the PPKeyGen algorithm, U first computes T, P_0, and proves $PoK\{(u, \rho_0, z^*) : T = g_p^{z^*} g_1^u \wedge P_0 = g_0^{\rho_0}\}$. So far, the two oracles should be computationally indistinguishable to A_k. Otherwise, it will violate the commitment scheme's hiding property and the witness undistinguishable of the zero-knowledge proof. Assume that A_k outputs secret keys for the first oracle with some computing strategies. The next thing to prove is that A_k can predict secret keys for user without interaction with the two oracles:

1) A_k checks $PoK\{(\alpha_k, \beta_k, r_{k,u}, \rho_1, \eta) : \tilde{D}_{k,u} = g_p^{\alpha_k} T^{\beta_k} P_0^{\frac{\rho_1 t_{k,u}}{\eta}} \wedge P_1 = g_0^{\rho_1} \wedge \tilde{D}_{k,i,j} = P_1^{\frac{t^u_{k,i,j}}{\eta a_{k,i,j}}}\}$. If the proof fails, A_k outputs $SK^k_{U_0} = \perp$.

2) A_k generates different $(\tilde{D}_{k,u}, \tilde{D}_{k,i,j})$ for the second oracle and proves $PoK\{(\alpha_k, \beta_k, r_{k,u}, \rho_1, \eta) : \tilde{D}_{k,u} = g_p^{\alpha_k} T^{\beta_k} P_0^{\frac{\rho_1 t_{k,u}}{\eta}} \wedge P_1 = g_0^{\rho_1} \wedge \tilde{D}_{k,i,j} = P_1^{\frac{t^u_{k,i,j}}{\eta a_{k,i,j}}}\}$. If the proof fails, A_k outputs $SK^k_{U_1} = \perp$.

3) Finally, A_k returns its prediction on (u_0, u_1). If $SK^k_{U_0} \neq \perp$ and $SK^k_{U_1} \neq \perp$, the prediction is $(SK^k_{U_0}, SK^k_{U_1})$; If $SK^k_{U_0} \neq \perp$ and $SK^k_{U_1} = \perp$, the prediction is $(\epsilon, \perp)$; If $SK^k_{U_0} = \perp$ and $SK^k_{U_1} \neq \perp$, the prediction is $(\perp, \epsilon)$; If $SK^k_{U_0} = \perp$ and $SK^k_{U_1} = \perp$, the prediction is $(\perp, \perp)$.

The predication has the same distribution with the oracles as A_k performs the same check as the honest user. Therefore, if A_k can predict the user secret keys, then A_k, with or without the final outputs, has the same advantage. It means that A_k can distinguish the two oracles before the prediction. However, based on the security of commitment scheme and zero-knowledge proof, the advantage of A_k in distinguishing the two oracles is negligible.

6 Performance Analysis

In this section, Table 2 and Figure 3 show the comparisons of computation costs and efficiency, respectively. The simulation is executed on a Windows machine with 2.70 GHz Intel(R) Core (TM) i5-4210U CPU and 4 GB RAM. The implementation is based on Java Pairing-Based Cryptography (PBC) Library (version 0.5.14). Random elements in group G_{p_1} are introduced into our scheme to implement policy hiding. Apart from this function, the authors compare the scheme with schemes [15,26,27]. Figure 3(a) and Figure 3(b) show the comparisons of Setup time and KeyGen time with different number of attribute authorities. In this simulation, the number of attribute authorities is increased from 2 to 14, each authority monitors 5 attributes, and each attribute has 3 values. Scheme [15] uses a hash function to map the user's attributes to a random group element. Attribute authorities do not need to calculate the public keys related to the attributes, so the setup stage of the scheme takes less time. Figure 3(b) shows that the proposed scheme takes less time to generate the secret keys than the other three schemes. The reason why the secret key generation time of scheme [27] increases rapidly with the number of authorities is that every two authorities have to interact with each other. The KeyGen time is a quadratic function of the independent variable N. Figure 3(c) and Figure 3(d) show the comparisons of encryption time and decryption time with different number of attributes in the access policy, and the number of attribute authorities in the system is fixed at 5. This proposed scheme encryption time is longer than scheme [27], but the difference is very small. Figure 3(d) shows that the proposed scheme is superior to the other three schemes in the decryption phase.

7 Conclusion

To implement data sharing, a privacy-preserving decentralized ABE scheme is proposed in this paper. All attribute authorities are independent of each other and do not require any collaboration. All users can get the secret keys from authorities by using the PPKeyGen protocol without revealing his/her GID, which meets users' requirement to protect their private information. Besides, this scheme supports policy anonymity, so one cannot get any information about the user attributes. The security of the scheme is proved under a standard model and the simulation results verify the efficiency of the scheme. The disadvantage of the proposed scheme is that it only supports

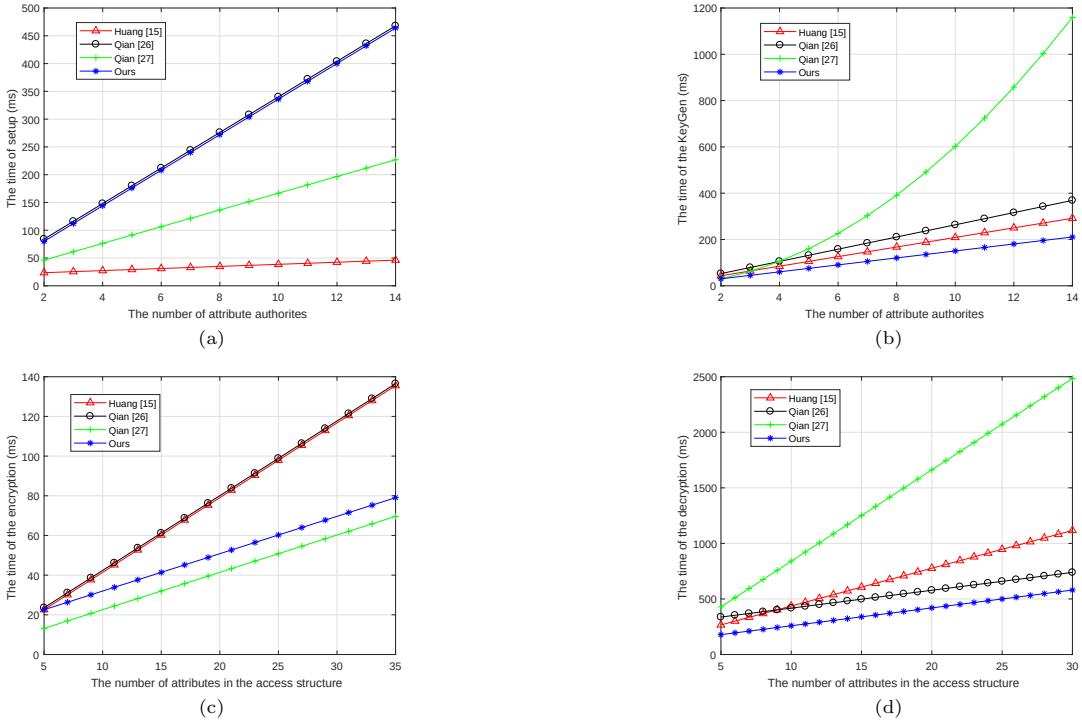


Figure 3: Efficiency comparisons of the proposed scheme with others

the AND-gate access structure. Constructing encryption schemes to support more flexible access structures are left as the future works.

Acknowledgments

This work was supported in part by the National Cryptography Development Fund under grant (MMJJ20180209).

References

[1] Y. Baseri, A. Hafid, and S. Cherkaoui, "Privacy preserving fine-grained location-based access control for mobile cloud," *Computers and Security*, vol. 73, pp. 249–265, 2018.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy*, pp. 321–334, 2007.

[3] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Advances in Cryptology (CRYPTO'97)*, pp. 410–424, 1997.

[4] Z. Cao, L. Liu, and Z. Guo, "Ruminations on attribute-based encryption," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, 2018.

[5] M. Chase, "Multi-authority attribute based encryption," in *Conference on Theory of Cryptography*, pp. 515–534, 2007.

[6] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Acm Conference on Computer and Communications Security*, pp. 121–130, 2009.

[7] Y. Fan, X. Wu, and J. Wang, "Multi-authority attribute-based encryption access control scheme with hidden policy and constant length ciphertext for cloud storage," in *IEEE Second International Conference on Data Science in Cyberspace*, pp. 205–212, 2017.

[8] T. Feng and J. Guo, "A new access control system based on CP-ABE in named data networking," *International Journal of Network Security*, vol. 20, no. 4, 2018.

[9] T. Feng, X. Yin, Y. Lu, J. Fang, and F.Li, "A searchable CP-ABE privacy preserving scheme," *International Journal of Network Security*, vol. 21, no. 4, 2019.

[10] A. Ge, J. Zhang, R. Zhang, C. Ma, and Z. Zhang, "Security analysis of a privacy-preserving decentralized key-policy attribute-based encryption scheme," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, 2013.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *ACM Conference on Computer and Communications Security*, pp. 89–98, 2006.

[12] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Parallel and Dis-*

tributed Systems, vol. 23, no. 11, pp. 2150–2162, 2012.

[13] J. Han, W. Susilo, and Y. Mu, *et al.*, "PPDCP-ABE: Privacy-preserving decentralized ciphertext-policy attribute-based encryption," in *Computer Security*, pp. 73–90, 2014.

[14] S. Hu, J. Li, and Y. Zhang, "Improving security and privacy-preserving in multi-authorities ciphertext-policy attribute-based encryption," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 10, 2018.

[15] X. Huang, Q. Tao, B.Qin, and Z. Liu, "Multi-authority attribute based encryption scheme with revocation," in *International Conference on Computer Communication and Networks*, pp. 1–5, 2015.

[16] T. Jung, X. Li, Z. Wan, and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, 2015.

[17] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal of Network Security*, vol. 15, no. 4, 2013.

[18] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology-eurocrypt -international Conference on the Theory and Applications of Cryptographic Techniques*, pp. 568–588, 2011.

[19] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, 2013.

[20] X. Li, D. Gu, Y. Ren, N. Ding, and K. Yuan, "Efficient ciphertext-policy attribute based encryption with hidden policy," in *Internet and Distributed Computing Systems*, pp. 146–159, 2012.

[21] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. H. Wang, "A survey of attribute-based access control with user revocation in cloud data storage," *International Journal of Network Security*, vol. 18, no. 5, 2016.

[22] L. Liu, Z. Cao, and C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, 2018.

[23] M. Lyu, X. Li, and H. Li, "Efficient, verifiable and privacy preserving decentralized attribute-based encryption for mobile cloud computing," in *IEEE Second International Conference on Data Science in Cyberspace*, 2017. DOI: 10.1109/DSC.2017.8.

[24] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *International Cryptology Conference on Advances in Cryptology*, pp. 129–140, 1991.

[25] H. S. G. Pussewalage and V. A. Oleshchuk, "A distributed multi-authority attribute based encryption scheme for secure sharing of personal health records," in *Acm on Symposium on Access Control Models and Technologies*, pp. 255–262, 2017.

[26] H. Qian, J. Li, and Y. Zhang, "Privacy-preserving decentralized ciphertext-policy attribute-based encryption with fully hidden access structure," in *International Conference on Information and Communications Security*, pp. 363–372, 2013.

[27] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *International Journal of Information Security*, vol. 14, no. 6, 2015.

[28] Y. Rahulamathavn, S. Veluru, J. Han, F. Li, M. Rajarajan, and R. Lu, "User collusion avoidance scheme for privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Transactions on Computers*, vol. 65, no. 9, 2016.

[29] S. Rezaei, M. A. Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, 2018.

[30] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology*, pp. 457–473, 2005.

[31] M. Wang, Z. Zhang, and C. Chen, "Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 4, 2015.

[32] F. Xhafa, J. Feng, Y. Zhang, X. Chen, and J. Li, "Privacy-aware attribute-based phr sharing with user accountability in cloud computing," *Journal of Supercomputing*, vol. 71, no. 5, 2015.

[33] R. Xu, Y. Wang, and B. Lang, "A tree-based CP-ABE scheme with hidden policy supporting secure data sharing in cloud computing," in *International Conference on Advanced Cloud and Big Data*, pp. 51–57, 2013.

[34] L. Zhang, P. Liang, and Y. Mu, "Improving privacy-preserving and security for decentralized key-policy attributed-based encryption," *IEEE Access*, vol. 6, pp. 12736–12745, 2018.

[35] L. Zhang and H. Yin, "Recipient anonymous ciphertext-policy attribute-based broadcast encryption," *International Journal of Network Security*, vol. 20, no. 1, 2018.

[36] Y. Zhang, D. Zheng, and R.H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet of Things Journal*, vol. 5, pp. 2130–2145, 2018.

Biography

Li Kang is a master degree student in the school of mathematics and statistics, Xidian University. Her research interests focus on computer and network security.

Leyou Zhang is a professor in the school of mathemat-

ics and statistics at Xidian University, Xi'an China. He received his PhD from Xidian University in 2009. From Dec. 2013 to Dec. 2014, he is a research fellow in the school of computer science and software engineering at the University of Wollongong. His current research interests include network security, computer security, and cryptography.

Evidence Gathering of Facebook Messenger on Android

Ming Sang Chang and Chih Ping Yen
(Corresponding author: Chih Ping Yen)

Department of Information Management, Central Police University
Taoyuan 33304, Taiwan
(Email: peter@mail.cpu.edu.tw)

(Received Aug. 28, 2019; Revised and Accepted Feb. 5, 2020; First Online Feb. 26, 2020)

Abstract

The trend in social networking is changing people's lifestyle. Since both the smart phone and computers are connected to the same tools, the newly developed applications must serve both ends to please the users. Therefore, the modes of cybercrime have also changed in accordance with the users' activities. In order to identify crimes, it is necessary to use appropriate forensic techniques to retrieve these traces and evidence. This study considers the social network, Facebook Messenger, as the research subject. We analyze the artifacts left on the Facebook Messenger application and show evidence of gathering, such as sending texts, pictures, and videos and making calls on the Android platform. This study explores the differences between the traces that are left on Non-Rooted and Rooted Android platform. Finally, the forensic analysis found, due to the differences in privacy control, can lead to discrepancies in recording the user behaviors on the same social network. It proves to be helpful to forensic analysts and practitioners because it assists them in mapping and finding digital evidences of Facebook Messenger on Android smart phone.

Keywords: Cybercrime; Digital Forensics; Facebook Messenger; Social Network

1 Introduction

Nowadays, social networking sites are increasingly popular. The popularity of social networking sites has given rise to the number of social networking users for business, recreation or any other likely purposes. Over the past few years, social networking sites have been significant mediums for people to enhance their interpersonal relationships. The prevalence of social networking websites has changed the living habits of many people. They share their emotion or daily life with their friends via texting, photographing or videoing. There is no doubt that people have integrated social networking sites into their lives and turned the use of social networking sites into daily activities.

Facebook Messenger is a messaging application. Originally developed as Facebook Chat in 2008, the company revamped its messaging service in 2010. It released standalone iOS and Android apps in August 2011. Over the years, Facebook has released new apps on a variety of different operating systems, launched a dedicated website interface, and separated the messaging functionality from the main Facebook app. Users can send messages and exchange photos, videos, stickers, audio, and files, as well as react to other users' messages and interact with bots. The service also supports voice and video calling. After being separated from the main Facebook app, Facebook Messenger had 600 million users in April 2015. This grew to 900 million in June 2016, 1 billion in July 2016, and more 1.3 billion monthly active users in October 2019 [22].

Social networking websites provide a virtual exchange space on the Internet for people with common interests, hobbies, and activities to easily share, discuss, and exchange their views without any limitation of space and time. Therefore, social networking websites continue to accumulate a large number of users. According to the Metcalfe's law, the value of a telecommunications network is proportional to the square of the number of connected users of the system. As a result, social networking has become a great force in today's society. However, this has also brought about endless criminal activities on social networks, such as cyberbullying, social engineering, and identity theft, among the other issues. Due to the following characteristics, the detecting cybercrime on social networks is different in comparison to other cybercrime [10]. Therefore, to assist the investigators in improving their efficiency of solving crimes, researches focusing on these upcoming technologies are needed [16].

1) Anonymity: Users are often unaware of the true identity of their counterpart in a social network because they are dealing with a fake account. Therefore, in the case of a social network cybercrime, it is difficult to extract the suspect's information and make arrests immediately [1].

2) Diffuseness: Any news published on the social network will be forwarded or shared immediately, which generates the diffusion effect [21]. Therefore, if a social network crime is not responded to immediately, it may cause the victim to suffer some serious damage.

3) Cross-Regional feature: Due to the nature of Internet, the location of the cybercrime is not necessarily the place where the criminal suspects are located. A bottleneck is formed during the crime investigation due to the difficulty in locating the suspects [13].

4) Vulnerability of evidence: The evidences obtained on social networks are in the form of digital data. In addition to the highly volatile nature of the digital evidences in the processing program from collection to storage, it is easy to change, delete, lose, or contaminate the digital evidences due to the anti-forensics operation of the suspects or negligence of the investigators [14].

This study considers the social network, Facebook Messenger, as the study subject. User activities are performed through Android smart phones. Forensic analysis is conducted to understand what type of user behavior leaves digital evidence on Android. We explore the differences between the traces that are left on Non-Rooted and Rooted Android platform. The results will be served as a reference for the future researchers in social network cybercrime investigation or digital forensics.

The rest of this paper is organized as follows. In the next section, we present our related works. In Section 3, we present our methodology. In Section 4, we present the results and findings of digital forensics on Facebook Messenger. In Section 5, we discuss the findings. Finally, we summarize our conclusions.

2 Related Works

The evidences were stored on three principle areas by using instant messenger program (IM). They are hard drive, memory, and network. Some IM services have the ability to log information on the user's hard drive. To use an IM, an account must be established to create a screen name provided with user information. Some instant messenger providers might assist the investigation with information of the account owner.

Evidence can be found in various internet file caches used by Internet Explorer for volatile IM and each cache holds different pieces of data. Apart from the normal files, files left by instant messenger on a hard drive can be in temp file format and will generally be deleted could be very difficult to retrieve once the machine is power down. An operating system generally stores information of all the installed and uninstalled applications in the system. The uninstalled application also leaves evidence. If a user has deleted an instant messenger application, there is a chance that a record can be found in the registry to prove that the instant messenger has once installed onto the system. Information is also stored within the memory. Since every application requires memory to execute, it is logical to think that there evidence could be left behind in the system's memory. The analysis on live memory allows us to extend the possibility in providing additional contextual information for any cases.

Presently, various researches focusing on the forensic analysis of social networking are being conducted. Artifacts of instant messaging have been of interest in many different digital forensic studies. Early work focused on artifacts left behind by many instant messaging applications, such as MSN Messenger [7], Yahoo Messenger [8], and AOL Instant Messenger [17]. In 2013 Mahajan performed forensic analysis of Whatsapp and Viber on five android phones using UFED and manual analysis [12]. Katie Corcoran forensic 7 Messaging applications and first analyzed the evidence of Facebook Messenger [5]. Levendoski concluded that artifacts of the Yahoo Messenger client produced a different directory structure on Windows Vista and 7 [11]. Wong and Al Mutawa demonstrated that artifacts of the Facebook web-application could be recovered from memory dumps and web browsing cache [15, 25].

Said investigated Facebook and other IM applications, it was determined that only BlackBerry Bold 9700 and iPhone 3G/3GS provided evidence of Facebook unencrypted [19]. Sgaras analyzed Skype and several other VoIP applications for iOS and Android platforms [20]. It was concluded that the Android apps store far less artifacts than of the iOS apps. Chu focused on live data acquisition from personal computer and was able to identify distinct strings that will assist forensic practitioners with reconstruction of the previous Facebook sessions [4]. The analysis was conducted on an iPhone running iOS6 and a Samsung Galaxy Note running Android 4.1. Walnycky analyzed 20 popular instant messaging applications for Android, of which Facebook Messenger can get evidences such as text chat, voice call, audio, video, image, location, and stickers [24]. Azfar adapt a widely used adversary model from the cryptographic literature to formally capture a forensic investigator's capabilities during the collection and analysis of evidentiary materials from mobile devices [2].

William Glisson explored the effectiveness of different forensic tools and techniques for extracting evidences on mobile devices [9]. In 2015, Nikos Virvilis presented studies based on the security of web browsers and reported the shortcomings and vulnerabilities of browsers operated on desktop and mobile devices. It was found that some browsers using secure browsing protocols had actually limited their own protection level [23]. Dezfouli examine four social networking: Facebook, Twitter, LinkedIn and Google+, on Android and iOS platforms, to detect remnants of users' activities that are of forensic interest [6]. In 2017, Yusoff report the results of investigation and analysis of three social media services (Facebook,

Twitter, and Google +) as well as three instant messaging services (Telegram, OpenWapp, and Line) for forensic investigators to examine residual remnants of forensics value in Firefox OS [27]. Song-Yang Wu describes several forensic examinations of Android WeChat and provides corresponding technical methods [26]. Imam Riadi performed a comparison of tool performance to find digital and chat and pictures from Instagram Messenger [18]. Although Zhang et al. analyzed the local artifacts of the 4 Instant Messaging applications, the types of these local artifacts are incomplete [28]. Jusop Choi analyzed the personal data files in three instant messaging applications (KakaoTalk, NateOn, and QQ) which are the most popularly used in China and South Korea [3].

This paper investigated the user activities of Facebook Messenger through Android smart phones. We conducted forensics on Non-Rooted and Rooted Android platform, and explored and compared the type of user behavior that leaves digital evidence on the device. The results will be served as a reference for the future researchers in the social network cybercrime investigation or digital forensics.

3 Methodology

In our research, we use the smart phone with an installation of Facebook Messenger. The study was focused on identifying data remnants of the activities of Messenger on an Android platform. This is undertaken to determine the remnants an examiner should search for when Instant Messenger is suspected. Our research includes the circumstances of Non-Rooted and Rooted Android platform.

3.1 Research Goal

This paper studies the user behaviors, including logging into Facebook Messenger, uploading images, exchanging information, GIS location sharing, and special application functions under the Non-Rooted and Rooted Android environment. The study also explored and compared the type of user behavior that leaves digital evidence on the device. We explore the differences between the artifacts that are left on Non-Rooted and Rooted Android platform. We checked the changes and discrepancies in the residual digital data and relevant evidence on the Android smart phone.

3.2 Experimental Environment and Tools

In this paper, all the experiments were conducted on the real system. This study is built on Sony Xperia Z1 C6902 with Android 4.3. Under the Android operating environment, the Facebook Messenger social networking application was installed to run the Messenger features directly. In addition, if Facebook Messenger have ever been installed, the "/com.facebook.orca" folder will appear in the directory structure.

Rooting is a process of allowing users to gain privileged control which is known as root over the various Android systems. The devices include mobile phones, tablets or any other electronic device that is running Android mobile operating system could obtain highest authority when they rooted the phone. Rooting is often carried out with the aim of overcoming limitations that mobile operators and developers put on some devices. In order to obtain more information on the mobile phone, the investigators should execute a series of rooting processes before examining a mobile phone.

XRY is a commercial digital forensics and mobile device forensics product by the Swedish company Micro Systemation. It used to analyze and recover information from mobile devices. XRY is designed to recover the contents of a device in a forensic manner so that the contents of the data can be relied upon by the user. The XRY system allows for both logical examinations and also physical examinations.

Autopsy is a free computer software that makes it simpler to deploy many of the open source programs. The graphical user interface displays the results from the forensic search of the underlying volume making it easier for investigators to flag pertinent sections of data. WinHex is a hex editor useful in data recovery and digital forensics. WinHex is a free powerful application that you can use as an advanced hex editor, a tool for data analysis, editing, and recovery, a data wiping tool, and a forensics tool used for evidence gathering.

SQLite is a software library that provides a relational database management system. SQLite database is integrated with the application that accesses the database. The applications interact with the SQLite database read and write directly from the database files stored on disk. SQLite is an open source. SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation. The physical smart phone uses SQLite to read and analyze the database files on mobile devices. Android debug bridge (ADB) is a versatile command line tool that lets users communicate with connected Android devices or emulators. Android debug bridge command also facilitates a variety of devices actions, for example, installing or debugging applications. All the specifications of the tools we used are listed in the Table 1.

3.3 Development of Experiments

Based on the experimental environment designed, we run the Messenger features, including logging in, sending messages, exchanging photos, videos, audio, and files, making a call, etc. After that, the relevant evidence on each device was extracted and analyzed using forensic tools.

3.3.1 Extract Data

XRY is a commercial tool specifically for mobile phone forensics. Besides XRY tool, we need backup the image file of smart phone to analyze for Autopsy and WinHex. We create an image file for physical memory on the smart

Table 1: List of hardware and software used

Devices/Tools	Description	Specification/Versions
Sony Xperia Z1 C6902	Android Smart Phone	Android 4.3, Memory 2GB/16GB
XRY	Mobile Forensics Tool	Version v7.4.1
Autopsy	Digital Forensics Tool	Version v4.4.1
WinHex	Digital Forensics Tool	Version v18.9
SQLite Expert Personal	Database Management Tool	Version v3.5.96.2516
Messenger	Social Networking App	Version v141.0.0.31.76
Minimal ADB and Fastboot	ADB Tool	Version v10.0.16299.371

phone and use forensic tools to extract and analyze important digital evidences from the image file.

We can extract data and create image file from physical memory. We connect the mobile phone with the computer by using the phone cable. First, we enter "adb devices" command to connect the two devices. If the two devices are connected, the message will show the list of devices attached. Next, we enter "adb shell" command to execute remote control and now the mark sign will become "$". Then, we need to obtain the administrator level permissions. Thus, we enter "su" command and the mark sign will become "#". Now, we can enter "busybox df -h" command to inspect the system partition, path, volume, space usage, available space and so on. However, most of the application data installed and stored on the phone would locate at the data partition. Therefore, we are interested in this data partition. The path of data partition is "/dev/block/by-name/data". Now, we use "dd" command to create an image file for this partition. "dd" command can perform physical imaging by adopting bit-by-bit method. We enter "busybox dd if=/dev/block/by-name/data of=/storage/MicroSD/test conv=noerror bs=4096" command to create an image file. The string behind "if" is a partition that we would create an image file. The string behind "of" is an image output path. In the experiment, we name the output image file "test.img" and store it on the external SD card. The "conv=noerror" shows that there is no interruption when there is an error occurs. The "bs" represents the block size that we would write and read per time.

3.3.2 Design of Analysis

In order to ensure the integrity of digital evidence and avoid interference between digital evidence, we divide the experiment into the following three cases according to different forensic tools XRY, Autopsy and WinHex. The three experimental scenarios include Scenario 1: XRY, Scenario 2: Autopsy, and Scenario 3: WinHex. And each experimental scenario, the same experimental steps are performed.

In addition, each of the above experimental scenarios is further divided into two models: non-root and root Android platforms.We then performed the same experiments on non-root and root Android platforms and compared the relevant evidence.

3.3.3 Experiment Elaboration

The experiment steps are summarized as follows.

1) We install the Messenger software on the smart phone, and the forensic tool on the personal computer.
2) We logged into the Messenger for running various features for any material evidence left by the users.
3) After the activities completed, the Messenger software is logged out.
4) Use the forensic tool to find out all kinds of artifacts about Messenger software on smart phone.
5) Perform a comprehensive evidences analysis.

4 Results and Findings

In this section, we will use three scenarios to describe the result and findings. Each scenario has two modes that are the Non-Rooted mode and Rooted mode. The details of result and findings are as follows.

4.1 Scenario 1: XRY

In the scenario 1, we follow the experiment steps as Section 3.3 and use the XRY tool to find the evidences of the activities on the Messenger.

4.1.1 Non-Rooted Mode

Using the XRY tool to find the evidences on the Messenger, we can't find the user account and password. The artifacts of user account can be found as Figure 1. The user's information is as Figure 1 that are phone number +886985028322, nickname Huang Gordon, profile picture URL, and Facebook number 100021304820523.

The friend list can be found as Figure 2. It includes the name and the profile picture URL of friends.

The artifacts of sending text, image, audio, video, GIS location, and GIF animation can be found. For an example, we could show the artifacts of sending video as Figure

Figure 1: The artifacts of user account

Related Application	Display Name	Name	Category	Related URL
Facebook	Nan Chia	Nan Chia	Friend	https://scontent.ftpe7-1.f...
Facebook	Shane Chen	Shane Chen	Friend	https://scontent.ftpe7-1.f...
Facebook	La Chen	La Chen	Friend	https://scontent.ftpe7-1.f...
Facebook Messenger		Nan Chia		

Figure 2: The artifacts of friend list

3. The video delivery time, the video URL of storage location, who sending the video can be found.

Figure 3: The artifacts of sending video

The artifacts of making a call can be found. We can find the calling record that includes the calling time, who making the call, and threads_db2 database. For an example, the artifacts of making a call is as Figure 4.

About the database, there are three main databases for the recovered Messenger artifacts. The threads_db2 database contains the sending messages. The call_log_db_10021304820523 database contains the setting of user account. 10021304820523 is the network login number. The contacts_db2 database contains the user information. The contacts_db2 database is on /data/data/com.facebook.katana/databases/. The threads_db2 database and the call_log_db_10021304820523 database are on /data/data/com.facebook.orca/databases. There are three significance tables in the threads_db2 database. It includes the message table, the thread users table, and the message_reactions table. The message table stores the sending messages. An example of message table is as Figure 5. The evidences of text, sender, and timestamp can be found on the table. We can find the user name, nickname, and network number in the thread users table. The stickers can be found on message_reactions table.

From the call_log_db_10021304820523 database, we can find the user network number. The contacts_db2 database contains the contacts table. It includes friend list. The nickname and the URL of profile picture are on the table. The path, /data/data/com.facebook.orca/, is the main storage location of Messenger. All the storage path of different artifacts is shown as Table 2.

Figure 4: The artifacts of making a call

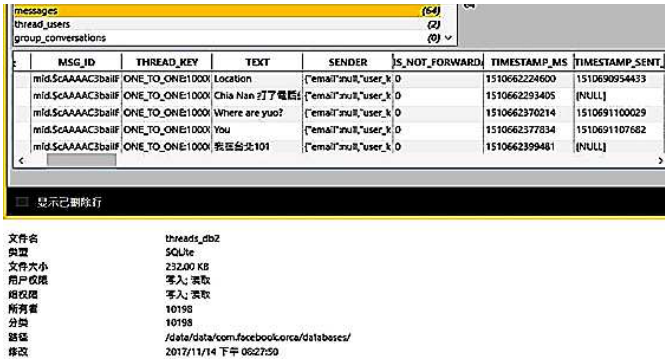


Figure 5: The artifacts of message table

Table 2: The storage path of various traces

Artifacts	Storage Path
Profile Picture	/data/data/com.facebook.orca/files/image/v2.ols100.1/52/
Friend Picture	/data/data/com.facebook.orca/files/image/v2.ols100.1/88/
Image	/data/data/com.facebook.orca/cache/image/v2.ols100.1/84/
Video	/data/data/com.facebook.orca/files/ExoPlayerCacheDir/videocache/
GIFAnimation	/data/data/com.facebook.orca/cache/image/v2.ols100.1/32/
Sticker	/data/data/com.facebook.orca/cache/image/v2.ols100.1/99/
Audio	/data/data/com.facebook.orca/cache/audio/v2.ols100.1/88/

4.1.2 Rooted Mode

Root is the highest privilege of the mobile phone, which is equivalent to the administrator privilege in the computer window system. After obtaining the root privilege, all the files of the mobile phone can be read and modified.

Using the forensic tool XRY, we can find the artifacts

of the Messenger account. It includes nickname, friend nickname, profile picture, user name, phone number, network number and related data URL. But the account and password cannot be found. The sending text message, picture, video, GIF animation, sticker, audio file, GIS location, and calling records are also can be found. It includes message content, sending time, name of the database, information about the sender and the recipient, etc. According to the experiment, the sending message is stored in the threads_db2 database, the account data is stored in the contacts_db2 database, and the network login number of the Messenger is known by the call_log_db_10021304820523database. The Rooted mobile phone experiment has the same results as the Non-rooted mobile phone experiment, and can find artifacts on various tables in the different database.

4.1.3 The Comparison of Findings on Rooted Mode and Non-Rooted Mode

The comparisons of findings between Rooted Mode and Non-Rooted Mode using XRY are as Table 3.

Table 3: The comparison of Rooted Mode and Non-Rooted Mode using XRY

Evidences	Rooted	Non-Rooted
Account	None	None
Password	None	None
Profile Name	Found	Found
Nickname	Found	Found
Friend Nickname	Found	Found
Text	Found	Found
Image	Found	Found
Video	Found	Found
GIF Animation	Found	Found
Stickers	Found	Found
Audio	Found	Found
GIS Location	Found	Found
Calling	Found	Found

4.2 Scenario 2: Autopsy

In the scenario 2, we follow the experiment steps as Section 3.3 and use the Autopsy tool to find the artifacts of the activities on the Messenger.

4.2.1 Non-Rooted Mode

The Autopsy is a free information security forensics tool that provides a graphical interface for digital forensic investigation. It can analyze Windows and UNIX disks and file systems such as NTFS, FAT, UFS1/2 and Ext2/3. In the case of Non-rooted mobile phone, we use the Autopsy tool to open the smart phone backup image file and analyze it. As a result, no artifacts about Messenger can be found.

4.2.2 Rooted Mode

We use the Autopsy tool to open the backup image file of smart phone. Our nickname, friend nickname, text, image, video, GIF animation, sticker, audio file, GIS location, and calling record can be found. But the account number, password, and user name can't be found. We also find the databases such as threads_db2, call_log_db_10021304820523 and contacts_db2. There is no data on the message_reactions table in the threads_db2 database. The Autopsy displays the artifacts of Messenger is as Figure 6.

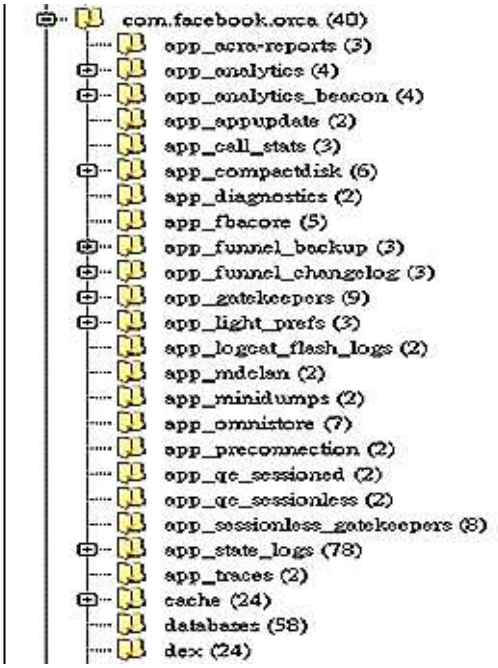


Figure 6: The artifacts of Messenger using Autopsy

User nickname and the network number can be found on the thread_users table in the threads_db2 database. We also find the sending text, sticker, GIS location, calling record on the messages table in the same database. The artifacts of video are located on data/com.facebook.orca/files/ExoPlayerCacheDir/videocache/. We find the artifacts of audio on data/com.facebook.orca/cache/audio/. The Image and GIF animation are on data/com.facebook.orca/cache/image/. For an example, the artifact of GIS location is shown in Figure 7.

Figure 7: The artifacts of GIS location using Autopsy

4.2.3 The Comparison of Findings on Rooted Mode and Non-Rooted Mode

The comparisons of findings between Rooted Mode and Non-Rooted Mode using Autopsy are as Table 4.

Table 4: The comparison of Rooted Mode and Non-Rooted Mode using Autopsy

Evidences	Rooted	Non-Rooted
Account	None	None
Password	None	None
Profile Name	None	None
Nickname	Found	None
Friend Nickname	Found	None
Text	Found	None
Image	Found	None
Video	Found	None
GIF Animation	Found	None
Stickers	Found	None
Audio	Found	None
GIS Location	Found	None
Calling	Found	None

4.3 Scenario 3: WinHex

In the scenario 3, we follow the experiment steps as Section 3.3 and use the WinHex tool to find the artifacts of the activities on the Messenger.

4.3.1 Non-Rooted Mode

WinHex is a disk editor and a hex editor useful in data recovery and digital forensics. WinHex is a free powerful application that you can use as an advanced hex editor, a tool for data analysis, editing, and recovery, and a forensics tool used for evidence gathering. In the case of Non-rooted mobile phone, we use the WinHex tool to open the smart phone backup image file and analyze it. As a result, no artifacts about Messenger can be found.

4.3.2 Rooted Mode

We use the WinHex tool to open the backup image file of smart phone. Our nickname, friend nickname, text, can be found. But the image, video, GIF animation, sticker, audio file, GIS location, calling record, account number, password, and user name can't be found. For an example, we find the user login email and nickname using the "username" keyword as Figure 8.

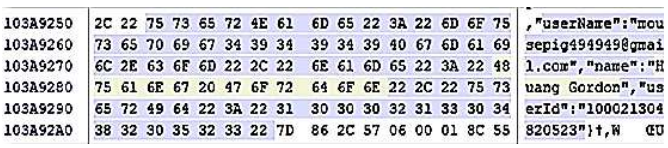


Figure 8: The artifacts of the user under using WinHex

4.3.3 The Comparison of Findings on Rooted Mode and Non-Rooted Mode

The comparisons of findings between Rooted Mode and Non-Rooted Mode using WinHex are as Table 5.

Table 5: The comparison of Rooted Mode and Non-Rooted Mode using WinHex

Evidences	Rooted	Non-Rooted
Account	None	None
Password	None	None
Profile Name	None	None
Nickname	Found	None
Friend Nickname	Found	None
Text	Found	None
Image	None	None
Video	None	None
GIF Animation	None	None
Stickers	None	None
Audio	None	None
GIS Location	None	None
Calling	None	None

5 Discussions

Using XRY tool, the path, /data/data/com.facebook.orca/, is the main storage location of Messenger. Under this path, there are two folders, files, and cache, have more multimedia artifacts. They include the profile picture, image, GIF animation, sticker, and audio files. The video artifacts are located in the ExoPlayerCacheDir folder. In addition, the storage path after the smart phone has been Rooted is the same as that of Non-rooted, but the pathname is changed to /userdata/data/com.facebook.orca/. Both the Non-rooted smart phone and the Rooted smart phone have the same artifacts. The reason is that the forensic tool XRY uses the way to downgrade the version of Messenger to get a lot of artifacts.

According to the experiments, in the case of the Non-rooted smart phone, the forensic tools, Autopsy, and WinHex, could not find any artifacts of Messenger. On the contrary, use the forensic tool XRY to bypass the security protection mechanism by downgrading the Messenger version. We can obtain many artifacts of the Messenger. But the account number and password cannot be found. Therefore, it is proved that the use of the forensic tool XRY to perform the forensic analysis in the Non-rooted smart phone is better.

In the Rooted smart phone, using the forensic tool Autopsy can find many traces, but the account number, password and user name cannot be found. Using the forensic tool WinHex can find our nickname, friend nickname, and text. According to the results of the experiments, it is

Table 6: The findings of three forensic tools

Forensic Tools / Messenger Artifacts	XRY		Autopsy		WinHex	
	Non-Rooted	Rooted	Non-Rooted	Rooted	Non-Rooted	Rooted
Account	None	None	None	None	None	None
Password	None	None	None	None	None	None
User name	Found	Found	None	None	None	None
My nickname	Found	Found	None	Found	None	Found
Friend nickname	Found	Found	None	Found	None	Found
Text	Found	Found	None	Found	None	Found
Image	Found	Found	None	Found	None	None
Video	Found	Found	None	Found	None	None
GIF animation	Found	Found	None	Found	None	None
Sticker	Found	Found	None	Found	None	None
Audio	Found	Found	None	Found	None	None
GIS location	Found	Found	None	Found	None	None
Calling log	Found	Found	None	Found	None	None

Table 7: The results of forensic analysis of relevant researchers for Facebook messenger

Researcher	**Artifacts**
Katie Corcoran [5]	User name, my nickname, friend nickname, text, GIS location, timestamp
Hao Zhang [28]	User name, my nickname, friend nickname, text, image, audio, GIS location, timestamp
Ming-Sang Chang & Chih-Ping Yen (this work)	User name, my nickname, friend nickname, text, image, video, GIF animation, sticker, audio, GIS location, calling log, timestamp

proved that the use of the forensic tool XRY to perform the forensic analysis in the Rooted smart phone and Non-Rooted smart phone is better. However, considering the funding or other restrictions, we can try to use the free forensic tool, Autopsy or WinHex, to assist in the forensic analysis work.

We know the investigation step of instant messaging from the research of the experiments. First, we find the basic information of the user, and then search for the behavior of the user through the keyword strings such as account number, nickname ,and network number. Using the user artifacts to estimate possible crimes. It also can use other accounts that may be additionally discovered during the search period. It may help to expand the search scope to see if there are accomplices or other victims.

We also can analyze the relationship between the criminal modus operandi and the timing chain. The investigator can infer the motives and tactics of possible crimes through the timing chain, and discover the criminal accomplices, transaction content, plans, time and place. When investigators find all kinds of traces of crimes, they can prevent crimes in advance.

Finally, we summarize the findings of three forensic tools based on Non-rooted mode and Rooted mode as Table 6. It also presents the results of forensic analysis of relevant researchers for Facebook Messenger, as shown in Table 7.

6 Conclusions

Although the instant messaging software has the advantages of convenience and immediacy, it is always inevitable that it will be abused by cyber criminals. Crimes are often exploited from software, website and web application exploits, using cloud services to spread malware, and further exploiting social media posts and links to trick users into fraud traps.

In this paper, we investigated the apps of Facebook Messenger to conduct a forensic analysis of the user behaviors in Android environments. The study found that different activities can lead to the discrepancies in recording the user behaviors on the same social network.

While investigating cybercrime on Facebook Messenger, we recommend that the first goal should be finding the account number, nickname, and network number of the criminal suspect. Using the account number and nickname, the operational behaviors of the criminal suspect on the social network can be searched, such as, uploading pictures, sending text, calling log, and timestamps. Then, based on the contents of the operation, the possible criminal activity or victimization practice can be deduced or estimated. At the same time, using the additional account numbers that are possibly discovered during the evidence gathering phase, the scope of the investigation can be expanded to find the possible accomplices or other victims. The full evidence scenario obtained in a step-by-

step and layer-by-layer outward expansion will be the key to solving the case.

References

[1] N. Al-Suwaidi, H. Nobanee, and F. Jabeen, "Estimating causes of cyber crime: Evidence from panel data fgls estimator," *International Journal of Cyber Criminology*, vol. 12, pp. 392–407, 2018.

[2] A. Azfar, K. Choo, and L. Liu, "An android social app forensics adversary model," in *Proceedings of the International Conference on System Sciences*, pp. 5597–5606, 2016.

[3] J. Choi, J. Yu, S. Hyun, and H. Kim, "Digital forensic analysis of encrypted database files in instant messaging applications on windows operating systems: Case study with kakaotalk, nateon and QQ messenger," *Digital Investigation*, vol. 28, pp. 50–59, Apr. 2019.

[4] H. C. Chu, D. J. Deng, and J. H. Park, "Live data mining concerning social networking forensics based on a facebook session through aggregation of social data," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 7, pp. 1368–1376, 2011.

[5] K. Corcoran, A. Read, J. Brunty, and T. Fenger, "Messaging application analysis for android and iOS platforms," *Research and Seminar in Forensic Science Graduate Program*, 2013. (`http://www.marshall.edu/forensics/research-and-seminar/class-of-2013`)

[6] F. N. Dezfouli, A. Dehghantanha, B. Eterovic-Soric, and K. R. Choo, "Investigating social networking applications on smartphones detecting facebook, twitter, linkedin and google+ artefacts on Android and iOS platforms," *Australian Journal of Forensic Sciences*, vol. 48, pp. 469–488, 2016.

[7] M. Dickson, "An examination into msn messenger 7.5 contact identification," *Digital Investigation*, vol. 3, no. 2, pp. 79–83, 2006.

[8] M. Dickson, "An examination into yahoo messenger 7.0 contact identification," *Digital Investigation*, vol. 3, no. 3, pp. 159–165, 2006.

[9] W. B. Glisson, T. Storer, and J. Buchanan-Wollaston, "An empirical comparison of data recovered from mobile forensic toolkits," *Digital Investigation*, vol. 10, no. 1, pp. 44–55, 2013.

[10] J. Golbeck, *Introduction to Social Media Investigation*, pp. 273-278, 2015. ISBN: 9780128016565.

[11] M. Levendoski, T. Datar, and M. Rogers, "Yahoo! messenger forensics on windows vista and windows 7," *Digital Forensics and Cyber Crime*, vol. 88, pp. 172–179, 2012.

[12] A. Mahajan, M. S. Dahiya, and H. P. Sanghvi, "Forensic analysis of instant messenger applications on android devices," *International Journal of Computer Applications*, vol. 68, no. 8, pp. 38–44, 2013.

[13] E. Martellozzo and E. A. Jane, *Cybercrime and Its Victims*, 2017. ISBN10: 1138639443.

[14] D. K. Mendoza, "The vulnerability of cyberspace – the cyber crime," *Journal of Forensic Sciences & Criminal Investigation*, vol. 2, no. 1, 2017.

[15] N. Mutawa, I. Awadhi, I. Baggili, and A. Marrington, "Forensic artifacts of facebook's instant messaging service," in *Proceedings of the International Conference for Internet Technology and Secured Transactions (ICITST'11)*, pp. 771–776, 2011.

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

[17] J. Reust, "Case study: Aol instant messenger trace evidence," *Digital Investigation*, vol. 3, no. 4, pp. 238–243, 2006.

[18] I. Riadi, A. Yudhana, and M. C. F. Putra, "Forensic tool comparison on instagram digital evidence based on android with the nist method," *Scientific Journal of Informatics*, vol. 5, no. 2, pp. 235–247, Nov. 2018.

[19] H. Said, A. Yousif, and H. Humaid, "Iphone forensics techniques and crime investigation," in *Proceedings of the International Conference and Work-shop on Current Trends in Information Technology*, pp. 120–125, 2011.

[20] C. Sgaras, M. T. Kechadi, and N. A. Le-Khac, "Forensics acquisition and analysis of instant messaging and voip applications," *Computational Forensics*, pp. 188–199, 2015.

[21] P. Shakarian, A. Bhatnagar, A. Aleali, E. Shaabani, and R. Guo, "Diffusion in social networks," *Computer Science*, 2015. ISBN: 978-3-319-23105-1.

[22] Statista, *Most Popular Global Mobile Messenger Apps as of October 2019, Based on Number of Monthly Active Users [Online]*, 2020. (`https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps`) [last accessed 10.01.2020].

[23] N. Virvilis, A. Mylonas, N. Tsalis, and D. Gritzalis, "Security busters: Web browser security vs.rogue sites," *Computer & Security*, vol. 52, pp. 90–105, 2015.

[24] D. Walnycky, I. Baggili, and A. Marrington, *et al.*, "Network and device forensic analysis of android social-messaging applications," *Digital Investigation*, vol. 14, pp. 77–84, 2015.

[25] K. Wong, A. Lai, J. Yeung, and W. Lee, *et al.*, "Facebook forensics," valkyrie-x security research group, 2011. (`https://www.fbiic.gov/public/2011/jul/facebook_forensics-finalized.pdf`)

[26] S. Y. Wu, Y. Zhang, and X. P. Wang *et al.*, "Forensic analysis of wechat on android smartphones," *Digital Investigation*, vol. 21, pp. 3–10, 2017.

[27] M .N. Yusoff, A. Dehghantanha, and R. Mahmod, "Chapter 4 – forensic investigation of social media and instant messaging services in firefox os: Facebook, twitter, google+, telegram, openwapp, and line as case studies," *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*, pp. 41–62, 2017.

[28] H. Zhang, L. Chen, and Q. Liu, "Digital forensic analysis of instant messaging applications on android smartphones," *International Conference on Computing, Networking and Communications (ICNC'18)*, pp. 647–651, 2018.

Biography

Ming Sang Chang received the Ph.D. degree from National Chiao Tung University, Taiwan, in 1999. In 2001 he joined the faculty of the Department of Information Management, Central Police University, where he is now a Professor. His research interest includes Computer Networking, Network Security, Digital Investigation, and Social Networks.

Chih Ping Yen is an Associate Professor, Department of Information Management, Central Police University. Received his Ph.D. degree from Department of Computer Science and Information Engineering, National Central University, Taiwan, in 2014. His research interest includes Digital Investigation, Artificial Intelligence & Pattern Recognition, Image Processing, and Management Information Systems.

Protection of User Data by Differential Privacy Algorithms

Jian Liu[1] and Feilong Qin[2]
(Corresponding author: Feilong Qin)
School of Automobile and Transportation, Chengdu Technological University[1]
School of Big Data and Artificial Intelligence, Chengdu Technological University[2]
No. 1, The second section of Zhongxin Avenue, Pidu District, Chengdu, Sichuan 611730, China
(Email: liujian@cdtu.edu.cn)
(Received Apr. 13, 2019; Revised and Accepted Jan. 5, 2020; First Online July 13, 2020)

Abstract

With the emergence of more and more social software users, increasingly larger social networks have appeared. These social networks contain a large number of sensitive information of users, so privacy protection processing is needed before releasing social network information. This paper introduced the hierarchical random graph (HRG) based differential privacy algorithm and the single-source shortest path based differential privacy algorithm. Then, the performance of the two algorithms was tested by two artificial networks without weight, which was generated by LFR tool and two real networks with weight, which were crawled by crawler software. The results show that after processing the social network through the differential privacy algorithm, the average clustering coefficient decreases, and the expected distortion increases. The smaller the privacy budget, the higher the reduction and the more significant the increase. Under the same privacy budget, the average clustering coefficient and expected distortion of the single-source shortest path differential privacy algorithm are small. In terms of execution efficiency, the larger the size of the social network, the more time it takes, and the differential privacy algorithm based on the single-source shortest path spends less time in the same network.

Keywords: Differential Privacy; Hierarchical Random Graph; Single Source Shortest Path Model; Social Network

1 Introduction

The popularity of wireless communication technology and intelligent mobile terminals makes people's communication more and more convenient, and a variety of community communication application software makes more and more registered users on the Internet, to build a vast and sophisticated social network [1, 12]. Social network contains different kinds of relevant information. Service providers of application software mine information using big data mining technology, analyze users' preferences, and provide more accurate personalized services [9]. However, the social network also contains sensitive private information, which is usually collected and archived by service providers, so the protection measures of privacy and confidential data become critical issues of service providers.

The traditional privacy protection is mainly to encrypt sensitive data, but this method is gradually challenging to play an active role in big data mining technology [13]. Differential privacy algorithm is a method to deal with the above problem. Its basic principle is to disturb the original data and network structure, including adding, deleting, exchanging, etc., to make the disturbing data different from the original data, *i.e.*, protecting original data through publishing the disturbed data. To reduce the large amount of noise caused by separate privacy in related data sets, Zhu *et al.* [15] proposed an effective correlated differential privacy solution. They found that the scheme was superior to the traditional differential privacy scheme in terms of mean square error on a large group of queries. Li *et al.* [7] proposed segmentation mechanisms based on privacy perception and utility to deal with the personalized privacy parameters of every individual in the data set and maximize the efficiency of the differential privacy calculation. Experiments a large amount of original data sets verified the effectiveness of the method.

Chen *et al.* [2] proposed two optimization techniques, PrivTHR and PrivTHREM, to optimize the differential privacy in wave clusters, and the simulation results showed that the optimization technique had high practicability when the privacy budget allocation was appropriate. This paper briefly introduced the differential privacy algorithm based on a hierarchical random graph (HRG) and the differential privacy algorithm based on the single-source shortest path. Then the performance of the two algorithms was tested by two artificial networks without weight, which was generated by LFR tool and two real

networks with weight, which were crawled by crawler software.

2 Differential Privacy Algorithm

2.1 The Concept of Differential Privacy

Social network is a network of points and lines in the visual image. Every node represents a user, while the line represents the connection between users. Points and lines in the social network diagram contain various vital data. At present, the commonly used social network privacy is divided into two categories, both of which substantially change the overall structure of the social network graph. One is to cluster the network nodes into "clusters" by using the clustering algorithm [8] and then encrypt them; the other is to add disturbance to the network graph structure, including deleting, exchanging, and adding nodes and connections. Although the former can hide the privacy data well, it seriously destroys the local structure of the network and affects the typical mining of the network structure data. Although the latter disturbs the network structure, the overall scale is the same, and the impact on the regular use of the data is not significant. Differential privacy is one of the protection methods of the latter. The definition of differential privacy is as follows. If the following equation holds:

$$Pr(F(D_1) \in S) \leq e^{\varepsilon} Pr(F(D_2) \in S).$$

Then the algorithm F can complete ε-differential privacy. D_1 and D_2 are two data sets which only had difference in one data; S is the output result of algorithm F to D_1 and D_2, and it is in the domain of definition of algorithm F; ε is called privacy budget [5], and its value determines the protection degree of differential privacy to data, in details, the smaller the value is, the higher the protection degree is and the larger the disturbance of data addition is.

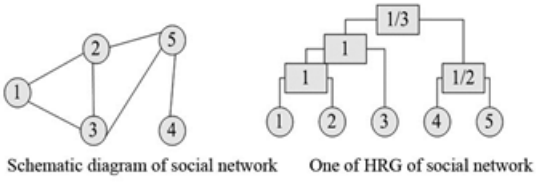


Figure 1: The schematic diagram of social network and one of its HRG

2.2 Differential Privacy Algorithm Based on HRG

HRG [6] divides the hierarchical structure of $G = (V, E)$ using binary tree, in which V is a set of nodes in a network and E is a set of relationships among network nodes. Figure 1 is one kind of HRG in the social network. The division of G by binary tree is similar to the random dichotomy of a node set. As shown in Figure 1, HRG dichotomizes five nodes into (1, 2, 3) and (4, 5) groups. The binary tree root (*i.e.* the box in Figure 1) of the two groups shows the connection probability of the two groups, and the formula is:

$$Pr = e_y/(n_{L,r} \cdot n_{R,r}),$$

where r is the internal node (root node) in the sample tree, *i.e.* the box node of HRG in Figure 1, $n_{L,r}$ and $n_{R,r}$ are the number of network nodes on the left and right sides under root node r, and e_r is the number of connection edges between node sets on both sides of the root node. After that, the dichotomy of groups continues, and the connection probability is calculated until the segmentation completes. The HRG based differential privacy algorithm is as follows.

1) Firstly, the sample tree of HRG is sampled by Markov Chain Monte Carlo (MCMC) method [3], and the details are as follows. A sample tree (HRG) T_0 is randomly selected. Then a neighbor tree is generated according to the previous sample tree, and whether to update the sample tree is determined according to the acceptance probability. The formulas are:

$$\begin{cases} T_i & = \begin{cases} T' & \alpha \\ T_{i-1} & 1-\alpha \end{cases} \\ \alpha & = \min(1, \exp(\frac{\varepsilon_1(\log L(T') - \log L(T_{i-1}))}{})) \\ \log L(T) & = -\sum_{r \in T} n_{L,r} n_{R,r} h(p_r) \\ h(p_r) & = -p_r \log p_r - (1-p_r)\log(1-p_r) \end{cases} \quad (1)$$

where T_i, T_{i-1}, and T' are sample trees after and before the update and the neighbor tree of T_{i-1} respectively, α is the probability of acceptance, $\log L(T)$ is the logarithm of similarity between the sample tree and G, $h(p_r)$ is Gibbs Shannon entropy function. Through Equation (1), the sample tree is updated and iterated until $\log L(T)$ before and after update and iteration is smaller than the set threshold. The number of samples is selected after a certain number of iterations, and finally the stable sample tree set $S_{ST} = (T_{S1}, T_{S2}, \cdots, T_{SN})$ is obtained through sampling, where T_{SN} stands for the sample tree which is obtained by the N-th sampling after the sample tree becomes stable through iterations.

2) Sample tree set S_{ST} is added with Laplace noise [5]. After noise addition, the calculation formula of the connection probability of node r inside the sample tree is:

$$Pr' = \min(1, \frac{e_r + laplace(\varepsilon_2^{-1})}{n_{L,r} \cdot n_{R,r}}),$$

where p'_r is the connection probability of internal node r after noise addition.

3) The lower triangular matrix of every HRG in S_{ST} after noise addition is calculated, and then the lower triangular mean value matrix of S_{ST} is calculated. The element in the lower triangular matrix is the connection probability of each pair of network nodes, which can be obtained through the multiplication of p'_r in HRG.

4) According to the connection probability of network nodes in the lower triangular mean value matrix, the connection edges between nodes is set.

2.3 Single Source Shortest Path Constraint Model Based Differential Privacy Algorithm

The HRG based differential privacy algorithm described above can effectively protect the privacy of social networks, but it is more aimed at the weightless social networks, *i.e.* although the degree of connection between nodes in the social networks in this algorithm is different, the importance of each connection is similar, or it does not matter to the algorithm. However, with the expansion of the scale of the Internet and the increase of social software users, social networks not only increase in scale, but also have different sensitivities between nodes. In order to describe social networks more accurately, in addition to the connection between nodes, different weights are also given to the connection, which is used to indicate the importance of the connection. The original expression of the social network transforms to: $G = (V, E, W)$, where W represents the weight set of the corresponding connection edges. For the social network with weight, the weight that it has is also part of the sensitive information, and moreover it t is also necessary to deal with the weight when dealing with the differential privacy of the social network as the importance degree of connection edges is represented by weight.

The HRG based differential privacy algorithm will affect the edge weight in the processing of differential privacy of social network with weight. Once the edge weight in the social network with weight changes, the structure of the whole graph will change; although the encryption of the information is achieved, the data availability seriously reduces. Therefore, the constraint model of social network was constructed by the single source shortest path algorithm [10] in this study, and linear constraints were applied to the disturbance of differential privacy on the basis of the constraint model. The single source shortest path constraint model based differential privacy algorithm is divided into two steps: 1) Building a single source shortest path constraint model; 2) Adding noise to differential privacy.

1) The first step is to build a single source shortest path constraint model. For $G = (V, E, W)$, nodes in the network are induced into the corresponding spanning tree using Dijkstra algorithm, and the constraint matrix representing the constraints is obtained. The relevant steps are as follows.

   a. Firstly, node is selected from the network as a source point and induced into to set V_0, and then nodes which can be reached in one step from v_0 are selected from the remaining nodes to form set Q.

   b. Node μ which has the smallest edge weight with V_0 is selected from Q. Then a row is added in constraint matrix A according to constraint condition $\{f(v_0, pre\_\mu) \leq f(v_0, \mu)\}$. Values of the corresponding positions in the matrix are constraint coefficients obtained by the set of constraint conditions. pre_μ is μ which is selected from Q previously. Then μ is induced into V_0, and the path between V_0 and Q is updated. Set Q is updated, *i.e.*, deleting μ.

   c. Step 2 repeats until Q becomes an empty set. Then the spanning tree which is composed of nodes in V_0 that has complete induction and corresponding connection edges is added to spanning tree sequence T.

   d. A new source point is selected from the remaining nodes which are not induced, and then Steps 1, 2, and 3 repeat until all the nodes are induced. Finally constraint matrix A and spanning tree sequence T are output.

2) After getting the single source shortest path constraint model of social network, noise is added to differential privacy. The noise addition of social network includes two aspects: one is to add constraint noise to the weight of the network connection edge, and the other is to disturb network nodes. The algorithm steps of the former are as follows.

   a. Firstly, according to the spanning tree in spanning tree sequence T, edge set E in G is divided into E_T and E_N, where E_T is the edge set of spanning tree and E_N is the remaining edge set.

   b. Laplace noise is added to the edge weight in E_T, and the formula of noise addition is:

   $$w'_i = w_i + laplace(\varepsilon_1),$$

   where $S(f)$ stands for the sensitivity of f, i stands for the edge of node pair which accepts search by f, and w_i and w'_i are edge weights of the corresponding node pair before and after noise addition respectively.

   c. The edge weight in E_N is solved based on the weight in E_T after noise addition, constraint matrix A and constraint inequation.

   d. Every spanning tree in spanning tree sequence T is processed as follows. Node pair which has no edge originally in the spanning tree is randomly

Table 1: Data sets of artificial network and Weibo social network

	Number of nodes	Number of edges	Average node degree	The maximum number of nodes in the community	The minimum number of nodes in the community
LFR1	1200	4125	30	110	30
LFR2	5000	9230	35	250	50
Weibo 1	12530	151132	55	1123	122
Weibo 2	21650	213578	64	1624	231

selected. A new edge is added between the node pair. The weight of the new edge is the smaller one among the maximum weight of the spanning tree and the shortest path of node pair.

The algorithm steps of network node disturbance are as follows.

a. Firstly, the number of nodes to be disturbed is calculated according to the set privacy budget,

$$N_n = |laplace(1/\varepsilon_2)|.$$

b. In order to reduce the influence of disturbance such as addition and deletion of nodes on the sensitivity of query function, nodes whose node degree is smaller than the set threshold are selected firstly. Node v is randomly selected, and then node set V_1 which is connected with v is processed as follows.

If $(\mu_1, v) \in E$, $(\mu_2, v) \in E$ and $f(\mu_1, \mu_2) = w(\mu_1, v) + w(v, \mu_2)$, then $w(\mu_1, \mu_2) = w(\mu_1, v) + w(v, \mu_2)$; if there is no edge between μ_1 and μ_2, then an edge is constructed. μ_1 and μ_2 are any two nodes in set V_1. f is a query function in the single source shortest path model, and it returns the shortest path between two nodes. After nodes in V_1 are processed, v and its edge are deleted.

The increase of virtual nodes is as follows. Node v is randomly selected. Then virtual node v_1 is added. A connection line is added between cv and v_1, and the weight of the connection edge is the average value of edge weights of other nodes which connected with v. Moreover, node μ which connects with v is also connected with v_1, and the estimation formula of its weight value is:

$$w(v_1, \mu) = w(\mu, v) + laplace(S(f)/\varepsilon_2).$$

c. Step 2 repeats to disturb network nodes until the number of disturbed nodes reaches N_n. After the noise addition of differential privacy for original social network G, social network G' is output.

3 Simulation Experiment

3.1 Experimental Environment

In this study, the coding of the above algorithm was realized by Python software [4]. The experiment was carried out with a laboratory server which was configured with Core i7 processor (2.6 GHz), Windows 7 operating system and 16 GB memory.

3.2 Experimental Setup

The performance of the two differential privacy algorithms was tested by the artificial network data set generated by LFR tool and the Weibo social network data set crawled by crawler software. The relevant parameters of the artificial network data set generated by LFR and the Weibo social network data set crawled from the Weibo interface by crawler software are shown in Table 1. LFR generated two artificial network data sets, and the artificial network also forms communities of different sizes for simulating the real network. In LFR1, there were 1200 nodes and 4125 edges, with an average node degree of 30; the maximum and minimum number of nodes in the community composed of nodes was 110 and 30 respectively. In LFR2, there were 5000 nodes and 9230 edges, with an average node degree of 35; The maximum and minimum number of nodes in the community composed of nodes was 250 and 50 respectively. The artificial network generated by LFR tool only contained node identification and connection relationship, which belongs to undirected network graph without weight. According to the preliminary statistics of two Weibo data networks which were composed of Weibo data crawled by crawler software, there were 12530 nodes and 151132 edges in Weibo 1, and an average node degree of 55, and the maximum and minimum number of nodes in the community was 1123 and 122 respectively; there were 21650 nodes and 213578 edges in Weibo 2, and an average node degree of 64, and the maximum and minimum number of nodes in the community was 1623 and 231 respectively. Besides the basic node identification and connection relationship, the real network which is composed of Weibo data also included weight information such as attribute labels, and the real network is a social network with weight.

For the above four social network data sets, the soil

networks are processed by the above two differential privacy algorithms. Privacy parameter ε of two algorithms in differential privacy processing was set as 10, 1 and 0.1 respectively.

1) Privacy parameter $\varepsilon = \varepsilon_1 + \varepsilon_2$ was used when the social network was processed by the HRG based differential privacy algorithm (Algorithm 2.2), where $\varepsilon_1 : \varepsilon_2 = 1 : 1$.

2) Privacy parameter $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$ was used when the social network was processed by the single source shortest path based differential privacy algorithm (Algorithm 2.3), where $\varepsilon_1 : \varepsilon_2 : \varepsilon_3 = 2 : 1 : 2$.

3.3 Performance Evaluation

In this study, the performance of the two algorithms was measured by average clustering coefficient, expected distortion degree and data processing time. The average clustering coefficient [14] could reflect the structure of social network. Comparing the average clustering coefficient of the network before and after the differential processing could understand the degree of privacy protection of an algorithm; the greater the difference was, the higher the degree of privacy protection was. The formula is:

$$C = \frac{1}{n}\sum_{i=1}^{n}\frac{2E_i}{k_i(k_i-1)},$$

where n is the total number of nodes, E_i is the actual number of connections between nodes adjacent to node i, and k_i is the number of nodes adjacent to node i.

The expected distortion degree [11] could reflect the degree of distortion of the data after differential privacy processing and could measure the availability of data; the larger the value was, the lower the degree of data distortion after processing was and the higher the availability was. The calculation formula is:

$$E[d(X, X')] = \sum_{X}\sum_{X'} p(x)q(x'|x)d(x, x'),$$

where X and X' are data sets before and after differential privacy processing respectively, $d(x, x')$ is the Hamming distance of the data before and after processing, $p(x)$ is the probability distribution of data before processing, and $q(x'|x)$ is the probability of differential privacy transfer condition.

Due to the randomness of the noise added in the differential privacy algorithm, the differential privacy algorithm of each social network was repeated 10 times under different privacy budgets, and the average value was taken as the final result.

3.4 Experimental Results

The average clustering coefficient could reflect the degree of clustering among nodes in the network, and it could reflect the structural distribution of the network to some extent. In this study, two differential privacy algorithms were applied to deal with four kinds of social networks under different privacy budgets. The average clustering coefficient before and after the processing is shown in Figures 2 and 3. Algorithm 2.2 represents the HRG based differential privacy algorithm; Algorithm 2.3 represents the single source shortest path based differential privacy algorithm, and numbers in brackets after the algorithm represent the privacy budget adopted. It was seen from Figures 2 and 3 that the average clustering coefficients of different social networks before and after differential privacy processing were different; the larger the scale of social networks was, the larger the average clustering coefficient was; the average clustering coefficient of real Weibo networks was significantly larger than that of artificial networks, which was because that connections between users in real networks are more close and frequent in addition to the reason of larger scale.

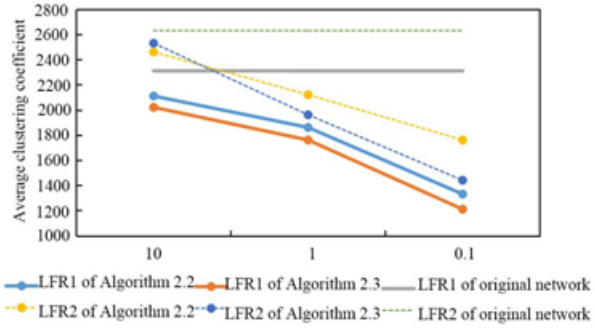


Figure 2: Average clustering coefficients of two LFR obtained by two algorithms under different privacy budgets

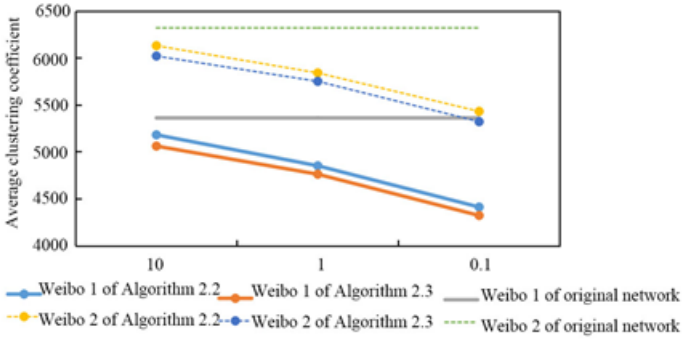


Figure 3: Average clustering coefficients of two Weibo networks obtained by two algorithms under different privacy budgets

The comparison of the average clustering coefficient under the same network data set suggested that the average clustering coefficient after differential privacy processing reduced; the smaller the privacy budget was, the more the reduction was. The comparison of the average clustering coefficient under the same privacy budget suggested that the average clustering coefficient of Algorithm 2.3 in the same social network was smaller. Overall, Algorithm 2.3 was better in the differential privacy protection

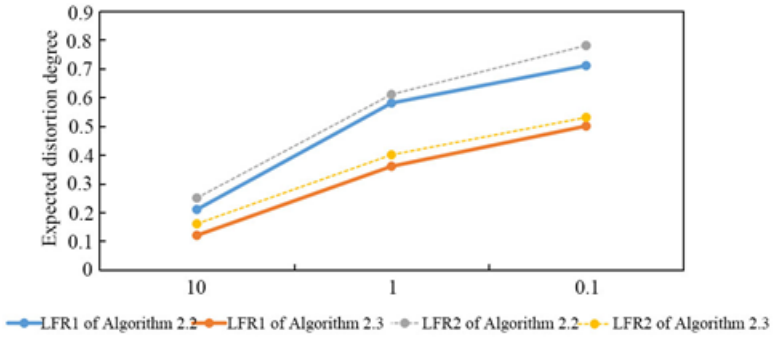


Figure 4: Expected distortion degrees of two LFR obtained by two algorithms under different privacy budgets

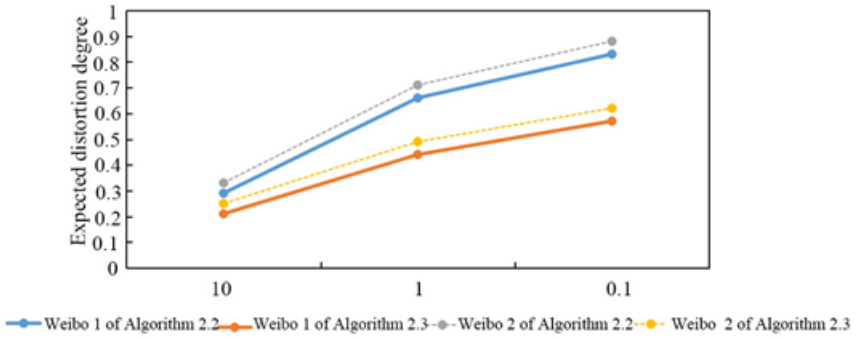


Figure 5: Expected distortion degrees of two Weibo networks obtained by two algorithms under different privacy budgets

of social networks.

The expected distortion degree could reflect the average degree of distortion between the original data set and the data set after differential privacy processing. This index measured the loss degree of effective information in the process of differential privacy processing of social networks. Once the loss degree of effective information was too large, social networks would not have the value of information mining. Under different privacy budgets, the expected distortion degree of the four social networks processed by the two differential privacy algorithms is shown in Figures 4 and 5. It was seen from Figures 4 and 5 that the expected distortion degree increased after differential privacy processing with the reduction of privacy budget no matter what kind of network it was; under the same privacy budget, no matter what kind of network it was, the expected distortion degree of Algorithm 2.3 was smaller; moreover, the expansion of social network scale also increased the expected distortion degree of networks after processing by algorithms.

The purpose of applying differential privacy algorithm to social network is to add noise to the privacy information, so as to achieve the effect of privacy protection. Therefore, in addition to the encryption effect, the execution efficiency of its encryption is also an important performance index. The average time of the two algorithms in processing differential privacy of four networks is shown in Figure 6. It was seen from Figure 6 that the expansion of social network scale and the existence of weights significantly increased the time required for differential privacy processing; under the same social network, the average time required by Algorithm 2.3 was significantly less than that of Algorithm 2.2, *i.e.* the single source shortest path based differential privacy algorithm was more efficient for differential privacy processing of social networks. The HRG based difference privacy algorithm needed to generate neighbor trees constantly in constructing the most matched HRG and sampled after converging to stability; in this process, it takes some time to converge to stability and sample. The single source shortest path constraint model completed at one time without repeated generation and convergence, so it took less time.

4 Conclusion

This paper briefly introduces the differential privacy algorithm based on HRG and the differential privacy algorithm based on a single-source shortest path. Then, two artificial networks without weights generated by LFR Gongzu and two real networks with weights crawled by searcher software were used to test the performance of these two algorithms. The results are as follows.

1) After the two differential privacy algorithms process the community network, the average clustering coefficient is reduced; the lower the privacy budget, the

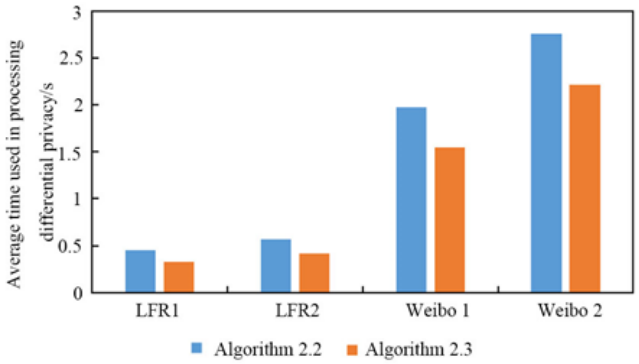


Figure 6: The average time of two algorithms for differential privacy processing

greater the reduction; under the same privacy budget, the single-source shortest path algorithm can reduce more.

2) After the community network is processed by the differential privacy algorithm, the smaller the privacy budget of the algorithm, the greater the expected distortion of the network; under the same privacy budget, the expected distortion of the network processed by the algorithm based on the single-source shortest path is smaller.

3) As the scale of social networks increases, the time required for the two algorithms to process social networks also increases, and the algorithm based on the single-source shortest path requires less time to process the same social network.

References

[1] A. Bhardwaj, V. Avasthi, S. Goundar, "Impact of Social Networking on Indian Youth: A Survey," *International Journal of Electronics and Information Engineering*, vol. 7, no. 1, pp. 41-51, 2017.

[2] L. Chen, T. Yu, R. Chirkova, "Wave cluster with differential privacy," *Computer Science*, vol. 11, no. 2, pp. 191–198, 2015.

[3] L. Chen, P. Zhu, "Preserving network privacy with a hierarchical structure approach," in *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'15)*, 2015.

[4] G. Eibl, D. Engel, "Differential privacy for real smart metering data," *Computer Science Research & Development*, vol. 32, no. 1-2, pp. 173-182, 2016.

[5] A. Friedman, S. Berkovsky, M. A. Kaafar, "A differential privacy framework for matrix factorization recommender systems," *User Modeling and User-Adapted Interaction*, vol. 26, no. 5, 2016.

[6] K. Kalantari, L. Sankar, A. D. Sarwate, "Robust privacy-utility tradeoffs under differential privacy and hamming distortion," *IEEE Transactions on Information Forensics & Security*, vol. 13, no. 11, pp. 1-1, 2018.

[7] H. Li, L. Xiong, Z. Ji, X. Jiang, "Partitioning-based mechanisms under personalized differential privacy," pp. 615-627, 2017.

[8] R. Rogers, A. Roth, A. Smith, O. Thakkar, "Max-information, differential privacy, and post-selection hypothesis testing," 2016.

[9] T. Steinke, J. Ullman, "Between pure and approximate differential privacy," *Computer Science*, vol. 8096, no. 2, pp. 363-378, 2015.

[10] X. C. Wang, Y. D. Li, "Geo-social network publication based on differential privacy," *Frontiers of Computer Science*, vol. 12, no. 6, 2018.

[11] X. Wu, Y. Wei, Y. Mao, L. Wang, "A differential privacy DNA motif finding method based on closed frequent patterns," *Cluster Computing*, vol. 21, pp. 1-13, 2018.

[12] B. Yang, I. Sato, H. Nakagawa, "Bayesian differential privacy on correlated data," in *ACM Sigmod International Conference on Management of Data*, 2015.

[13] D. Zhang, D. Kifer, "LightDP: Towards automating differential privacy proofs," *ACM Sigplan Notices*, vol. 52, no. 1, pp. 888-901, 2016.

[14] G. Q. Zhou, S. Qin, H. F. Zhou, "A differential privacy noise dynamic allocation algorithm for big multimedia data," *Multimedia Tools & Applications*, vol. 78, no. C, pp. 1-19, 2018.

[15] T. Zhu, P. Xiong, G. Li, W. Zhou, "Correlated differential privacy: Hiding information in non-IID data set," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 229-242, 2015.

Biography

Jian Liu, born in 1975, has received the doctor's degree. He is a lecturer in Chengdu Technological University. He is interested in Internet of vehicles, big data analysis and risk management.

Feilong Qin, born in 1983, has received the doctor's degree. He is a lecturer in Chengdu Technological University. He is interested in mathematical geology and applied statistics.

Verifiable Attribute-based Keyword Search Encryption with Attribute Revocation for Electronic Health Record System

Zhenhua Liu[1], Yan Liu[1], Jing Xu[1], and Baocang Wang[2]
(Corresponding author: Yan Liu)
School of Mathematics and Statistics, Xidian University[1]
Xi'an 710071, P. R. China
State Key Laboratory of Integrated Services Networks, Xidian University[2]
(Email: ly10_xidian@163.com)
(Received Apr. 7, 2019; Revised and Accepted Dec. 1, 2019; First Online Jan. 29, 2020)

Abstract

Considering the security requirements of electronic health record (EHR) system, we propose a ciphertext-policy attribute-based encryption scheme, which can support data retrieval, result verification and attribute revocation. In the proposed scheme, we make use of the BLS signature technique to achieve result verification for attribute-based keyword search encryption. In addition, key encrypting key (KEK) tree and re-encryption are utilized to achieve efficient attribute revocation. By giving thorough security analysis, the proposed scheme is proven to achieve: 1) Indistinguishability against selective ciphertext-policy and chosen plaintext attack under the decisional q-parallel bilinear Diffie-Hellman exponent hardness assumption; 2) Indistinguishability against chosen-keyword attack under the bilinear Diffie-Hellman assumption in the random oracle model. Moreover, the performance analysis results demonstrate that the proposed scheme is efficient and practical in electronic health record system.

Keywords: Attribute-Based Encryption; Attribute Revocation; Electronic Health Records; Keyword Search; Verifiability

1 Introduction

Electronic health record (EHR) system can provide health record storage service that allows patients to store, manage and share their EHR data with intended clients [13]. With the development of electronic health record system, much sensitive information from patients is being uploaded into the cloud. Since the cloud server may be dishonest, it is of vital importance to protect the confidentiality of the sensitive EHR data. Furthermore, it remains to be solved that how to securely share and search EHR data without revealing the information of patients.

Traditional public key encryption can only support "one-to-one" model, which is not suitable for multi-client data sharing in EHR scenarios. Fortunately, Sahai and Waters [16] first proposed the concept of attribute-based encryption (ABE) in 2005, which can provide "one-to-many" service and be considered as one of the most appropriate encryption technologies for cloud storage. Attribute-based encryption contains two variants: Ciphertext-Policy ABE (CP-ABE), where the ciphertext is associated with access policy, and key-policy ABE (KP-ABE), where a client's secret key is associated with access policy. Furthermore, Narayan *et al.*, [12] proposed a privacy preserving EHR system using attribute-based encryption technology in 2010, which enables patients to share their data among health care providers in a flexible, dynamic and scalable manner. Li *et al.*, [9] designed a new ABE scheme for personal health records system using multi-authority ABE, which avoids the key escrow problem. Reedy *et al.*, [14] proposed a secure framework for ensuring EHR's integrity, and solved the key escrow issue by using two-authority key generation scheme. Since then, some attribute-based encryption schemes [5, 8] for EHR system have been presented.

Although attribute-based encryption can achieve fine-grained data sharing, there are many problems to be considered in practical applications. For example, when a client leaves the system or discloses the secret key, it is essential to revoke the client's attributes or secret key. In order to solve the problem, a lot of revocable ABE schemes (RABE) [3, 17, 23] have been put forward. Yu *et al.*, [25] presented a revocable CP-ABE scheme by using proxy re-encryption, which allows an untrusted server to update a ciphertext into a new ciphertext without decryption. Hur *et al.*, [7] proposed an attribute-based access control scheme with efficient revocation in data outsourcing system using key encrypting key (KEK) tree. By using Chinese remainder theorem, Zhao *et al.*, [26] in-

troduced an efficient and revocable CP-ABE scheme in cloud computing. However, there is few RABE schemes for electronic health record system.

In addition, attribute-based encryption can protect data confidentiality, but hinder data retrieval from encrypted data in cloud storage. To address this issue, searchable encryption (SE) is proposed. SE contains two types: symmetric searchable encryption (SSE) and asymmetric searchable encryption (ASE). Song *et al.*, [18] first proposed the concept of symmetric searchable encryption. Boneh *et al.*, [1] introduced the first public-key encryption with keyword search (PEKS) scheme, and formalized a well-defined security notion of semantic security under chosen-keyword attack. After that, a lot of searchable encryption schemes [6, 15, 21] have been proposed. Furthermore, searchable encryption has widely been used in electronic health record system. For example, Xhafa *et al.*, [24] presented an efficient fuzzy keyword search scheme with multi-user over encrypted EHR data. Florence *et al.*, [4] proposed an enhanced secure sharing of personal health record system scheme with keyword search in cloud.

Searchable encryption allows a client to search over the encrypted data in cloud storage to retrieve the interested data without decryption. Nevertheless, the semi-trusted cloud server maybe performs search operation on the encrypted data and only returns a fraction of the results. In order to resist the cloud server's dishonest behavior, the verification technique [20] was introduced. Zheng *et al.*, [27] proposed a verifiable attribute-based keyword search scheme using bloom filter and digital signature techniques, which has good performance in search efficiency, but needs huge computational overhead in the verification process. Sun *et al.*, [19] introduced a verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud, but the verification efficiency is low. Furthermore, Miao *et al.*, [10] proposed a verifiable multi-keyword search over the encrypted cloud data for dynamic data-owner.

Unfortunately, the above existing schemes can not achieve fine-grained access control with attribute revocation, data retrieval and result verification for EHR system, simultaneously.

1.1 Our Contributions

Based on Waters' scheme [22], we will propose a verifiable attribute-based keyword search encryption scheme with attribute revocation (VABKS-AR) for electronic health record system. Our contributions are described as follows:

1) We can achieve efficient attribute-level revocation by using a KEK tree and re-encryption. A KEK tree is utilized to distribute attribute group key and re-encryption assures that the updated ciphertext cannot be decrypted by the revoked clients.

2) Since the cloud service provider is semi-trusted, the result verification mechanism [10] is used to achieve the verifiability for attribute-based keyword search encryption, which can reduce the computational overhead of the client.

3) We provide thorough analysis of the security and performance of the proposed secure EHR sharing system. The performance analysis results show that the proposed scheme is efficient and practical for electronic health record system.

1.2 Organization

The rest of this paper is organized as follows. We describe some preliminaries and system architecture in Section 2. A formal definition and security model are given in Section 3. The proposed VABKS-AR scheme is presented in Section 4. The security proof and performance analysis are given in Section 5. Finally, we make the conclusions in Section 6.

2 Preliminaries and System Architecture

2.1 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p, and g be a generator of $\mathbb{G}$. The map $\hat{e}$: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is said to be an admissible map if it satisfies the following properties [22]:

1) **Bilinearity**: $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_p$.

2) **Non-degeneracy**: $\hat{e}(g, g) \neq 1$.

3) **Computability**: There is an efficient polynomial-time algorithm to compute $\hat{e}(g, g)$.

2.2 Access Structure

Let $P_1, P_2, \cdots, P_n$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{P_1, P_2, \cdots, P_n}$ is monotone for $\forall B, C$: if $B \in A$ and $B \subseteq C$, then $C \in A$. An access structure [22] (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $P_1, P_2, \cdots, P_n$, i.e. $\mathbb{A} \subseteq 2^{P_1, P_2, \cdots, P_n} \backslash \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

2.3 Linear Secret Sharing Scheme (LSSS)

A linear secret-sharing scheme [22] Π over a set of parties P is described as follows:

1) The shares of each party form a vector over $\mathbb{Z}_p$.

2) There exists a share-generating matrix M for Π, where M has ℓ rows and n columns. For all $i = 1, 2, \cdots, \ell$, the function ρ labels the i-th row of M as $\rho(i)$. Consider the vector $\vec{v} = (s, r_2, \cdots, r_n)$, where $s \in \mathbb{Z}_p$ is a secret to be shared, and $r_2, \cdots, r_n \in \mathbb{Z}_p$

are chosen at random. $\mu_i = M_i \cdot \vec{v}$ is one of ℓ shares of the secret s according to Π, where $M_i \in \mathbb{Z}_p^n$ is the i-th row of the matrix M. The share $M_i \cdot \vec{v}$ belongs to party $\rho(i)$.

Linear reconstruction property [22]: Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \cdots, \ell\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\mu_i\}$ are valid shares of any secret s according to Π, we have $\Sigma \omega_i \mu_i = s$. These constants ω_i can be found in polynomial time in the size of the share-generating matrix M.

2.4 Bilinear Diffie-Hellman (BDH) Assumption

Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups with prime order p, and g be a generator of $\mathbb{G}$. Given a tuple $\vec{y} = (g, g^a, g^b, g^c)$, where a, b, c are selected from $\mathbb{Z}_p$ randomly. The Bilinear Diffile-Hellman (BDH) problem [1] is to compute $\hat{e}(g,g)^{abc} \in \mathbb{G}_T$. An algorithm $\mathcal{B}$ has at least advantage ε in solving the Bilinear Diffile-Hellman (BDH) problem if

$$\Pr[\hat{e}(g,g)^{abc} \leftarrow \mathcal{B}(\vec{y})] \geq \varepsilon.$$

BDH Assumption: We say the BDH assumption [1] holds if no probabilistic polynomial time algorithm can solve the BDH problem with a non-negligible probability ε.

2.5 Decisional Parallel Bilinear Diffie-Hellman Exponent Assumption

Let $\mathbb{G}$ be a group with order p, and g be a generator of $\mathbb{G}$. Given

$$\vec{y} = \Big(g, g^s, g^a, \cdots, g^{a^q}, g^{a^{q+2}}, \cdots, g^{a^{2q}}, \forall_{1 \leq j \leq q},\ g^{s \cdot b_j}, g^{a/b_j}, \cdots, g^{a^q/b_j}, g^{a^{q+2}/b_j}, \cdots, g^{a^{2q}/b_j}, \forall_{1 \leq k,j \leq q, k \neq j},\ g^{a \cdot s \cdot b_k/b_j}, \cdots, g^{a^q \cdot s \cdot b_k/b_j}\Big),$$

where $a, s, b_1, \cdots, b_q \in \mathbb{Z}_p$ are chosen randomly, the decisional q-parallel bilinear Diffie-Hellman exponent (BDHE) problem [22] is to distinguish a valid tuple $\hat{e}(g,g)^{a^{q+1} \cdot s} \in \mathbb{G}_T$ from a random element $R \in \mathbb{G}_T$. An algorithm $\mathcal{B}$ has advantage ε in solving the q-parallel BDHE problem if

$$\Big|\Pr[\mathcal{B}(\vec{y}, \hat{e}(g,g)^{a^{q+1}s}) = 0] - \Pr[\mathcal{B}(\vec{y}, R) = 0]\Big| \geq \varepsilon.$$

Decisional q-Parallel BDHE Assumption: We say the decisional q-parallel BDHE assumption [22] holds if no probabilistic polynomial time algorithm can solve the decisional q-parallel BDHE problem with a non-negligible probability ε.

2.6 KEK Tree

Let $\mathcal{U} = \{u_1, u_2, \cdots, u_n\}$ be the universe of clients and $\mathcal{L}$ be the universe of descriptive attributes in the system. Let $G_j \subset \mathcal{U}$ be s set of clients that hold the attribute λ_j $(j = 1, 2, \cdots, q)$, which is referred to as an attribute group. G_j will be used as a client access list to λ_j. Let $\mathcal{G} = \{G_1, G_2, \cdots, G_q\}$ be the universe of attribute groups and GK_{λ_j} be the attribute group key that is shared among the non-revoked clients in $G_j \in \mathcal{G}$.

In a KEK tree [7], each node holds a KEK_j. A set of KEKs on the path node from leaf to root are called the path keys. A KEK tree is constructed by the data service manager as follows:

1) Each client u_{id} $(id = 1, 2, \cdots, n)$ in the universe $\mathcal{U}$ is assigned to a leaf node of the tree. Random keys are generated and assigned to all leaf nodes and internal nodes.

2) Each client $u_{id} \in \mathcal{U}$ obtains the path keys PAK_{id} from its leaf node to the root node of tree, securely. For example, the client u_4 has the path keys $PAK_4 = \{KEK_{11}, KEK_5, KEK_2, KEK_1\}$ in Figure 1.

3) The minimum cover sets [11] $node(G_j)$ is a minimum set of nodes in the tree, which can cover all of the leaf nodes associated with clients in G_j. $KEK(G_j)$ is a set of KEK values owned by $node(G_j)$. To consider the intersection of PAK_{id} and $KEK(G_j)$, we have $KEK = KEK(G_j) \cap PAK_{id}$.

Let us give an example to illustrate the attribute groups G_j. Suppose $\{u_1, u_2, u_3\}$ are associated with $\{\lambda_1, \lambda_2\}, \{\lambda_1, \lambda_2, \lambda_3\}, \{\lambda_2, \lambda_3\}$, respectively. We have the attribute group $G_1 = \{u_1, u_2\}, G_2 = \{u_1, u_2, u_3\}, G_3 = \{u_2, u_3\}$.

Consider the example in Figure 1. If the attribute group for attribute λ_j is $G_j = \{u_2, u_3, u_5, u_6, u_7, u_8\}$ and u_6 is associated with leaf node v_{13}, we compute the minimum cover sets $node(G_j) = \{v_9, v_{10}, v_3\}$ and get $KEK(G_j) = \{KEK_9, KEK_{10}, KEK_3\}$, which will be used to encrypt the attribute group key GK_{λ_j} in the data re-encryption phase. Since u_6 stores path keys $PAK_6 = \{KEK_{13}, KEK_6, KEK_3, KEK_1\}$, we have $KEK = KEK(G_j) \cap PAK_6 = \{KEK_3\}$, then u_6 can decrypt the header message to get the attribute group key GK_{λ_j} using KEK_3.

2.7 System Architecture

As shown in Figure 2, a verifiable attribute-based keyword search encryption scheme with attribute revocation (VABKS-AR) system consists of five entities: Trusted Authority (TA), Cloud Service Provider (CSP), Data Owner/Patient, Client/Doctor, and Third Party Audit (TPA).

- **Trusted Authority (TA)**: TA generates the public parameter, the master secret key and the clients'

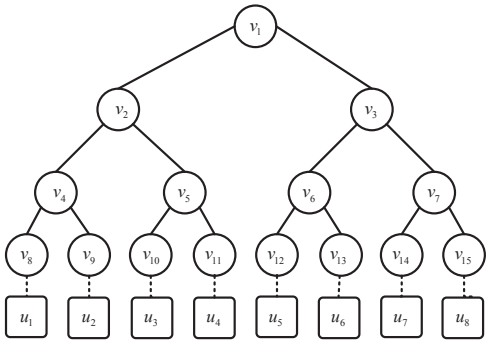


Figure 1: KEK tree

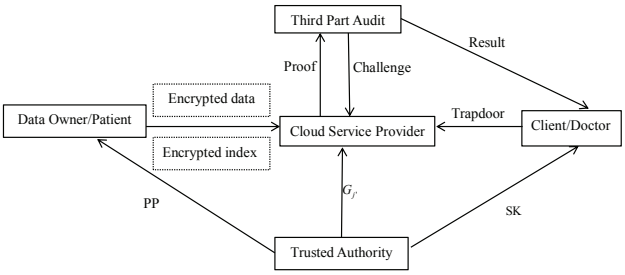


Figure 2: System architecture of VABKS-AR

secret key according to their attributes. TA is fully trusted in the system.

- **Cloud Service Provider (CSP)**: CSP consists of a data server and a data service manager. The data server has huge storage space and a strong computational power. The data service manager is in charge of managing the attribute group keys of each attribute group and providing the corresponding services. We assume the data service manager is honest-but-curious. i.e., it will honestly performs the operation but try to acquire much more information about the sensitive data.

- **Data Owner/Patient**: A patient is viewed as the data owner, which encrypts the sensitive data (i.e. electronic health record system data) and the keyword, and then uploads them to CSP in the form of ciphertext. Meanwhile, the data owner enforces the access policy for encrypted data, where the ciphertext will be shared with the client whose attributes satisfy the access structure embedded in ciphertext.

- **Client/Doctor**: A doctor is viewed as a client, which submits a search query to retrieve the encrypted EHR stored on the cloud server. Upon receiving the query, the cloud server searches the intended ciphertext by the use of trapdoor. If a client is not revoked and her or his attribute set satisfies the access policy, the client can decrypt the ciphertext.

- **Third Party Audit (TPA)**: TPA can provide the verification of search result and response a challenge to CSP. Upon receiving the challenge, CSP returns a proof to TPA. Finally, TPA calculates a value to verify the integrity of returned search results.

3 Formal Definition and Security Model

3.1 Formal Definition

In this section, the formal definition of a verifiable attribute-based keyword search encryption with attribute revocation is described as the following nine algorithms:

- **Setup**$(1^\lambda) \rightarrow (PP, MSK)$: TA takes a security parameter λ as input, and outputs the public parameter PP and a master secret key MSK.

- **KeyGen**$(MSK, id, S) \rightarrow SK$: TA runs this algorithm, which takes the master secret key MSK, the identifier id of a legal client u_{id} and attribute set $S \subseteq \mathcal{L}$ as input. This algorithm outputs a secret key SK to the client u_{id}.

- **Encrypt**$(PP, (M, \rho), \mathcal{K}, W) \rightarrow (CT, I_W)$: The data owner runs this algorithm, which takes the public parameter PP, an access policy (M, ρ), the symmetric key $\mathcal{K}$, and a set of keyword W as input. Using key encapsulation technology, this algorithm outputs a ciphertext CT and an encrypted index set I_W.

- **Re-encrypt**$(CT, G, RL) \rightarrow (CT', \hat{C})$: This algorithm is performed by the data service manager. Taking the ciphertext CT, attributes group $G \subseteq \mathcal{G}$ and a revocation list RL as input. This algorithm outputs a re-encrypted ciphertext CT' and a header message $\hat{C}$.

- **Trapdoor**$(SK, w) \rightarrow tk$: The client runs this algorithm, which takes a secret key SK and a keyword w as input. This algorithm outputs tk to CSP.

- **Search**$(PP, tk, I_W) \rightarrow (C', ID')$: CSP runs this algorithm, which takes the public parameter PP, a search token tk and an encrypted index set I_W as input. This algorithm outputs intended encrypted file set C' and corresponding identifier ID' to TPA if the search token tk matches with the index set I_W; otherwise, outputs $\perp$.

- **Verify**$(PK, C', ID') \rightarrow (0, 1)$: TPA runs this algorithm, which takes the data owner's PK, the returned encrypted file set C' and corresponding identifier set ID' as input. This algorithm outputs 1 if passes the result verification; otherwise, outputs 0.

- **Decrypt**$(CT', SK) \rightarrow \mathcal{K}$: The client runs this algorithm, which takes the ciphertext CT' and a secret key SK as input. This algorithm outputs the symmetric key $\mathcal{K}$.

- **CTUpdate**$(CT', RL') \rightarrow (CT'', \hat{C}')$: The data service manager runs this algorithm, which takes the re-encrypted ciphertext CT' and a new revocation list RL' as input. This algorithm outputs an updated ciphertext CT'' and a new header message $\hat{C}'$.

3.2 Security Model

In this section, we will give two security models: indistinguishability against selective ciphertext-policy and chosen plaintext attack (IND-sCP-CPA) game and indistinguishability against chosen keyword attack (IND-CKA) game. The security of our scheme is based on the following two games:

Firstly, according to Waters' scheme [22], we describe the **IND-sCP-CPA game** as follows:

- **Init.** The adversary $\mathcal{A}$ gives the challenge access policy (M^*, ρ^*) and a revocation list RL^*, where M^* has $n^* \leq q$ columns.
- **Setup.** The challenger $\mathcal{B}$ runs the *Setup* algorithm, sends the public parameter PP to $\mathcal{A}$, and then keeps the master secret key MSK for himself.
- **Phase 1.** The adversary $\mathcal{A}$ issues polynomial time secret key queries for (id, S). The challenger $\mathcal{B}$ sends SK to the adversary A, but with the restriction that:
  1) if $u_{id} \notin RL^*$, $S' = S$, and the set of attribute S' does not satisfy the challenge access policy (M^*, ρ^*).
  2) if $u_{id} \in RL^*$, then $S' = S \setminus \{\lambda_{j^*}\}$, and the set of attribute S' does not satisfy the challenge access policy (M^*, ρ^*).
- **Challenge.** The adversary $\mathcal{A}$ selects two equal length message k_0 and k_1 to the challenger $\mathcal{B}$. Then $\mathcal{B}$ randomly selects one bit $b \in \{0, 1\}$ and encrypts k_b under (M^*, ρ^*) and the revocation list RL^*. Finally, $\mathcal{B}$ sends the challenge ciphertext CT^* to $\mathcal{A}$.
- **Phase 2.** Same as Phase 1.
- **Guess.** The adversary $\mathcal{A}$ outputs its guess b' of b and wins the game if $b' = b$.

The advantage of the adversary $\mathcal{A}$ is defined as follows:

$$Adv_{\mathcal{A}}^{IND-sCP-CPA} = \left| \Pr[b' - b] - \frac{1}{2} \right|.$$

Definition 1. *A verifiable attribute-based keyword search encryption scheme with attribute revocation is IND-sCP-CPA secure if all polynomial time adversaries have at most a negligible advantage in the above game.*

Secondly, according to Boneh's scheme [1], we define the **IND-CKA game** as follows:

- **Setup.** The challenger $\mathcal{B}$ runs the Setup algorithm, sends the public parameter PP to $\mathcal{A}$, and then keeps the master secret key MSK for himself.
- **Phase 1.** The adversary $\mathcal{A}$ can adaptively query the challenger $\mathcal{B}$ for the trapdoor T_w of any keyword $w \in \{0, 1\}^*$ in polynomial time.
- **Challenge.** The adversary $\mathcal{A}$ sends two equal length keywords w_0 and w_1 to the challenger $\mathcal{B}$. The only restriction is that w_0 and w_1 have not been queried for the trapdoor. The challenger $\mathcal{B}$ randomly selects one bit $b \in \{0, 1\}$, generates index I_{w_b} for keyword w_b, and submits the challenge index I_{w_b} to the adversary $\mathcal{A}$.
- **Phase 2.** The adversary $\mathcal{A}$ can issue more trapdoor queries for keyword w with the restriction $w \neq w_0, w_1$.
- **Guess.** The adversary $\mathcal{A}$ outputs its guess b' of b and wins the game if $b' = b$.

The advantage of the adversary $\mathcal{A}$ is defined as follows:

$$Adv_{\mathcal{A}}^{IND-CKA} = \left| \Pr[b' - b] - \frac{1}{2} \right|.$$

Definition 2. *A verifiable attribute-based keyword search encryption scheme with attribute revocation is IND-CKA secure if all polynomial time adversaries have at most a negligible advantage in the above game.*

4 Concrete Construction

The concrete construction is described as follows:

- **Setup**(1^λ): This algorithm selects a bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, such that $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order p, an λ-bit prime, and $E(\cdot)$ be a probabilistic symmetric encryption algorithm. We define three hash functions $H : \mathbb{Z}_p \rightarrow \mathbb{G}, H_1 : \{0, 1\}^* \rightarrow \mathbb{G}, H_2 : \mathbb{G}_T \rightarrow \{0, 1\}^{\log_2 p}$. Let CL be a client list and RL be a revocation list, where CL and RL are initially empty. TA runs this algorithm as follows:
  1) Pick random $a, \alpha \in \mathbb{Z}_p$ and compute
  $$h = g^a, Y = \hat{e}(g, g)^\alpha.$$
  2) Publish the public parameter
  $$PP = \left(\hat{e}, h, Y, H, H_1, H_2, E(\cdot), CL, RL\right)$$
  and keep the master secret key $MSK = \alpha$ himself.
- **KeyGen**(MSK, id, S): A client sends its identifier id and a set of attributes $S \subseteq \mathcal{L}$ to TA. TA runs this algorithm as follows:
  1) Select $t \in \mathbb{Z}_p$ randomly and compute a secret key
  $$SK = \left(K = g^\alpha g^{at}, L = g^t, \{K_j = H(\lambda_j)^t\}_{\lambda_j \in S}\right)$$
  for the client $u_{id} \in \mathcal{U}$.

2) Add (id, g^{at}) to the client list CL and send SK to the client.

- **Encrypt**$(PP, (M,\rho), \mathcal{K}, \mathcal{F}, W)$: The data owner inputs the public parameter PP, an access policy (M,ρ), a symmetric key $\mathcal{K}$, a set of data file $\mathcal{F}$ and a set of keyword W. This algorithm is run by the data owner as follows:

1) Encrypt the data file $\mathcal{F} = (f_1, f_2, \cdots f_d)$ as $c_k = E_{\mathcal{K}}(f_k)$ $(1 \le k \le d)$ with the symmetric key $\mathcal{K}$.

2) Select random $s, y_2, \cdots, y_n \in \mathbb{Z}_p$ and set a column vector $\vec{v} = (s, y_2, \cdots, y_n)$. For $1 \le i \le \ell$, compute $\mu_i = M_i \cdot \vec{v}$, where M_i is the i-th row of M. Choose random numbers $r_1, \cdots, r_\ell \in \mathbb{Z}_p$ and calculate

$$CT = \{C_{0,0} = \mathcal{K} \cdot \hat{e}(g,g)^{\alpha s}, C_{0,1} = g^s, C_{0,2} = h^s, \\ \forall i = 1, \cdots, \ell : C_i = g^{a\mu_i} H(\rho(i))^{-r_i}, D_i = g^{r_i}\}.$$

3) Extract a set of keywords

$$W = (w_1, w_2, \cdots, w_m)$$

from the data files $\mathcal{F}$. For each keyword w_δ $(1 \le \delta \le m)$, compute

$$\varphi_\delta = \hat{e}(g,g)^{\alpha s} \cdot \hat{e}(g, H_1(w_\delta))^s$$

and

$$I_W = \{I_{w_\delta} = H_2(\varphi_\delta)\}_{\delta=1}^m,$$

where I_W is the encrypted index set for the keyword set W.

4) Select a random $x \in \mathbb{Z}_p$, and compute $PK = g^x$ as its public key. For each encrypted data file c_k with identifier k, calculate a signature $\sigma_k = (H_1(k) g^{c_k})^x$ with the data owner's secret key x.

After the construction of CT, the data owner sends $(CT, I_W, \{c_k, \sigma_k\}_{k=1}^d)$ to CSP.

- **Re-encrypt**(CT, G, RL): This algorithm inputs a ciphertext CT, a set of attribute group $G \subseteq \mathcal{G}$ and a revocation list RL. The data service manager runs this algorithm as follows:

1) For $\forall\, G_j \in G$, choose a random attribute group key $GK_{\lambda_j} \in \mathbb{Z}_p^*$, and re-encrypt CT as:

$$CT' = \{C'_{0,0} = C_{0,0}, C'_{0,1} = C_{0,1}, C'_{0,2} = C_{0,2}, \\ \forall\, i = 1, \cdots, \ell : C'_i = C_i, \\ RL = \emptyset : D'_i = D_i, \\ RL \neq \emptyset : D'_i = D_i^{GK_{\lambda_j}}\}.$$

2) Compute the minimum cover sets $node(G_j)$ of G_j in the KEK tree, get the corresponding $KEK(G_j)$, and generate a ciphertext $\hat{C} = \{E_\kappa(GK_{\lambda_j})\}_{\kappa \in KEK(G_j)}$, which called the header message.

- **Trapdoor**(SK, w): A client with identifier id and attribute set S inputs a secret key SK and a keyword w, The algorithm runs as follows:

1) The client selects $u \in \mathbb{Z}_p$ randomly, computes $q_u = g^{1/u}$, and sends (id, q_u) to TA. Then, TA retrieves g^{at} according to id in the client list CL, computes $q_{id} = g^{at} q_u^\alpha$, and sends q_{id} to the client.

2) The client calculates a search token $tk = \big(T_w = H_1(w) q_{id}^u, L' = L^u, \{K'_j = K_j^u\}_{\lambda_j \in S}\big)$ and sends tk to the data service manager.

- **Search**(PP, tk, I_W): This algorithm inputs the public parameter PP, a search token tk, and encrypted index set I_W. CSP runs this algorithm as follows:

1) Compute

$$l_w = \frac{\hat{e}(C'_{0,1}, T_w)}{\hat{e}(L', C'_{0,2})} = \hat{e}(g,g)^{\alpha s} \cdot \hat{e}\big(g, H_1(w)\big)^s.$$

2) If there exists some encrypted index I_{w_δ} such that $H_2(l_w) = I_{w_\delta}$, send $(CT', \hat{C})$, the relevant encrypted file set $C' = \{c_1, c_2, \cdots, c_\tau\}$ and the corresponding identifier set $ID' = \{1, 2 \cdots, \tau\}$ to TPA, where τ is the number of returned files; otherwise, return $\perp$.

- **Verify**(PK, C', ID'): This algorithm inputs the data owner's PK, the returned encrypted file set C' and corresponding identifier set ID'. TPA runs this algorithm as the following steps:

1) TPA randomly selects $v_r \in \mathbb{Z}_p$, and generates a $chal = < r, v_r > (r \in [1, \tau])$ to CSP.

2) Upon receiving the $chal$ of TPA, CSP computes $\zeta = \Sigma_{r \in [1,\tau]} v_r c_r$ and $\sigma = \Pi_{r \in [1,\tau]} {\sigma_r}^{v_r}$, where $\sigma_r = (H_1(r) g^{c_r})^x$. Then CSP sends (ζ, σ) to TPA.

3) TPA verifies whether the following equation holds or not. If hold, return 1 and send $(CT', \hat{C}, C', ID')$ to the client; otherwise, return 0.

$$\hat{e}(\sigma, g) = \hat{e}\big(g^\zeta \cdot \Pi_{r \in [1,\tau]} H_1(r)^{v_r}, PK\big).$$

- **Decrypt**(CT', SK): This algorithm inputs the ciphertext CT', a secret key SK, and runs as follows:

1) If a client has a valid attribute λ_j, i.e. $u_{id} \in G_j$, he can use a $KEK \in \big(KEK(G_j) \cap PAK_{id}\big)$ to get the attribute group key GK_{λ_j}. And then u_{id} updates its secret key with the attribute group keys as follows:

$$SK = \big(K = g^\alpha g^{at}, L = g^t, \\ \{K_j = H(\lambda_j)^{t/GK_{\lambda_j}}\}_{\lambda_j \in S}\big).$$

2) Output $(0, \perp)$ if S does not satisfy (M, ρ).

3) Otherwise, let $I \subset \{1,2,\cdots,\ell\}$ be defined as $I = \{i : \rho(i) \in S\}$ and $\{\omega_i \in \mathbb{Z}_p | i \in I\}$ be a set of constants such that if μ_i are valid shares of any secret s according to M, then $\Sigma\omega_i\mu_i = s$. We have

$$Q_{CT} = \prod_{i\in I}\Big(\hat{e}(C'_i, L)\cdot\hat{e}(D'_i, K'_j)\Big)^{\omega_i} = \hat{e}(g,g)^{ast}.$$

4) Decrypt the ciphertext and obtain the symmetric key: $\mathcal{K} = C'_{0,0}\cdot\frac{Q_{CT}}{\hat{e}(C'_{0,1},K)}$.

5) Decrypt the encrypted data files C' using $\mathcal{K}$.

- **CTUpdate**(CT', RL'): This algorithm inputs CT' and a new revocation list RL'. If an attribute $\lambda_{j'}$ of the client is revoked, TA sends the updated membership list $G_{j'}$ to CSP. The data service manager runs this algorithm as follows:

1) Select random $s', y'_2, \cdots, y'_n \in \mathbb{Z}_p^n$, a new attribute group key $GK'_{\lambda_{j'}}$, and set a column vector $\vec{v'} = (s', y'_2, \cdots, y'_n) \in \mathbb{Z}_p^n$. For $1 \le i \le \ell$, compute $\mu'_i = M_i \cdot \vec{v'}$, where M_i is the i-th row of M.

2) Choose random numbers $r'_1, \cdots, r'_\ell \in \mathbb{Z}_p$ and update the ciphertext CT' as:

$$\begin{aligned} CT'' = \big(&C''_{0,0} = C'_{0,0}\cdot\hat{e}(g,g)^{\alpha s'},\\ &C''_{0,1} = C'_{0,1}\cdot g^{s'}, C''_{0,2} = C'_{0,2}\cdot h^{s'},\\ &\forall\, i = 1,\cdots,\ell : C''_i = C'_i\cdot g^{a\mu'_i}H(\rho(i))^{-r'_i},\\ &\rho(i)\in RL' : D''_i = D'_i\cdot(g^{r'_i})^{GK'_{\lambda_{j'}}},\\ &\rho(i)\notin RL' : D''_i = D'_i\cdot(g^{r'_i})^{GK_{\lambda_j}}\big). \end{aligned}$$

3) Compute a new minimum cover set and generate a new header message with updated $KEK(G_{j'})$ as follows:

$$\begin{aligned} \hat{C'} =&(\{E_\kappa(GK'_{\lambda_{j'}})\}_{\kappa\in KEK(G_{j'})},\\ &\forall_{\lambda_j\in S\setminus\{\lambda_{j'}\}} : \{E_\kappa(GK_{\lambda_j})\}_{\kappa\in KEK(G_j)}). \end{aligned}$$

Correctness. The proposed scheme is correct as the following equations hold:

$$\begin{aligned} l_w &= \frac{\hat{e}\big(C'_{0,1}, T_w\big)}{\hat{e}(L', C'_{0,2})} = \frac{\hat{e}\big(g^s, H_1(w)g^{atu}g^{\alpha}\big)}{\hat{e}\big((g^t)^u, g^{as}\big)}\\ &= \frac{\hat{e}(g,g)^{\alpha s}\cdot\hat{e}\big(g, H_1(w)\big)^s\cdot\hat{e}(g,g)^{astu}}{\hat{e}(g,g)^{astu}}\\ &= \hat{e}(g,g)^{\alpha s}\cdot\hat{e}\big(g, H_1(w)\big)^s \end{aligned}$$

$$\begin{aligned} Q_{CT} &= \prod_{i\in I}\Big(\hat{e}(C'_i, L)\cdot\hat{e}(D'_i, K_j)\Big)^{\omega_i}\\ &= \prod_{i\in I}\Big(\hat{e}\big(g^{a\mu_i}H\big(\rho(i)\big)^{-r_i}, g^t\big)\\ &\qquad\cdot\hat{e}\big((g^{r_i})^{GK_{\lambda_j}}, H\big(\rho(i)\big)^{t/GK_{\lambda_j}}\big)\Big)^{\omega_i}\\ &= \hat{e}(g,g)^{\Sigma a\mu_i\omega_i t} = \hat{e}(g,g)^{ast}\\ \mathcal{K} &= C'_{0,0}\cdot\frac{Q_{CT}}{\hat{e}(C'_{0,1},K)}\\ &= \mathcal{K}\cdot\hat{e}(g,g)^{\alpha s}\frac{\hat{e}(g,g)^{ast}}{\hat{e}(g^s, g^{\alpha}g^{at})}\\ &= \mathcal{K}\cdot\hat{e}(g,g)^{\alpha s}\frac{\hat{e}(g,g)^{ast}}{\hat{e}(g^s,g^{\alpha})\cdot\hat{e}(g,g)^{ast}}\\ &= \mathcal{K}\cdot\hat{e}(g,g)^{\alpha s}\frac{1}{\hat{e}(g,g)^{\alpha s}} = \mathcal{K}\\ \hat{e}(\sigma,g) &= \hat{e}\big(\Pi_{r\in[1,\tau]}\sigma_r^{v_r}, g\big)\\ &= \hat{e}\Big(\Pi_{r\in[1,\tau]}\big(H_1(r)g^{c_r}\big)^{xv_r}, g\Big)\\ &= \hat{e}\Big(\Pi_{r\in[1,\tau]}\big(H_1(r)^{v_r}\cdot g^{\Sigma v_r c_r}\big)^x, g\Big)\\ &= \hat{e}\Big(\Pi_{r\in[1,\tau]}H_1(r)^{v_r}\cdot g^{\zeta}, g^x\Big)\\ &= \hat{e}\Big(g^{\zeta}\cdot\Pi_{r\in[1,\tau]}H_1(r)^{v_r}, PK\Big) \end{aligned}$$

5 Security and Performance

5.1 Security Analysis

Theorem 1. *If a probabilistic polynomial-time adversary $\mathcal{A}$ wins the IND-sCP-CPA game with non-negligible advantage ε, then we can construct a simulator $\mathcal{B}$ to solve the q-parallel BDHE problem with non-negligible advantage $\varepsilon' = \varepsilon/2$.*

Proof. Suppose $\mathcal{A}$ is an adversary that has advantage ε in breaking the IND-sCP-CPA game. We construct a simulator $\mathcal{B}$ that can solve the q-parallel BDHE problem with probability at least ε'.

- **Init.** The simulator $\mathcal{B}$ is given a decisional q-parallel challenge vector $\vec{y}$ and a random number T. The adversary $\mathcal{A}$ selects the challenge access policy (M^*, ρ^*) and the revocation list RL^*, where M^* has $n^* \le q$ columns.

- **Setup.** The simulator $\mathcal{B}$ randomly selects $\alpha' \in \mathbb{Z}_p$, computes $\hat{e}(g,g)^{\alpha} = \hat{e}(g^a, g^{a^q})\cdot\hat{e}(g,g)^{\alpha'}$, which implies that $\alpha = \alpha' + a^{q+1}$. We use a list called H-list to run the random oracle H for $\mathcal{B}$. The simulator $\mathcal{B}$ responds as follows:

1) If $H(j)$ has already appeared on the H-list, then $\mathcal{B}$ returns the value that was predefined before.

2) Otherwise, let X be the set of indices i that makes $\rho^*(i) = \lambda_{j^*}$ true. $\mathcal{B}$ randomly selects a number $z_{j^*} \in \mathbb{Z}_p$, and executes the random oracle:

– If $X = \emptyset$, $H(j^*) = g^{z_{j^*}}$;
– If $X \neq \emptyset$, we have

$$H(j^*) = g^{z_{j^*}} \prod_{i \in X} g^{aM^*_{i,1}/b_i} \cdot g^{a^2 M^*_{i,2}/b_i} \cdots g^{a^{n^*} M^*_{i,n^*}/b_i} \ (n^* \leq q).$$

- **Phase 1.** The adversary $\mathcal{A}$ issues polynomial time secret key queries for (id, S). Suppose the adversary sends the identifier and the corresponding set of attributes (id, S) to the simulator $\mathcal{B}$, but the following restrictions must be satisfied:

  1) If $u_{id} \notin RL^*$, $S' = S$, and the attributes set S' does not satisfy (M^*, ρ^*).
  2) If $u_{id} \in RL^*$, then $S' = S \setminus \{\lambda_{j^*}\}$, and the attributes set S' does not satisfy (M^*, ρ^*).

  The simulator $\mathcal{B}$ chooses a vector $\overrightarrow{\omega} = (\omega_1, \omega_2, \cdots, \omega_{n^*}) \in \mathbb{Z}_p^{n^*}$. For any i, such that $\rho^*(i) \in S'$ and $\omega_1 = -1$, we have $M_i^* \cdot \overrightarrow{\omega} = 0$. $\mathcal{B}$ randomly selects a number $r \in \mathbb{Z}_p$, and computes a secret key as follows:

$$K = g^{\alpha} g^{at} = g^{\alpha'} g^{ar} \prod_{i=2,\cdots,n^*} (g^{a^{q+2-i}})^{\omega_i},$$
$$L = g^t = g^r \cdot \prod_{1,\cdots,n^*} (g^{a^{q+1-i}})^{\omega_i},$$

  which implies

$$t = r + \omega_1 a^q + \omega_2 a^{q-1} + \cdots + \omega_{n^*} a^{q-n^*+1}.$$

  For $\forall\ \lambda_{j^*} \in S'$, when there is no i such that $\rho^*(i) = \lambda_{j^*}$, we let $K_j = L^{z_{j^*}}$. While for those attributes $\lambda_{j^*} \in S'$ that satisfy the access structure, $\mathcal{B}$ can not simulate the items $g^{a^{(q+1)/b_i}}$. However, we have $M_i^* \cdot \overrightarrow{\omega} = 0$. Therefore, all of these terms of $g^{a^{(q+1)/b_i}}$ can be canceled. Let X be the set of indices i such that $\rho^*(i) = \lambda_{j^*}$. The simulator $\mathcal{B}$ computes K_{j^*} as follows:

$$K_{j^*} = L^{z_{j^*}} \prod_{i \in X} \prod_{j=1,\cdots,n^*} \Big((g^{(a^j/b_i)r}) \cdot \prod_{k=1,\cdots,n^*, k \neq j} (g^{a^{q+1+j-k}/b_i})^{\omega_k} \Big)^{M^*_{i,j}}.$$

- **Challenge.** The adversary $\mathcal{A}$ chooses two equal length challenge message k_0 and k_1 to the simulator $\mathcal{B}$. $\mathcal{B}$ randomly selects a number $s \in \mathbb{Z}_p$ and a random bit $\gamma \in \{0, 1\}$. Then it computes as follows:

$$C^*_{0,0} = k_\gamma \cdot T \cdot e(g^s, g^{\alpha'}), C^*_{0,1} = g^s, C^*_{0,2} = h^s.$$

  It is difficult for $\mathcal{B}$ to simulate C_i^* since it contains $g^{a^j s}$ that $\mathcal{B}$ can not simulate. However, $\mathcal{B}$ randomly selects $y'_2, \cdots, y'_{n^*} \in \mathbb{Z}_p$ and $r'_1, \cdots, r'_\ell \in \mathbb{Z}_p$. Then simulator $\mathcal{B}$ shares the secret s utilizing the vector $\vec{v} = (s, sa + y'_2, sa^2 + y'_3, \cdots, sa^{n^*-1} + y'^*_n) \in \mathbb{Z}_p^{n^*}$. For $i = 1, \cdots, \ell$, we define R_i as the set of all $k \neq i$ such that $\rho^*(i) = \rho^*(k)$. The challenge ciphertext C_i^* is set as:

$$C_i^* = H(\rho^*(i))^{r'_i} \Big(\prod_{j=2,\cdots,n^*} (g^a)^{-M^*_{i,j} y'_j} \Big) (g^{s \cdot b_i})^{-z_{\rho^*(i)}} \cdot \Big(\prod_{k \in R_i} \prod_{j=1,\cdots,n^*} (g^{a^j \cdot s \cdot b_i / b_k}) \Big)^{-M^*_{k,j}}.$$

  1) For the non-revoked attribute $\rho^*(i)$, a challenge ciphertext is set as $D_i^* = g^{-r'_i} g^{-sb_i}$.
  2) For the revoked attribute $\rho^*(i) = \lambda_{j^*}$ and $j^* \neq i$, by selecting a random number $GK'_{\lambda_{j^*}}$, $\mathcal{B}$ computes the challenge ciphertext $D_i^* = (g^{-r'_i} g^{-sb_i})^{GK'_{\lambda_{j^*}}}$.

  $\mathcal{B}$ gives the challenge ciphertext

$$CT^* = (C^*_{0,0}, C^*_{0,1}, C^*_{0,2}, \{C_i^*, D_i^*\}_{i=1,\cdots,\ell})$$

  to $\mathcal{A}$.

- **Phase 2.** Same as Phase 1.

- **Guess.** The adversary $\mathcal{A}$ outputs γ' of γ. $\mathcal{B}$ returns $\mu = 0$ and responds $T = e(g, g)^{a^{q+1} \cdot s}$ if $\gamma' = \gamma$; otherwise, $\mathcal{B}$ returns $\mu = 1$ and responds $T \in \mathbb{G}_T$ as a random element.

If $\mu = 0, \mathcal{A}$ obtains a valid ciphertext of k_γ. The advantage of $\mathcal{A}$ in this situation is ε, therefore $\Pr[\gamma' = \gamma | \mu = 0] = 1/2 + \varepsilon$. Since $\mathcal{B}$ guesses $\mu' = 0$ when $\gamma' = \gamma$, we have $\Pr[\mu' = \mu | \mu = 0] = 1/2 + \varepsilon$.

If $\mu = 1$, we have $\Pr[\gamma' \neq \gamma | \mu = 1] = 1/2$. As $\mathcal{B}$ guesses $\mu' = 1$ when $\gamma' \neq \gamma$, we have $\Pr[\mu' = \mu | \mu = 1] = 1/2$.

The advantage of $\mathcal{B}$ to solve the decisional q-parallel BDHE problem is $\varepsilon' = \varepsilon/2$ as follows.

$$\begin{aligned} &\Pr[\mu' = \mu] - \frac{1}{2} \\ =&\frac{1}{2} \Pr[\mu' = \mu | \mu = 0] + \frac{1}{2} \Pr[\mu' = \mu | \mu = 1] - \frac{1}{2} \\ =&\frac{1}{2}(\frac{1}{2} + \varepsilon) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} \\ =&\frac{\varepsilon}{2}. \end{aligned}$$

□

Theorem 2. *If a probabilistic polynomial-time adversary $\mathcal{A}$ wins the IND–CKA game with non-negligible advantage ϵ, then we can construct a simulator $\mathcal{B}$ to solve the BDH problem with non-negiligible advantage $\epsilon' = \epsilon/(e \cdot q_T \cdot q_{H_2})$ where e is the base of the nature logarithm.*

Proof. Suppose $\mathcal{A}$ is an attack algorithm that has advantage ϵ in breaking the IND–CKA game. We construct an algorithm $\mathcal{B}$ that solve the BDH problem with probability at least ϵ'.

Suppose that $\mathcal{A}$ makes at most q_{H_2} hash function queries to H_2 and at most q_T trapdoor queries. The algorithm $\mathcal{B}$ is given $g, u_1 = g^{\alpha}, u_2 = g^{\beta}, u_3 = g^{\gamma} \in \mathbb{G}$. It aims at outputing $\hat{e}(g,g)^{\alpha\beta\gamma} \in \mathbb{G}_T$.

- **Setup.** The algorithm $\mathcal{B}$ starts by giving $\mathcal{A}$ the public parameters PP. $\mathcal{B}$ simulates the challenger and interacts with $\mathcal{A}$ as follows:

- **Phase 1.** The adversary $\mathcal{A}$ can query the following random oracle at any time.

  $\mathcal{O}_{H_1}(w_\eta)$: The algorithm $\mathcal{B}$ creates a list of tuple $< w_\eta, h_\eta, a_\eta, c_\eta >$ called the H_1-list. The list is initially empty. When $\mathcal{A}$ asks the random oracle H_1 at the point of $w_\eta \in \{0,1\}^*$, $\mathcal{B}$ responds as follows:

  1) If the query w_η appears on the H_1-list in a tuple $< w_\eta, h_\eta, a_\eta, c_\eta >$, then $\mathcal{B}$ responds with $H_1(w_\eta) = h_\eta$.
  2) Otherwise, $\mathcal{B}$ selects a random $c_\eta \in \{0,1\}^*$ so that $\Pr[c_\eta = 0] = 1/(q_T + 1)$.
  3) $\mathcal{B}$ picks a random $a_\eta \in \mathbb{Z}_p$. If $c_\eta = 0$, $\mathcal{B}$ computes $h_\eta = u_2 \cdot g^{a_\eta}$; otherwise, $\mathcal{B}$ computes $h_\eta = g^{a_\eta}$. The algorithm $\mathcal{B}$ adds the tuple $< w_\eta, h_\eta, a_\eta, c_\eta >$ to the H_1-list and responds to $\mathcal{A}$ with $H_1(w_\eta) = h_\eta$.

  $\mathcal{O}_{H_2}(\varphi_\eta)$: The H_2-list is initially empty. At any time the adversary $\mathcal{A}$ can issue a query to H_2. The algorithm $\mathcal{B}$ responds as follows:

  1) If the query on φ_η exists in the H_2-list, $\mathcal{B}$ responds I_{w_η} to $\mathcal{A}$.
  2) Otherwise, $\mathcal{B}$ picks a new random value $I_{w_\eta} \in \{0,1\}^{\log p}$ for each new φ_η and sets $H_2(\varphi_\eta) = I_{w_\eta}$. The algorithm $\mathcal{B}$ adds the pair $(\varphi_\eta, I_{w_\eta})$ to the H_2-list and sends I_{w_η} to $\mathcal{A}$.

  $\mathcal{O}_{q_{id}}(id)$: The algorithm B creates a list of tuple $< SK, q_{id}, C >$ called the table T. Upon receiving a query of secret key on $\mathcal{A}$ and a commitment value C. $\mathcal{B}$ checks whether the tuple appears on T.

  1) If so, $\mathcal{B}$ returns q_{id} to $\mathcal{A}$.
  2) Otherwise, $\mathcal{B}$ sets $q_{id} = g^{\alpha/u} \cdot g^{at}$ and sends it to $\mathcal{A}$.

  $\mathcal{O}_{tk}(id, w_\eta)$: The adversary $\mathcal{A}$ issues a query for the trapdoor corresponding to the keyword w_η and the client identifier id, and then $\mathcal{B}$ responds as follows:

  1) $\mathcal{B}$ runs the above H_1-queries to obtain $h_\eta \in \mathbb{G}$ such that $H_1(w_\eta) = h_\eta$. Let $< w_\eta, h_\eta, a_\eta, c_\eta >$ be the corresponding tuple on the H_1-list. If $c_\eta = 0$, then $\mathcal{B}$ responds failure and aborts the game;
  2) Otherwise, we know $c_\eta = 1$ and $h_\eta = g^{a_\eta}$. $\mathcal{B}$ selects u from $\mathbb{Z}_p$ randomly, searches the table T for SK, and sets $tk = (g^{a_\eta} \cdot g^{\alpha}(g^{at})^u = g^{a_\eta} q_{id}^u, L' = L^u, \{K'_j = K_j^u\}_{j \in S})$. Therefore, $tk = (T_w, L', K'_j)$ is a valid search token. $\mathcal{B}$ returns tk to $\mathcal{A}$.

- **Challenge.** The adversary $\mathcal{A}$ sends two equal-length keywords w_0 and w_1 to $\mathcal{B}$. The algorithm $\mathcal{B}$ generates a challenge index as follows:

  1) $\mathcal{B}$ runs H_1-queries twice to obtain $h_0, h_1 \in \mathbb{G}$ such that $H_1(w_0) = h_0$ and $H_1(w_1) = h_1$. For $\eta = \{0,1\}$, let $< w_\eta, h_\eta, a_\eta, c_\eta >$ be the corresponding tuples on the H_1-list. If both $c_0 = 1$ and $c_1 = 1$, then $\mathcal{B}$ reports failure and terminates.
  2) We know that at least one of c_0, c_1 is equal to 0. $\mathcal{B}$ randomly picks $b \in \{0,1\}$ such that $c_b = 0$.
  3) The algorithm $\mathcal{B}$ selects s from $\mathbb{Z}_p$ randomly, and sets $I = (C_{0,1} = g^s, C_{0,2} = h^s)$. Let $\varphi_b = \hat{e}(u_1, u_2)^{\gamma} \cdot \hat{e}(g, u_2 g^{a_b})^{\gamma}$. $\mathcal{B}$ runs the above H_2-queries algorithm to obtain $J \in \{0,1\}^{\log^p}$. $\mathcal{B}$ stores the tuple $< \varphi_b, J >$ in the H_2-list and responds to $\mathcal{A}$ with the challenge $I_{w_b} = J$ for a random $J \in \{0,1\}^{log^p}$. Let $\gamma = s$, we have

  $$H_2(\hat{e}(u_1, u_2)^{\gamma} \cdot \hat{e}(g, H_1(w_b))^{\gamma}) = J,$$

  i.e $J = H_2(\hat{e}(u_1, u_2)^{\gamma} \cdot \hat{e}(g, H_1(w_b))^{\gamma}) = H_2(\hat{e}(u_1, u_2)^{\gamma} \cdot \hat{e}(g, u_2 g^{a_b})^{\gamma})$.

- **Phase 2.** Same as Phase 1. $\mathcal{A}$ can continue to issue the trapdoor queries for keywords w_η, where the only restriction is that $w_\eta \neq w_0, w_1$. $\mathcal{B}$ responds to these queries as before.

- **Guess.** The adversary $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ of b. $\mathcal{A}$ computes φ_b as follows:

$$\begin{aligned}
\varphi_b &= \frac{\hat{e}(C_{0,1}, T_w)}{\hat{e}(L', C_{0,2})} \\
&= \frac{\hat{e}\big(g^s, H_1(w) q_{id}\big)}{(L^u, h^s)} \\
&= \frac{\hat{e}(g^s, g^{a_\eta} g^{atu} g^{\alpha})}{\hat{e}\big((g^t)^u, g^{as}\big)} \\
&= \frac{\hat{e}(g,g)^{\alpha s} \cdot \hat{e}(g, g^{a_\eta})^s \cdot \hat{e}(g,g)^{astu}}{\hat{e}(g,g)^{astu}} \\
&= \hat{e}(g,g)^{\alpha s} \cdot \hat{e}(g, g^{a_\eta})^s.
\end{aligned}$$

If $\mathcal{A}$ can break our scheme, we have $\varphi_b = \hat{e}(u_1, u_2)^s \cdot \hat{e}(g, g^{a_b})^s$. $\mathcal{B}$ searches I_{w_b} from the H_2-list for φ_b and outputs $\varphi_b / \hat{e}(K_1, u_2 g^{a_b}) = \hat{e}(g,g)^{\alpha\beta\gamma}$. The adversary $\mathcal{A}$ must have issued a query for either $H_2(\hat{e}(u_1, u_2)^{\gamma} \cdot \hat{e}(g, H_1(w_0))^{\gamma})$ or $H_2(\hat{e}(u_1, u_2)^{\gamma} \cdot \hat{e}(g, H_1(w_1))^{\gamma})$. Therefore, with the probability $1/2$ the H_2-list contains a pair whose left hand side is $\varphi_\eta = \hat{e}(u_1, u_2)^{\gamma} \cdot \hat{e}(g, H_1(w_b))^{\gamma}$. If $\mathcal{B}$ picks this pair (φ_η, I) from the H_2-list, then $\varphi_\eta / \hat{e}(K_1, u_2 g^{a_b}) = \hat{e}(g,g)^{\alpha\beta\gamma}$ as required.

We will analyze that $\mathcal{B}$ correctly outputs $\hat{e}(g,g)^{\alpha\beta\gamma}$ with probability at least ϵ'. The probability that $\mathcal{B}$ does

not abort during the simulation phase is at least $1/e$, and the probability that $\mathcal{B}$ does not abort during the challenge phase is at least $1/q_T$. Therefore, $\mathcal{B}$ does not abort with the probability at least $1/eq_T$. In a real attack game $\mathcal{A}$ issues a query for $\varphi_\eta = \hat{e}(u_1, u_2)^\gamma \cdot \hat{e}(g, H_1(w_b))^\gamma$ with probability at least ϵ. The adversary $\mathcal{A}$ issues an H_2 query for either $H_2(\hat{e}(u_1, u_2)^\gamma \cdot \hat{e}(g, H_1(w_0))^\gamma)$ or $H_2(\hat{e}(u_1, u_2)^\gamma \cdot \hat{e}(g, H_1(w_1))^\gamma)$ with probability at least 2ϵ. The detailed analysis of above results is shown in Boneh *et al.*'s scheme [1]. $\mathcal{B}$ will choose the correct pair with probability at least $1/q_{H_2}$. Assuming $\mathcal{B}$ does not abort during the simulation, it will produce the correct answer with probability ϵ/q_{H_2}. Since $\mathcal{B}$ does not abort with the probability at least $1/eq_T$, the probability of $\mathcal{B}$ successfully outputs $\hat{e}(g, g)^{\alpha\beta\gamma}$ with probability $\epsilon/(e \cdot q_T \cdot q_{H_2})$. □

Table 1: Notations

Symbols	Description
P	the pairing operation
E	the group exponentiation in $\mathbb{G}$
E_T	the group exponentiation in $\mathbb{G}_T$
n	the number of attributes in the system
$n_{a,u}$	the number of attributes a client possesses
k	the number of attributes embedded in a ciphertext

5.2 Performance Analysis

In this section, we will give the performance analysis from the perspective of functional comparison, computation cost, and experiment result. In Table 1, we define some notations which will be used in this section.

Functionality comparisons: In Table 2, we give the comprehensive comparisons according to some important features, including expressive, attribute revocation, keyword search and the verifiability. From Table 2, Hur *et al.*'s scheme [7] can achieve fine-grained attribute revocation, but not support data retrieval and result verification. Zheng *et al.*'s scheme [27] can provide verifiability and fine-grained keyword search, but there are huge computational overhead in the verification process. Sun *et al.*'s scheme [19] can achieve data retrieval, the verifiability, and revocation, but the verification progress is low and only support system-level client revocation. Wang *et al.*'s scheme [21] can achieve attribute revocation and keyword search, but the verifiability of search results is not considered. In general, compared with the above schemes, our scheme has better functionality.

Computation cost: In Table 3, since we have the same functionality as Sun *et al.*'s scheme [19], we briefly compare our computational costs with Sun *et al.*'s. As the operation cost over $\mathbb{Z}_p$ is much less than group and pairing operation, we ignore the computational time over $\mathbb{Z}_p$. From Table 3, In *Setup* algorithm, Sun *et al.*'s scheme needs $3n$ exponentiations in $\mathbb{G}$, one exponentiation in $\mathbb{G}_T$, and one pairing operation, while our scheme only requires one exponentiations in $\mathbb{G}$, one exponentiation in $\mathbb{G}_T$, and one pairing operation. In *Keygen* algorithm, our scheme needs $(3+n_{a,u})$ exponentiations in $\mathbb{G}$, but Sun *et al.*'s scheme needs $(2n+1)$ exponentiations in $\mathbb{G}$ and two exponentiations in $\mathbb{G}_T$. In *Encrypt* algorithm, our scheme needs $(3k+2)$ exponentiations in $\mathbb{G}$, three exponentiations in $\mathbb{G}_T$ and one pairing operation, but Sun *et al.*'s scheme needs $(n+1)$ exponentiations in $\mathbb{G}$, one exponentiation in $\mathbb{G}_T$ and one pairing operation. The time cost of our scheme is a little larger than Sun *et al.*'s scheme. In *Trapdoor* algorithm, our scheme needs $(k+4)$ exponentiations in $\mathbb{G}$, However, Sun *et al.*'s scheme needs $(2n+1)$ exponentiations in $\mathbb{G}$, which is larger than our scheme. In *Search* algorithm, Sun *et al.*'s scheme requires one exponentiation and $(n+1)$ pairing operations, but our scheme only needs two exponentiations in $\mathbb{G}_T$ and two pairing operations.

Experiment result: We conduct our experiments using Java Pairing-Based Cryptography (JPBC) library [2]. We implement the proposed scheme on a Windows machine with Intel Core 2 processor running at 3.30 GHz and 4.00 G memory. The running environment of our scheme is Java Runtime Environment 1.7, and the Java Virtual Machine(JVM) used to compile our programming is 64 bit which brings into correspondence with our operation system.

In our experiments, we compare the proposed scheme with Sun *et al.*'s scheme [19] and Wang *et al.*'s scheme [21] in the search time. The modulus of the elements in the group is chosen to be 512 bits, the number of attributes ranges from 10 to 50.

From Figure 3, we know that the search time cost grows linearly with the number of attributes in Sun *et al.*'s scheme, and the search time cost of Wang *et al.*'s scheme is less than Sun *et al.*'s scheme. However the search time cost of the proposed scheme is the most efficient than the other two schemes. For example, when the number of attributes is 50, the search time consumption of the proposed scheme only needs 0.03s, while Sun *et al.*'s scheme needs about 0.8s and Wang *et al.*'s scheme needs 0.05s. Therefore, compared with the above two schemes, the proposed scheme is more efficient and practical.

6 Conclusions

In this paper, we have proposed a verifiable attribute-based keyword search encryption with attribute revocation for electronic health record system. By using a KEK tree and re-encryption techniques, the proposed scheme can achieve efficient revocation and assure that the updated ciphertext cannot be decrypted by the revoked

Table 2: The comparisons of the functionality

Schemes	Expressive	Revocation	Keyword Search	Verifiability
Hur et al.*'s scheme [7]*	access tree	√	×	×
Sun et al.*'s scheme [19]*	AND gate	√	√	√
Wang et al.*'s scheme [21]*	LSSS	√	√	×
Zheng et al.*'s scheme [27]*	access tree	×	√	√
Ours	LSSS	√	√	√

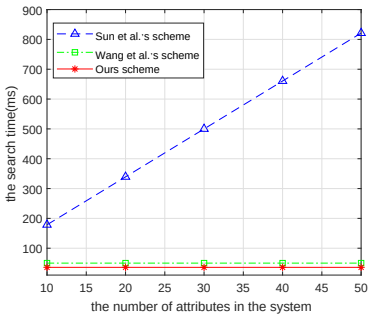


Figure 3: The time cost of EHR system

clients. In addition, we introduce TPA to verify the integrity of the returned search results, which can reduce the client's computation overhead. Furthermore, performance analysis shows that our scheme is efficient and practical for electronic health record system. Since the policy may contain some sensitive information of patient. The proposed scheme do not support policy hiding. For the future work, we intend to propose a privacy-preserving attribute-based keyword search encryption scheme.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants No. 61807026, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant No. 2019JM-198, the Plan For Scientific Innovation Talent of Henan Province under Grant No. 184100510012, and in part by the Program for Science and Technology Innovation Talents in the Universities of Henan Province under Grant No. 18HASTIT022.

Table 3: The comparisons of computation cost

Operations	Sun *et al.*'s scheme [19]	Ours
Setup	$3nE + E_T + P$	$E_T + E + P$
KeyGen	$(2n+1)E + 2E_T$	$(3 + n_{a,u})E$
Encrypt	$(n+1)E + E_T + P$	$(3k+2)E + 3E_T + P$
Trapdoor	$(2n+1)E$	$(4+k)E$
Search	$(n+1)P + E_T$	$2P + 2E_T$

References

[1] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 506–522, May 2004.

[2] A. D. Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of The 16th IEEE Symposium on Computers and Communications (ISCC'11)*, pp. 850–855, 2011.

[3] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments," *International Journal of Network Security*, vol. 16, no. 1, pp. 1–13, 2014.

[4] M. L. Florence and D. Suresh, "Enhanced secure sharing of PHRs in cloud using attribute-based encryption and signature with keyword search," *Advances in Big Data and Cloud Computing*, vol. 645, pp. 375–384, 2018.

[5] R. Gandikota, "A secure cloud framework to share EHRs using modified CP-ABE and the attribute bloom filter," *Education and Information Technologies*, vol. 23, no. 5, pp. 2213–2233, 2018.

[6] M. T. Hu, H. Gao, and T. G. Gao, "Secure and efficient ranked keyword search over outsourced cloud data by chaos based arithmetic coding and confusion," *International Journal of Network Security*, vol. 21, no. 1, pp. 105–114, 2019.

[7] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[8] M. Joshi, K. Joshi, and T. Finin, "Attribute based encryption for secure access to cloud based EHR systems," in *Proceedings of IEEE 11th Intenational Conference on Cloud Computing*, pp. 932–935, July 2018.

[9] M. Li, S. C. Yu, Y. Zheng, K. Ren, and W. J. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

[10] Y. Miao, J. Ma, X. Liu, and F. Wei Z. Liu, L. Shen, "VMKDO: Verifiable multi-keyword search over encrypted cloud data for dynamic data-owner," *Peer-to-Peer Networking and Applications*, vol. 11, no. 2, pp. 287–297, 2018.

[11] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proceedings of The 21st Annual International Cryptology Conference*, pp. 41–62, Aug. 2001.

[12] S. Narayan, M. Gagne, and R. S. Naini, "Privacy preserving HER system using attribute-based infrastructure," in *Proceedings of The ACM Workshop on Cloud Computing Security Workshop*, pp. 47–52, Oct. 2010.

[13] Y. S. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Generation Computer System*, vol. 67, pp. 133–151, 2017.

[14] B. E. Reedy and G. Ramu, "A secure framework for ensuring EHR's integrity using fine-grained auditing and CP-ABE," in *Proceedings of The IEEE 2nd International conference on Big Data Security*, pp. 85–89, Apr. 2016.

[15] S. Rezaei, M. A. Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115–131, 2018.

[16] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of The 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, May 2005.

[17] D. Sethia, H. Saran, and D. Gupta, "CP-ABE for selective access with scalable revocation: A case study for mobile-based healthfolder," *International Journal of Network Security*, vol. 20, no. 4, pp. 689–701, 2018.

[18] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 44–55, May 2000.

[19] W. Sun, S. Yu, W. Lou, Y. Hou, and H. Li, "Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, 2016.

[20] S. F. Tzeng, C. C. Lee, and M. S. Hwang, "A batch verification for multiple proxy signature," *Parallel Processing Letters*, vol. 21, no. 1, pp. 77–84, 2011.

[21] S. P. Wang, D. Zhang, Y. Zhang, and L. Liu, "Efficiently revocable and searchable attribute-based encryption scheme for mobile cloud storage," *IEEE Access*, vol. 6, pp. 30444–30457, 2018.

[22] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proceedings of The 14th International Conference on Practice and Theory in Public Key Cryptography (PKC'11)*, pp. 53–70, Mar. 2011.

[23] A. Wu, D. Zheng, Y. L. Zhang, and M. L. Yang, "Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing," *Sensors*, vol. 18, no. 7, pp. 1–17, 2018.

[24] F. Xhafa, J. F. Wang, X. F. Chen, J. K. Liu, J. Li, P. Krause, and D. S. Wong, "An efficient PHR service system supporting fuzzy keyword search and fine-grained access control," *Soft Computing*, vol. 18, no. 9, pp. 1795–1802, 2014.

[25] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of The 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS'10)*, pp. 261–270, Apr. 2010.

[26] Y. Zhao, M. Ren, S. Jiang, G. Zhu, and H. Xiong, "An efficient and revocable storage CP-ABE scheme in the cloud computing," *Computing*, pp. 1–25, 2018.

[27] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: Verifiable attribute-based keyword search over outsourced encrypted data," in *Proceedings of IEEE Conference on Computer Communications*, pp. 522–530, Apr. 2014.

Biography

Zhenhua Liu received the B. S. degree from Henan Normal University, and the master's and Ph.D. degrees from Xidian University, China, in 2000, 2003, and 2009, respectively. He is currently a Professor with Xidian University. His research interests include cryptography and information security.

Yan Liu received the B. S. degree from Shenyang Agricultural University in 2017. She is currently pursuing the master's degree in mathematics with Xidian University. Her research interests include cryptography and cloud security.

Jing Xu received the B. S. degree from Henan Normal University in 2017. She is currently pursuing the master's degree in mathematics with Xidian University. Her research interests include cryptography and cloud security.

Baocang Wang received the B. S. degree in Computational Mathematics and Their Application Softwares from Xidian University, China, in 2001, the M. S. degree in Cryptology from Xidian University, China, in 2004, and the Ph.D. degree in Cryptology from Xidian University, China, in 2006. He is currently a Professor and Ph. D. supervisor with Xidian University. His research interests include postquantum cryptography, fully homomorphic cryptography, number theoretic algorithms, and cloud security.

An Unlinkable Key Update Scheme Based on Bloom Filters for Random Key Pre-distribution

Bin Wang
(Corresponding author: Bin Wang)
Information Engineering College, Yangzhou University
No. 196 West HuaYang Road, Yangzhou City, Jiangsu Province, P. R. China, 225127
(Email: jxbin76@yeah.net)
(Received Apr. 16, 2019; Revised and Accepted Dec. 10, 2019; First Online Jan. 29, 2020)

Abstract

Eschenauer *et al.* presented an efficient random key pre-distribution scheme for WSNs that assigns symmetric keys to sensor nodes by randomly sampling from a large key pool. Most research in this line assume nodes exchange key identifiers to determine common keys between them. However, an adversary can learn topology information of the underlying random key graph by intercepting exchanged key identifiers. In addition, when key exposure occurs, compromised nodes should be revoked and uncompromised nodes' key rings should be updated securely. In this paper, we design an unlinkable key update mechanism that can revoke compromised nodes while an adversary is infeasible to link key identifiers with a node. A key update node is responsible for distributing a random seed among uncompromised nodes in order to update their key rings securely. The revoked keys are represented by a bloom filter to avoid exchange of key identifiers when checking whether a node is compromised. As a bloom filter has zero false negative rate, we utilize negative answers returned by a bloom filter to identify uncompromised keys and nodes with high probability. Then a local broadcast mechanism is used to speed up update of uncompromised nodes' key ring securely.

Keywords: Bloom Filter; Key Pre-distribution; Unlinkablity

1 Introduction

Wireless sensor networks (WSNs) are networks consisting of battery-powered sensor nodes that are able to perform sensing tasks, data processing and multi-hop wireless communication. With the rapid development in sensor technologies and wireless communication, WSNs have been widely used in applications such as environment monitoring, target tracking, military operations and attracted a lot of attention from research communities [15]. As sensor nodes may be deployed in hostile environments, they must forward data packets to a base station (a sink node) in a secure manner to prevent an adversary from breaking data privacy or mounting a forgery attack [1, 13]. Hence security is an important issue to be addressed for wide deployment of WSNs. To provide security services (*e.g.*, data encryption or identity authentication) for WSNs, it is necessary to establish shared keys between nodes via appropriate key management mechanisms. However, as sensor nodes are resource-constrained equipments with limited storage and computational capability, traditional public key cryptographic schemes(*e.g.*, Diffie-Hellman key agreement protocol [5]) are not applicable for WSNs since computational cost of public key operations are too costly to be implemented for sensor nodes [4]. That is, energy efficiency is an important factor to be considered when handling security challenges for WSNs [8].

As senor nodes can only afford light-weight operations such as hash operations, symmetric encryption/decryption operations, Eschenauer and Gligor [6] suggested a random key pre-distribution scheme for WSNs in which each sensor node is equipped with a fixed-sized key ring comprising symmetric keys randomly sampled from a large key pool before network deployment. Afterwards, two nodes can compute a session key for secure communication if their key rings share at least a common key. Connectivity of the induced random key graph is proven to hold asymptotically under certain choices of system parameters [17]. Several improvements or extensions to this kind of random key pre-distribution schemes are suggested such as q-composite random key pre-distribution scheme [3], random pairing key pre-distribution scheme [16]. On the other hand, deterministic key pre-distribution schemes based on combinatorial designs [14] are also presented as alternatives to key pre-distribution schemes for WSNs. Deterministic key pre-distribution schemes have the advantage that secure connectivity property can be proven to hold in a deterministic way.

A subtle point inherent in random key pre-distribution schemes for WSNs is how to determine common keys be-

tween a pair of nodes. That is, a kind of key confirmation mechanism is a pre-requisite in order to find common keys between nodes. Currently, it is generally assumed that nodes can exchange their own key identifiers directly as a solution for key confirmation. However, the potential security risk of this simple solution is that an adversary may link observed key identifiers with nodes after observing communication between nodes. These information can help an adversary to deduce topology structure of the underlying key graph.

To counteract the above mentioned security risk of the simple solution for key confirmation, Marek Klonowski and Piotr Syga [9] presented a novel unlinkable key confirmation solution based on bloom filters to determine common keys between a pair of nodes. Each node should compute a local bloom filter as a compact representation for secret keys hold by itself. Then two nodes can exchange their bloom filters other than key identifiers. A node can check whether a key hold by itself is also an element of the other node's key ring by issuing set membership queries to the other node's bloom filter. As positive answers returned by bloom filters may be faulty with non-zero probability (false positive rate), their key confirmation process should be repeated with fresh randomness for several times between a pair of nodes to ensure that positive answers from bloom filters can be considered as correct with high probability. The additional communication overhead will also consume a large amount of nodes' energy.

As networks structure of WSNs may vary due to factors such as node failure or malfunctioning, single phase key pre-distribution is not able to adapt to dynamic changes in WSNs.To support a flexbile secure infrastructure for new nodes deployments,Albert Levi, and Salim Sarimurat [10] suggested use of multiple generation of dynamic key pools. As nodes should evolve their key rings by iterative hash computations by the end of each generation, it is implicit that their method requires all nodes to refresh their key rings synchronously. On the other hand, when WSNs are deployed in hostile environments, some compromised keys should be revoked since they may have been broken by an adversary. A node is compromised if its key ring is a subset of the compromised keys controlled by an adversary. The scheme in [10] is unable to revoke compromised nodes from WSNs. Moreover, an adversary is still able to intercept key identifiers exchanged between nodes during a specific generation. Generally speaking, revoking nodes from WSNs is more difficult than addition of new nodes.

As a result, how to efficiently exclude compromised keys and nodes from WSNs and evolve uncompromised nodes' key ring in an unlinkable way is also an issue to be addressed for random key pre-distribution. In this paper, we suggest an unlinkable key update mechanism based on bloom filters to ensure that uncompromised nodes can refresh their key rings securely while an adversary is infeasible to link key identifiers with a node.

Given a set of keys to be revoked, a key update node in our solution uses a bloom filter to represent the set of revoked keys and is responsible for evolving key pool and uncompromised nodes' key rings as well as excluding compromised nodes. Then the key update node runs several rounds of unlinkbale key confirmation process based on this bloom filter to determine whether a node is compromised or not. Recall that a node is compromised if its key ring is a subset of the revoked keys. If we use positive answers from the bloom filter to identify compromised keys, efficiency issues due to non-zero false-positive rate of bloom filters will also be encountered. As a bloom filter has zero false negative rate, we suggest that negative answers from the bloom filter can be used as indications of uncompromised keys to identify uncompromised nodes. Analysis shows that an uncompromised node can determine its unrevoked keys shared with the key update node with high probability in specified parameters setting.

Then a random seed that can be used to refresh uncompromised nodes' key rings should be generated and distributed among uncompromised nodes securely.The key update node broadcasts the random seed encrypted under a shared unrevoked key to a selected uncompromised node and its neighboring nodes in the formed key graph. Finally, uncompromised nodes can apply hash operations with the received random seed to update their key rings. Simulation results shows that this local broadcast mechanism help speed up propagation of the random seed. Section 2 decribes concept of bloom filters and CPA security of symmetric encryption. Section 3 decribes a key update scheme for revoking compromised nodes in WSNs and defines unlinkability notion for this kind of key update schemes. The presented key update scheme is proven to be unlinkable when the underlying symmetric encryption scheme is assumed to be CPA secure. Section 4 concludes this paper.

2 Preliminaries

2.1 Bloom Filter

The concept of bloom filter is presented by [2], which is a compact data structure used for answering set membership queries. Given l hash functions $h_1(\cdot), \cdots, h_l(\cdot)$ with range $[1, m]$,a bloom filter $BF_{m,l}$ is a bit vector with length m. To represent a set S consisting of n elements by $BF_{m,l}$, compute l entries $h_1(s), \cdots, h_l(s)$ for each element $s \in S$ and set $BF_{m,l}[h_1(s)] = 1, \cdots, BF_{m,l}[h_l(s)] = 1$. Given a membership query x, $BF_{m,l}$ returns a positive answer $BF_{m,l}(x) = 1$ to indicate that $x \in S$ if $BF_{m,l}[h_1(x)] == 1 \bigcap \cdots \bigcap BF_{m,l}[h_l(x)] == 1$ is true; Otherwise it returns a negative answer $BF_{m,l}(x) = 0$ to indicate that $x \notin S$. It is well known that a bloom filter will probably return false positive answers for membership queries but its false negative rate is always zero.

2.2 Semantic Security

Given a symmetric encryption scheme $\Pi = (KG, E, D)$,where E is encryption function and D denotes decryption function, define an experiment CPA^A_Π,where A is a probabilistic polynomial time (PPT) adversary:

1) Challenger S generates a secret key $k = KG(\cdot)$;
2) A is given access to an encryption oracle $O_k(\cdot)$ that outputs a ciphertext $c = E_k(m)$ when taking as input a plaintext m.
3) A outputs two distinct equal-length plaintexts m_0 m_1.
4) S picks a random bit $b \leftarrow \{0,1\}$ and provides A with $c^* = Enc_k(m_b)$.
5) A outputs a random bit b'.

A symmetric encryption scheme is CPA secure if $|Pr[b' = b] - \frac{1}{2}|$ is negligible for any PPT adversary A in the experiment CPA^A_Π.

3 An Unlinkable Key Update Scheme for WSNs

3.1 System Setup

Assume there are n sensor nodes $N_j, 1 \le j \le n$, distributed in a geographic region. Let ID be the set of key identifiers, K^i be the key pool with constant size P at the start of the i^{th} round,$1 \le i$. Each node N_j holds a key ring $R^i_j \subseteq K^i$ with fixed size r at the start of i^{th} round.$M^i : K^i \to ID$ is a one-to-one mapping from the key pool of the i^{th} round to the set of key identifiers. Initially, the key ring R^1_j hold by N_j contains symmetric keys randomly sampled from a large key pool K^1 at the start of first round. Afterwards, key pool K^{i+1} will be derived from key pool K^i by executing the key update process described in subsection 3.2.

In case of key exposure, some compromised keys should be revoked and uncompromised nodes' key rings should be evolved to exclude compromised nodes. Our key update process is divided into several rounds. A key update node V constructs a set RK^i of revoked keys with size q at the start of the i^{th} round. For the sake of simplicity, we assume the full visible communication assumption as in [9] that assumes each pair of nodes can communicate with each other directly. This assumption help speed up propagation of a secure random seed.

The key update node V maintains a table S_j that records key identifiers associated with each node N_j. V also picks a random secret seed $seed_i$ at the start of i^{th} round and must ensure the following hold by the end of i^{th} round:

1) Key pool K^i will be replaced by K^{i+1} by the end of i^{th} round as follows: Each key $k^i \in K^i$ is replaced by a key $k^{i+1} = G(k^i||seed_i) \in K^{i+1}$, where $G(\cdot)$ is a secure hash function.
2) A node N_j is uncompromised at the start of i^{th} round if the set $R^i_j \backslash RK^i$ is not empty. In other words, an uncompromised node must hold at least one unrevoked key in its current key ring. Key ring R^i_j of an uncompromised node N_j at the start of i^{th} round will be replaced by R^{i+1}_j by the end of the i^{th} round as follows: Each key $k^i_j \in R^i_j$ is replaced by a key $k^{i+1}_j = G(k^i_j||seed_i) \in R^{i+1}_j$.
3) A PPT adversary that has knowledge of the revoked keys RK^i at the start of i^{th} round should have not enough knowledge to link key identifiers of the key set $R^i_j \bigcap \{K^i \backslash RK^i\}$ with the corresponding uncompromised node N_j by the end of i^{th} round by intercepting communications between nodes.

Define an experiment $Link^{n,M}_\Pi$ as a notion for unlinkbility, where M is a PPT adversary and Π is a symmetric encryption scheme.

1) Challenger C generates a key pool by running the key generation algorithm of Π and selects n node identifiers $id_1, \cdots, id_n$;
2) M is allowed to choose $h \le n-2$ compromised nodes from $\{id_1, \cdots, id_n\}$ and keep their corresponding key rings. The compromised nodes' identifiers is kept in RID.
3) M is given access to an oracle $O_C(\cdot)$ that outputs communication transcript between a node id and a key update node when taking as input a node identity id.
4) C outputs two distinct uncompromised nodes' identifiers mid_0 mid_1 from $\{id_1, \cdots, id_n\} \backslash RID$.
5) C picks a random bit $b \leftarrow \{0,1\}$ and provides A with communication transcript between the chosen node mid_b and a key update node.
6) M outputs a random bit b'.

A key update scheme is unlinkable if $|Pr[b' = b] - \frac{1}{2}|$ is negligible for any PPT adversary M in the experiment $Link^{n,M}_\Pi$.

3.2 One Round of Key Update Process

The key update node V first generates a bloom filter $RBF^i_{n_b,l}$ with n_b bits at the start of i^{th} round by choosing l hash functions $h_0(\cdot), \cdots, h_{l-1}(\cdot)$,and initializes all entries of $RBF^i_{n_b,l}$ to zero.V executes the following Algorithm 1 to construct $RBF^i_{n_b,l}$ associated with the revoked key set RK^i with size q.

We use notation $RBF^i_{n_b,l}(k)$ to denote an answer return by the bloom filter $RBF^i_{n_b,l}$ for a membership query

Algorithm 1 Construction of RBF

```
Begin
for each k ∈ RK^i do
    for j = 0 to l − 1 do
        RBF^i_{n_b,l}[h_j(k)] = 1;
    end for
end for
End
```

k. $RBF^i_{n_b,l}(k) = 1$ is a positive answer to indicate that $k \in RK^i$. $RBF^i_{n_b,l}(k) = 0$ is a negative answer to indicate that $k \notin RK^i$. Define a set $FRK^i = \{k_x | k_x \in K^i \backslash RK^i \bigcap RBF^i_{n_b,l}(k) == 1\}$. That is,$FRK^i$ contains all keys that are not revoked but get positive answers from $RBF^i_{n_b,l}$.

Step 1. In the following, V broadcasts the bloom filter $RBF^i_{n_b,l}$ to all nodes. Having received $RBF^i_{n_b,l}$,a node executes Algorithm 2 to construct a set $USK^i_j \subseteq R^i_j \backslash RK^i$. Recall that R^i_j is the key ring of node N_j with size r at the start of i^{th} round.

Algorithm 2 Construction of unrevoked keys

```
Begin
USK^i_j = ∅
for k ∈ R^i_j do
    Initialize the answer RBF^i_{n_b,l}(k) = 1;
    for j = 0 to l − 1 do
        if RBF^i_{n_b,l}[h_j(k)] == 0 (a) then
            Set the answer RBF^i_{n_b,l}(k) = 0; and break;
        end if
    end for
    if RBF^i_{n_b,l}(k) == 0 (b) then
        USK^i_j = USK^i_j ⋃ {k};
    end if
end for
End
```

Claim 1. *We have* $USK^i_j = R^i_j \backslash \{RK^i \bigcup FRK^i\}$ *by the end of Algorithm 2.*

Proof. By construction of the bloom Filter $RBF^i_{n_b,l}$,if $k \in RK^i$, we have $RBF^i_{n_b,l}(k) == 1$ holds. In addition, the set FRK^i enumerates all keys $k \in K^i \backslash RK^i$ with false positive answer $RBF^i_{n_b,l}(k) == 1$.As a result, $RBF^i_{n_b,l}(k) == 0$ holds if and only if $k \in R^i_j \backslash \{RK^i \bigcup FRK^i\}$. As $R^i_j \subset K^i$,$R^i_j \backslash \{RK^i \bigcup FRK^i\}$ is a subset of $K^i \backslash \{RK^i \bigcup FRK^i\}$. When $R^i_j \backslash \{RK^i \bigcup FRK^i\}$ is not empty, $k \in R^i_j \backslash \{RK^i \bigcup FRK^i\}$ implies $RBF^i_{n_b,l}(k) == 0$ holds and we conclude that $k \in USK^i_j$ by condition (b) in Algorithm 2. □

As $R^i_j \backslash \{RK^i \bigcup FRK^i\}$ is a subset of $R^i_j \backslash RK^i$, it is possible that USK^i_j is empty for some uncompromised node. That is,some uncompromised node will be identified as compromised by algorithm 3.2. Let E_j denotes the event that USK^i_j is empty for some uncompromised node N_j. T_j is a random variable to count the number of revoked keys in $R^i_j \bigcap RK^i$. We compute the probability of E_j as follows:

$$Pr[E_j] = \sum_{0 \le i \le r-1} Pr[E_j | T_j = i] Pr[T_j = i]. \quad (1)$$

As $RBF^i_{n_b,l}$ is used to represent q revoked keys, the probability of a false positive event $RBF^i_{n_b,l}(k) == 1$ for some unrevoked key $k \in R^i_j \backslash \{RK^i\}$ is approximately $(1 - (1 - \frac{1}{n_b})^{l \cdot q})^l$ [12] if we assume the hash functions are modeled by independent random functions. Given $T_j = j$,E_j occurs if and only if $RBF^i_{n_b,l}(k) == 1$ occurs for each of $r - i$ unrevoked keys in $R^i_j \backslash RK^i$. By the independence assumption, we have:

$$Pr[E_j | T_j = i] \approx (1 - (1 - \frac{1}{n_b})^{l \cdot q})^{l \cdot (r-i)}. \quad (2)$$

As key rings assigned to nodes are assumed to be randomly sampled:

$$Pr[T_j = i] \approx \binom{r}{i} (\frac{q}{P})^i (1 - \frac{q}{P})^{r-i}. \quad (3)$$

When taking parameters $P = 1000$, $q = 100$, $l = 2$, $r = 50$, $n_b = 64$, we get $Pr[E_j] \approx 0.0197$ by numerical computation. On the other hand, the optimum false positive rate of bloom filter $RBF^i_{n_b,l}$ in this setting is $\approx 0.6185^{\frac{n_b}{q}} \approx 0.7353$ [12]. Choose larger values for parameters l and n_b can further reduce $Pr[E_j]$ at the cost of additional communication overhead.

Step 2. When USK^i_j is not empty,the corresponding uncompromised node N_j will send a response message to V, which contains node identifier id_j of N_j.

Step 3. Having received response messages from uncompromised nodes with non-empty set USK^i_j, V will randomly picks a node N_j among uncompromised nodes that have sent response messages. Then V transmits a random number r_V to the chosen node N_j.

Step 4. Having received r_V, N_j randomly picks a key $k^*_j \in USK^i_j$ and transmits ciphertext $c^*_j = E_{k^*_j}(id_j || r_V)$ to V, which is encrypted under the chosen symmetric key k^*_j.

Step 5. Having received c^*_j,V performs Algorithm 3 to construct a shared key SK^i_j with the uncompromised node N_j.

Remark: Condition (c) denotes extraction of the matching key k by key identifier kid; Condition (d) denotes decryption of the ciphertext c^*_j under the extracted matching key k.

By the correctness of decryption, we have $k^*_j = k$, where k is the extracted matching key.

Algorithm 3 Key Extraction

Begin
Set SK_j^i=NULL,$flag = 0$;
for $kid \in S_j$ **do**
  $k = (M^i)^{-1}(kid)$; (c)
  if $D_k(c_j^*) == id_j||r_V$ (d) **then**
    $flag = 1$,$SK_j^i = k$,break;
  end if
end for
End

Step 6. V picks a random *seed* and broadcasts ciphertext $c^* = E_{SK_j^i}(r_V + 1||seed)$ to the chosen node N_j and its neighboring nodes in the key graph.

Having received c^*, N_j decrypts c^* to recover $r_V + 1||seed = D_{SK_j^i}(c^*)$. Then N_j utilizes *seed* to update its key ring as follows:

Algorithm 4 Update Key Ring

Begin
for each $k^i \in R_j^i$ **do**
  $k^{i+1} = G(k^i||seed) \in R_j^{i+1}$
end for
End

In addition, each neighboring node of N_j in the random key graph can also try to decrypt the ciphertext c^* by iterating the keys shared with N_j. If their decryption operations are consistent with condition (d), these neighboring nodes of N_j can also use *seed* to update their own key rings.The rest of uncompromised nodes that do not get *seed* continue to interact with V by executing Steps 2-6 repeatedly until their key rings can be successfully updated by the end of *ith* round.

When taking parameters $P = 1000$, $q = 100$, $l = 2$, $r = 50$, $n_b = 64$, $n = 100$, our simulation results shows that Steps 2-6 should be looped by 35 times on average to finish one round of the presented key update process in this parameters setting.

Claim 2. *A PPT adversary that has knowledge of revoked keys in RK^i at the start of ith round is computationally infeasible to link key identifiers with any uncompromised node N_j by intercepting communications between nodes, if the underlying symmetric encryption scheme is CPA secure.*

Proof. Note that the bloom filter $RBF_{n_b,l}^i$ contains no information with respect to the unrevoked keys in $R_j^i \bigcap \{K^i \backslash RK^i\}$ associated with an uncompromised node N_j. The ciphertexts $c_j^* = E_{k_j^*}(id_j||r_V)$, $c^* = E_{k_j^*}(r_V + 1||seed)$ encrypted under the symmetric key k_j^* are the only sources that an adversary can gain information about unrevoked keys in $R_j^i \bigcap \{K^i \backslash RK^i\}$. Intuitively, by semantic security of symmetric encryption schemes [11],it is computationally infeasible for a PPT adversary to learn information of the plaintext and encryption key when he can only intercept ciphertexts. As a result, a PPT adversary is not able to link key identifiers with any uncompromised node N_j.

Assume there is an adversary M can break unlinkability of our key uodate scheme with non-negligible probability. We construct an adversary A against a symmetric encryption scheme $\Pi = (KG, E, D)$ in CPA_Π^A.

A simulates $Link_\Pi^{n,M}$ for M in one round as follows:

A generates $n > 1$ node identifiers $id_1, \cdots, id_n$;

When M chooses $h \leq n-2$ compromised nodes in the simulated $Link_\Pi^{n,M}$, we assume without loss of generality that they are $(id_1, \cdots, id_h)$.

A picks two distinct uncompromised identifiers mid_0 and mid_1 and we implicitly assume the corresponding two nodes share a common key k that is the challenge secret key used by the challenger in CPA_Π^A. Then A generates key rings for nodes in $\{id_1, \cdots, id_n\} \backslash \{mid_0, mid_1\}$. and M is provides with key rings of compromised nodes chosen by him.

Oracle $O_C(\cdot)$ in $Link_\Pi^{n,M}$ is simulated by A as follows:

Given a node identifier id as input, if $id \notin \{mid_0, mid_1\}$, A can simply simulate communication transcript between id and a key update node. Otherwise, A uses oracle access to $O_k(\cdot)$ in CPA_Π^A to generate ciphertexts for simulating communication transcript between $id \in \{mid_0, mid_1\}$ and a key update node. It is implicitly assumed that the challenge secret key k in CPA_Π^A is chosen as the shared key to generate communication transcript in the simulated $Link_\Pi^{n,M}$.

A outputs mid_0 and mid_1 in the simulated $Link_\Pi^{n,M}$.

A submits two distinct equal-length plaintexts $m_0 = mid_0||r_V$ $m_1 = mid_1||r_V$ to challenger S in experiment CPA_Π^A. S picks a random bit $b \leftarrow \{0,1\}$ and provides A with $c^* = Enc_k(mid_b||r_V)$.

A concatenates c^* with the rest of communication transcript between node mid_b and a key update node.that can be generated by access to $O_k(\cdot)$ in CPA_Π^A and provided M with the correctly generated full communication transcript.

When M outputs a random bit b' in the simulated $Link_\Pi^{n,M}$, A also outputs a random bit b' in CPA_Π^A.

By the above construction, A succeeds in CPA_Π^A with non-negligible probability by the assumption M can break unlinkability of our key uodate scheme with non-negligible probability. This contradicts the assumption that the underlying symmetric encryption scheme is CPA secure. Hence the presented key update scheme is unlinkable.

□

Furthermore, it is relatively straightforward to see that the ciphertexts $c_j^* = E_{k_j^*}(id_j||r_V)$, $c^* = E_{k_j^*}(r_V + 1||seed)$ also provides authentication functionality between an uncompromised node and a key update node to prevent an adversary mount a forgery attack.

4 Conclusions

Random key pre-distribution is an important security technique for WSNs. In this paper, we consider an unlinkable key update mechanism for WSNs to revoke a subset of compromised nodes and evolve uncompromised nodes' key rings. Our scheme uses a bloom filter as a compact representation for the set of revoked keys. The use of a bloom filter make it unnecessary to exchange key identifiers for nodes to identify revoked keys. As a bloom filter has zero false positive rate, we suggest using negative answers returned by the bloom filter to identify uncompromised keys. Our analysis shows that uncompromised nodes can recover unrevoked keys with high probability. Then a key update node broadcasts a random seed encrypted by a shared unrevoked key to a specified uncompromised node and its neighboring nodes in the key graph. Then these nodes can use the recovered random seed to update their local key rings. In the future, the key update mechanism may be considered in other communication interference model such as protocol interference model [7].

Acknowledgments

The research is supported by Natural Science Foundation in JiangSu province,P.R.China (Grant No. BK20170512) and Natural Science Foundation of Colleges and Universities in Jiangsu Province, P.R.China (Grant No. 17KJB413003), Natural Science Foundation in Yangzhou City, JiangSu province, P.R. China (Grant No. YZ2016113),Yangzhou University Science and Technology Innovation Foundation Project?(Grant No. 2016CXJ025).

References

[1] S. Akleylek, A. Karakaya, "A survey on security threats and authentication approaches in wireless sensor networks," in *International Symposium on Digital Forensic and Security (ISDFS'18)*, pp. 1–4, Mar. 2018.

[2] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[3] H. Chan, A. Perrig, D. Song, "Random key predistribution schemes for sensor networks," in *Symposium on Security and Privacy*, pp. 197–213, 2003.

[4] A. Diaz and P. Sanchez, "Simulation of attacks for security in wireless sensor network," *Sensors*, vol. 16, no. 11, p. 1932, 2016.

[5] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[6] V. Gligor, L. Eschenauer, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM conference on Computer and Communications Security (CCS'02)*, pp. 41–47, 2002.

[7] P. Gupta, P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[8] M. S. Hwang, C. T. Li, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks," *Information Sciences*, vol. 181, no. 23, p. 5333-5347, 2011.

[9] M. Klonowski and P. Syga, "Enhancing privacy for ad hoc systems with predeployment key distribution," *Ad Hoc Networks*, vol. 59, pp. 35–47, 2017.

[10] A. Levi and S. Sarimurat, "Utilizing hash graphs for key distribution for mobile and replaceable interconnected sensors in the IoT context," *Ad Hoc Networks*, vol. 57, pp. 3–18, 2017.

[11] Y. Lindell, J. Katz, *Introduction to Modern Cryptography*, 2014. (`https://repo.zenk-security.com/Cryptographie\%20.\%20Algorithmes\%20.\%20Steganographie/Introduction\%20to\%20Modern\%20Cryptography.pdf`)

[12] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probalistic Analysis*, 2005. ISBN 13: 978-0521835404.

[13] J. Newsome, E. Shi, D. Song, A. Perrig, "The sybil attack in sensor networks: Analysis & defenses," in *The Third International Symposium on Information Processing in Sensor Networks (IPSN'04)*, pp. 259–268, 2004.

[14] M. B. Paterson and D. R. Stinson, "A unified approach to combinatorial key predistribution schemes for sensor networks," *Design, Codes and Cryptography*, vol. 71, no. 3, pp. 433–457, 2014.

[15] K. D. Wong, Y. H. Hu, D. Li, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.

[16] O. Yagan and A. M. Makowski, "On the connectivity of sensor networks under random pairwise key predistribution," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 5754–5762, 2012.

[17] O. Yagan and A. M. Makowski, "Zero-one laws for connectivity in random key graphs," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 2983–2999, 2012.

Biography

Bin Wang biography. Bin Wang received his Ph. D. degree of communication and information system in Shanghai Jiaotong University, P. R. China. His research interests include cryptography and network security. Dr. Wang is now an associate professor of Department of Electronics and Communication Engineering, Information Engineering College, Yangzhou University, located in No.196 West HuaYang Road, Yangzhou city, Jiangsu province, P. R. China.

Survey on Attribute-based Encryption in Cloud Computing

P. R. Ancy, Addapalli V. N. Krishna, K. Balachandran, M. Balamurugan, and O. S. Gnana Prakasi
(Corresponding author: P. R. Ancy)

Department of Computer Science, Engineering, Christ University
Bengaluru, Karnataka, India
(Email: ancy.prasadam@res.christuniversity.in)

(Received Feb. 10, 2020; Revised and Accepted June 8, 2020; First Online July 26, 2020)

Abstract

Attribute-Based Encryption (ABE) is an appropriate solution to the access mechanism and security issues in cloud computing. As we know cloud computing is the emerging technique and solution for problems such as storage, security, efficiency, and many more facing today. As cloud computing arises as a new technology many issues and confusion are arising with it mainly regarding the security of the data. The cloud providers are capable of storing enough data, but the user is having doubt about how much security the cloud providers are giving to the data. Due to this reason, many organizations and companies are not willing to move to the cloud environment. So, Attribute-Based Encryption came as a solution to this problem. As per our comprehensive survey that has been done on different ABE schemes, we have included the recent ABE schemes, different access policy, problems, and their solutions for ABE in this paper. Finally, this paper also includes the security comparison and efficiency comparison of different ABE schemes.

Keywords: Access Control; Ciphertext-Policy Attribute-Based Encryption; Multiauthority; Revocation Mechanism

1 Introduction

Security is one of the main aspect of cloud computing. With the development of cloud computing, many data owners store their data in the cloud server for simplifying local IT management and reducing the cost [17]. Cloud storage is one of the major services provided by cloud computing [12]. It enables data owners to remotely host their data by outsourcing them to cloud servers, which brings great convenience for both individuals and enterprises to share data over the Internet [16, 24, 25]. As today every organization is moving to cloud for different services their main concern is on how much secure is their data in the cloud provider's environment. Therefore this is the main area where many types of research are going through. For enhancing security in cloud storage many methods are introduced and one among them is ABE (Attribute-Based Encryption) [4]. ABE is a type of public-key encryption in which the sender encrypts the message using receiver's public key and the receiver decrypt the message using the private key. The main key point of ABE is outsourcing which means the data is encrypted within the source or sender's environment and sends it to the service provider's environment for storing it.

In ABE encryption and decryption are done using a set of attributes and access policy. Attributes are the secret key of a user and the ciphertext is dependent upon attributes. Like traditional identity-based encryption, the sender in an ABE system only needs to know the receiver's description in order to determine their public key [2]. The decryption of the ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext.

There are two types of access structure monotonic access structure and non-monotonic access structure. Monotonic access structure support "AND " and "OR" between attributes and non-monotonic access structure support "NOT" with "AND" and "OR" between attributes. [10] proposed a method that can convert any monotone circuit to an equivalent access tree. There are two types of ABE, Key Policy Attribute-Based Encryption (KP-ABE) and Ciphertext Policy Attribute-Based Encryption (CP-ABE). In KP-ABE data is encrypted using attributes and decrypted using access policy and in CP-ABE data is encrypted using access policy and decrypted using attributes. About the number of authorities, a system has there are two types of system single authority system and multiauthority system. In the case of revocation, there are attribute revocation and user revocation. Attribute revocation is changing the user's attributes because of the expiring of attributes, revoke of attributes or need of adding new attributes. And in the case of user revocation, the system should satisfy both forward and backward security. In forward security, revoked

users should not access new ciphertext using an old secret key and in backward security newly joined users not give the privilege to access the previous ciphertext.

2 Related Work

A new scheme is suggested [3] that can withstand any number of corrupt authorities. The author applies this technique to attain a multi-authority form of identical access control Attribute-Based Encryption. According to this scheme each user has to prove his set of attributes to the third party to obtain a secret key. The user can use this secret key to decrypt the message. The main challenge of single authority Attribute-Based Encryption is preventing collusion. This problem is addressed in this paper by introducing the multiauthority scheme. A scheme is intrduced by [1] for the identification of encrypted information with complex access control called CP-ABE. According to the work, the author proposed that even if the storage server is untrusted, the information could be kept confidential. And also, this method is secure against collusion attacks. In this paper, the author used the concept of attributes.

The characteristics are used to define the credentials of a user, and information is determined by a party encryption policy for who can decrypt. A new security model of honest but curious servers to identify possible attacks [27]. This scheme allows the authority to cancel any characteristic at any period by inserting a minimum load on the user. This method applies to Key Policy ABE. The author addresses the attribute revocation issue in the attribute-based system. [11] addressed some challenging issues in different scenarios and different policy updates. Ciphertext-Policy ABE is a promising solution for these issues like access policy and data storage. Some problems can occur to these types of framework that is revocation problem. In this article, the author develops an access control policy that has an effective revocation capability that is user and attribute.

An access policy is efficient to promise safety in cloud data [26]. Data storing in insecure cloud third parity become an issue in deciding access policy. For this ciphertext-policy, ABE is an adequate scheme because the owner is deciding policy. But it is difficult to introduce due to revocation issue. For this reason, the author introduces an effective revocation access policy that applies to multiple authority scheme. Also, this system can achieve both backward and forward safety. [6] suggests access structures can represent by the monotonic or non-monotonic way. One of the best techniques to store data with encryption is ABE. In this article, the author addresses the key escrow problem and find a perfect solution to solve it. The author resolves this key escrow problem by introducing 2pc protocols in the system. A new multilevel secret sharing system is introduced by [13] expanding the Shamir's to the global threshold that is exclusively higher than the compartment's number of thresholds. The article also shows how to use the polynomial interpolation-based threshold secret sharing systems. Proposes two effective systems that linearly proportion the number of public shares to the number of respondents.

A novel multi secret-sharing system based on Hermite interpolation is implemented [21]. According to the threshold as well as value and number of secrets the proposed scheme is dynamic to the changes. This scheme has a key feature of multi-usability. The article also provides multi-user function over elliptic curves by solving the discrete logarithm issue. This assumes that n members and the dealer is not one of them. He will unbiassed set the system up and issues the standards. The members will give their shares to a combiner, who is any one of the members for reconstructing the secret. By extending the Key-Policy ABE with attribute privacy preservation in cloud storage [8] proposes an Effective Attribute-based Access Control with Authorized Search system. It is further effective than current solutions on calculation and storing expenses. The aim is to suggest Key Policy-ABE with partly concealed characteristics constructed on the search for sensitive keywords and to demonstrate that the system is safe under the q-2 decisional bilinear DH supposition. [14] defines a threshold version of the Localized Multi-secret Sharing Scheme (LMSS). The author provides lower limits on The decryption share size of localized multi-secret sharing systems in a particular background and gives clear development of systems. The author also analyses various methods to relax the model providing trade-offs among the shared scope and between the number of safety assurances given by this system to allow the development of a small number of shared systems.

3 Attribute-based Encryption

The concept of ABE is first introduced by Sahai and Waters on the paper called Fuzzy Identity-Based Encryption [19]. In ABE data is encrypted using a set of attributes. ABE is a scheme in which each user is identified by a set of attributes, and some function of those attributes is used to determine decryption ability for each ciphertext [3]. Attribute-Based Encryption (ABE) is a type of public-key encryption. Using this public-key encryption, the message is encrypted and decrypted using a public key and private key respectively for a specific receiver. In attribute-based encryption, it contains a set of attributes where secret key and ciphertext are attributes dependent. Here a user can decrypt the ciphertext only when the attributes of the ciphertext match the attributes of the user key. The user encrypts the data using an access policy or access structure. An example of access policy is Area=Italy AND age <30 AND Business=Researcher. There are two types of access structure monotonic and non-monotonic. Two kinds of ABE exist they are Key-Policy ABE and Ciphertext-Policy ABE. Due to some issues in Key-Policy ABE such as access policy is storing in

a third party so it may have a chance that he can change access policy so we are mainly focusing on Ciphertext-Policy ABE.

3.1 Key Policy Attribute-based Encryption

In KP-ABE secret keys for users are generated based on access structure and encryption of data is done based on a set of attributes. The decryption of the message is done when user attributes satisfy the access structure. The most common access structure used is a tree-based access structure. The main issue in Key-Policy ABE is access policy are storing in a third party so it may have a chance that he can change access policy. In this paper, we are mainly focusing on ciphertext policy attribute-based encryption.

3.2 Ciphertext Policy Attribute-based Encryption

In CP-ABE data is encrypted using access trees and a set of attributes is used for generating user's secret keys. There are two types of system models for the CP-ABE scheme, single authority and multi-authority. In single authority systems in Figure 1 there is only one attribute authority called central authority and all attributes of the system are managed by this central authority. In the case of a multi-authority system in Figure 2, there are multiple attribute authority and attributes that are shared across these authorities. The main entities of system architecture are:

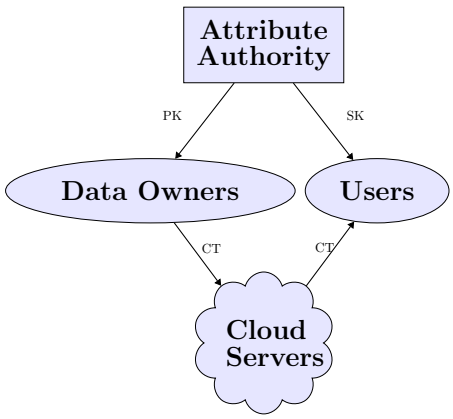


Figure 1: Single authority system

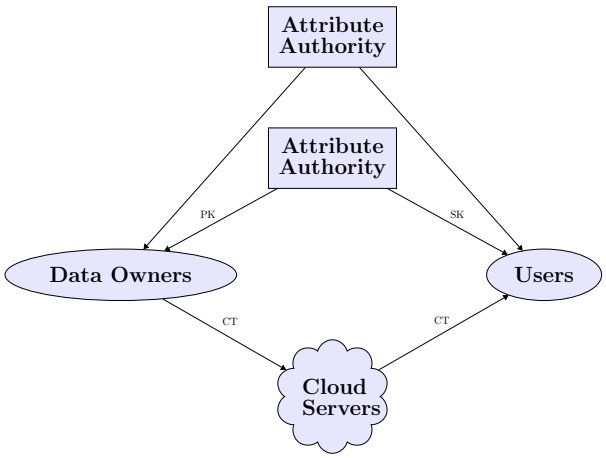


Figure 2: Multi-authority system

Attribute Authority: The entity which checks the validity of user attributes ana sets up parameters and secret key. It is responsible for key updating and revocation process.

Data Owners: The entity that wants to store data safely on cloud storage. This is achieved by outsourcing data that is sending encrypted data to the cloud. The data owner is the one who defines access policy to the data.

Data Users: The entity which has a set of attributes and secret key to access the ciphertext.

Cloud Servers: The entity which has the charge of storing encrypted data from data owners.

In the framework, five algorithms are used:

Setup: The attribute authority generates the parameters for Public Key (PK) and Secret Key (SK).

KeyGeneration: The attribute authority generates Public Key and Secret Key based on the parameters and attribute set.

Encryption: The message is encrypted using a Public Key and access structure.

Decryption: The ciphertext is decrypted by data users using the secret key.

Delegation: The updating and revocation process are handled by taking the private key and regenerate a new key.

3.2.1 CP-ABE schemes

In this section, we are discussing recent CP-ABE schemes and the different access policies. A directly revocable attribute-based encryption (DR-CP-ABE) scheme [22] is introduced by constructing a complete subtree for access structure and is solved by the subset cover method. For data confidentiality and to keep the privacy of signcryptor [18] a ciphertext-policy attribute-based signcryption (CP-ABSC) is used with an access policy of monotone span programs. Using this method, they achieved shorter ciphertext size when compared to the schemes which already exists. A new system architecture was introduced by [15] for securely sharing data to a cloud by a user of limited resources. Using a linear secret sharing scheme it achieves high- efficiency, fine-grainedness, and data confidentiality. Managing access policy in a cloud storage

system is an importing task [23] and a scalable ciphertext-policy attribute-based encryption (SCP-ABE) is introduced for that.

The most common access policy used is a monotonic access structure but here a different access policy called blocked linear secret sharing scheme (BLSSS) is used. A matrix is used to describe a tree circuit of access structure due to this special scheme scalability is achieved. Another recent ABE scheme is the ciphertext-policy attribute-based hierarchical document collection encryption (CP-ABHE) scheme [5] which access trees are combined according to attribute sets. This scheme uses an access tree that contains only "AND" gate and so it is called monotone access structure. Security of outsourced data is the main concern of personal health records in the cloud [7] and a hierarchical CP-ABE scheme with multiple authorities was introduced. As hierarchical files are sharing the author used a layered model of access structure. Access structures are combined to form a single one as this paper is dealing with the sharing of hierarchical files.

An ABE scheme that is applicable for resource-constrained mobile devices [9] is a lightweight attribute-based encryption (LBE) scheme. This paper introduced a scheme that is applicable to cyber-physical systems and based on monotone access structure. This means AND gate are used as access policy. Another one scheme for mobile cloud is decentralized multi-authority attribute-based encryption [20] scheme which is an appropriate solution of key escrow problem and is based on a monotone access policy. For preserving the privacy of the user, a CP-ABE scheme with hidden access policy [28] is introduced called hidden access policy CP-ABE (HP-CP-ABE) scheme is introduced. This scheme improves data confidentiality and protects the user's privacy. This is achieved by an access policy of a monotonic access structure with the "AND" gate.

3.2.2 Challenges

In this section, we are discussing different challenges and proposed solutions. The main problem of the existing ABE scheme is the lack of user revocation mechanism [22] and one solution for this is by introducing one more entity in the system architecture called user revocation centre (URC). URC is maintaining a revocation list and all revocation tasks are outsource to this third-party organization. Lack of scheme which achieves security goals such as data confidentiality, privacy, fine-grainedness [15, 18] of cloud data sharing still unresolved. Managing access policy [23] is a critical issue of the ABE scheme. The existing policy managing schemes have high computation complexity and communication overhead. One solution for this is using a blocked linear secret sharing scheme (BLSSS) in which a matrix is used to describe tree-circuit. Another one main challenge concerning limited resource devices and mobile cloud [9, 20] is to reduce computation and communication overhead.

Even though many ABE schemes are introduced for this challenge anyone can find out a better scheme that can achieve low processing time, which takes limited storage space, limited resources and limited energy. One major challenge addressed by [28] is protecting user's personal privacy. This is an important issue that has to address at present. Along with this one should concentrate on developing a scheme that is efficient in communication and computation cost. Some issues addressed by the author are data confidentiality, efficient decryption test, efficiencies such as parameter size and time complexity of algorithm. Scalable user revocation [24] is an important issue. Here one can develop an ABE scheme that provides both forward security and backward security. The above discussed are the major challenges faced by the recent ABE schemes.

3.2.3 Security

The different frameworks used to prove the system security is discussed in this section. The security of an ABE scheme can be proven using dual system encryption in the standard model [22]. This scheme is indistinguishable under chosen-plaintext attacks. Most of the ABE schemes prove their security under decisional bilinear Diffie-Hellman exponent (dBDHE) [5, 7, 9, 18, 20, 28]. For some schemes, security analysis is done with random oracles [18, 23]. Some paper proves the security of the scheme against chosen-ciphertext attacks [15]. Security can also be proven against a collusion attack [23]. One of the main security issue is Key escrow problem. Existing multi-authority attribute-based encryption schemes however still require a trusted central authority to publish system parameters and to generate user secret keys. They give to the trusted central authority enough privileges to access the plaintext information meant for the user, a problem referred to as key escrow issue [20]. This scheme solves the key escrow problem by removing the central authority, without making use of any global user identity.

3.3 Comparison

In this section, we compare the various existing CP-ABE schemes and is given in the following Table 1. we compare different scheme against the access structure used, the number of authorities, revocation mechanism is addressed or not and the security assumption followed by the scheme.

3.4 Conclusion

In this paper, we have done a comprehensive survey of recent CP-ABE schemes along with the access policy used by these schemes. We also discussed the major challenges faced by CP-ABE schemes and the existing solution for that issue. One who wants to work on this can take any mentioned issue and can develop a better CP-ABE scheme. The security analysis model of different schemes also discussed. Finally, we compare different

Table 1: Comparison of CP-ABE schemes

Scheme	Access structure	Authority	Revocation mechanism	Security assumption
Fu, J., and Wang, N. [5]	Tree	Single	No	DBDH
Gadge.,Snehlata et al [6]	Tree	Single	Yes	-
Guo, R., et al [7]	Tree	Multiple	Yes	DBDH
Hao, Liu., et al [8]	Tree	Single	No	DBDH
He, Q., et al [9]	LSSS	Single	Yes	DBDH
Hur, Junbeom., et al [11]	Tree	Single	Yes	-
Li, J., et al [15]	LSSS	Single	No	CCA
Liu.,Fan., et al [17]	LSSS	Single	No	DBDHE
Rao, Y. S. [18]	Tree	Single	Yes	DBDH
Arthur Sandor, V. K., et al [20]	LSSS	Multiple	No	DBDH
Wang, H., et al [22]	LSSS	Single	Yes	CPA
Wang, J., et al [23]	BLSSS	Single	Yes	DBDH
Wei, J., et al [24]	LSSS	Multiple	Yes	DBDH
Yang, Kan., et al [26]	LSSS	Multiple	Yes	BDHE
Zhang, L., et al [28]	Tree	Single	No	DBDH

schemes against the access structure used, the number of authorities, revocation mechanism is addressed or not and the security assumption followed by the scheme.

References

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy (SP'07)*, IEEE, pp. 321–334, 2007.

[2] Z. Cao, L. Liu, and Z. Guo, "Ruminations on attributebased encryption," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 9–19, 2018.

[3] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography Conference*, Springer, pp. 515–534, 2007.

[4] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments", *International Journal of Network Security*, vol. 16, no. 1, pp. 1-13, 2014.

[5] J. Fu and N. Wang, "A practical attribute-based document collection hierarchical encryption scheme in cloud computing," *IEEE Access*, vol. 7, pp. 36 218–36 232, 2019.

[6] M. S. V. Gadge, "Analysis and security based on attribute based encryption for data sharing," *International Journal of Emerging Research in Management & Technology*, pp. 2278–9359, 2014.

[7] R. Guo, X. Li, D. Zheng, and Y. Zhang, "An attribute-based encryption scheme with multiple authorities on hierarchical personal health record in cloud," *The Journal of Supercomputing*, pp. 1–20, 2018.

[8] J. Hao, J. Liu, H. Wang, L. Liu, M. Xian, and X. Shen, "Efficient attribute-based access control with authorized search in cloud storage," *IEEE Access*, 2019.

[9] Q. He, N. Zhang, Y. Wei, and Y. Zhang, "Lightweight attribute based encryption scheme for mobile cloud assisted cyber-physical systems," *Computer Networks*, vol. 140, pp. 163–173, 2018.

[10] P. Hu and H. Gao, "A key-policy attribute-based encryption scheme for general circuit from bilinear maps," *International Journal of Network Security*, vol. 19, no. 5, pp. 704–710, 2017.

[11] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2010.

[12] M. S. Hwang, T. H. Sun, C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of public auditing in cloud storage service," *Journal of Circuits, Systems, and Computers*, vol. 26, no. 5, 2017.

[13] P. S. Kumar, R. R. Kurra, A. N. Tentu, and G. Padmavathi, "Multi-level secret sharing scheme for mobile ad-hoc networks," *International Journal of Advanced Networking and Applications*, vol. 6, no. 2, pp. 22–53, 2014.

[14] T. M. Laing, K. M. Martin, M. B. Paterson, and D. R. Stinson, "Localised multisecret sharing," *Cryptography and Communications*, vol. 9, no. 5, pp. 581–597, 2017.

[15] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, vol. 72, pp. 1–12, 2018.

[16] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of attribute-based access control with user revocation in cloud data storage," *International Journal of Network Security*, vol. 18, no. 5, pp. 900–916, 2016.

[17] Z. Liu and Y. Fan, "Provably secure searchable attribute-based authenticated encryption scheme," *International Journal of Network Security*, vol. 21, no. 2, pp. 177–190, 2019.

[18] Y. S. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Generation Computer Systems*, vol. 67, pp. 133–151, 2017.

[19] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2005, pp. 457–473.

[20] V. K. A. Sandor, Y. Lin, X. Li, F. Lin, and S. Zhang, "Efficient decentralized multi-authority attribute based encryption for mobile cloud data storage," *Journal of Network and Computer Applications*, vol. 129, pp. 25–36, 2019.

[21] M. H. Tadayon, H. Khanmohammadi, and M. S. Haghighi, "Dynamic and verifiable multi-secret sharing scheme based on hermite interpolation and bilinear maps," *IET Information Security*, vol. 9, no. 4, pp. 234–239, 2014.

[22] H. Wang, Z. Zheng, L. Wu, and P. Li, "New directly revocable attribute-based encryption scheme and its application in cloud storage environment," *Cluster Computing*, vol. 20, no. 3, pp. 2385–2392, 2017.

[23] J. Wang, C. Huang, N. N. Xiong, and J. Wang, "Blocked linear secret sharing scheme for scalable attribute based encryption in manageable cloud storage system," *Information Sciences*, vol. 424, pp. 1–26, 2018.

[24] J. Wei, W. Liu, and X. Hu, "Secure and efficient attribute-based access control for multiauthority cloud storage," *International Journal of Network Security*, vol. 12, no. 2, pp. 1731–1742, 2016.

[25] C. Yang, Q. Chen, Y. Liu, "Fine-grained outsourced data deletion scheme in cloud computing," *International Journal of Electronics and Information Engineering*, vol. 11, no. 2, pp. 81–98, 2019.

[26] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," *IIEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1735–1744, 2013.

[27] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 261–270, 2010.

[28] L. Zhang, Y. Cui, and Y. Mu, "Improving security and privacy attribute based data sharing in cloud computing," *IEEE Systems Journal*, 2019.

Biography

Ancy P. R. is currently pursuing her PhD in Computer Science and Engineering from CHRIST(Deemed to be University), Bangalore, India. She has received her M.Tech in Computer Science and Engineering from Marian Engineering College, India. She has worked as Assistant Professor for over three years.Her research interest includes Network Security, Cryptography and Cloud Computing.

Addepalli V. N. Krishna is working as Professor in Computer Science and Engineering department at CHRIST(Deemed to be University), Bangalore, India. He received Ph.D in 2010 at Department of Computer Science and engineering, Acharya Nagarjuna University, AP, India and M. Tech degree in 2001 from department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi, India. He is having 25 years of teaching experience at undergraduate and postgraduate engineering level. His research interests includes Security algorithm design in wireless sensor networks, Advanced key management algorithms for computer security and protocol design for network security.

K. Balachandran is working as H.O.D and Professor in Computer Science and Engineering department at CHRIST(Deemed to be University), Bangalore, India. He received Ph.D in 2015 at Department of Computer Science and engineering, Anna University, India.He pursued his BSc and MCA from Bharathidasan University, MSc Physics from Annamalai University, MTech (IT) from Allahabad Agricultural University and MPhil from Alagappa University. He has worked as a scientific officer for two decades with the Department Atomic Energy, India. Many of his research papers have won the best paper award at national and international conferences. His area of research includes Data Science, Networks, Big data Analytics and Computer Architecture.

Balamurugan M. is working as Associate Professor in Computer Science and Engineering department at CHRIST(Deemed to be University), Bangalore, India. He received Ph.D in 2013 at Department of Computer Science and engineering, Anna University, India. He is having 13 years of teaching experience at undergraduate and postgraduate engineering level. His area of research includes Machine Learning, Big data Analytics and Data mining.

O. S. Gnana Prakasi is working as Assistant Professor in Computer Science and Engineering department at CHRIST(Deemed to be University), Bangalore, India. She received Ph.D. in 2017 from Anna University, India. She received her B.Tech. degree in Information Technology and M.E. degree in Software Engineering from Anna University. Her research interest includes, Mobile Ad hoc Networks, IoT, Software Engineering and Machine Learning.

A Sequential Cipher Algorithm Based on Feedback Discrete Hopfield Neural Network and Logistic Chaotic Sequence

Shoulin Yin, Jie Liu, and Lin Teng
(Corresponding author: Jie Liu)

Software College, Shenyang Normal University
Shenyang, Liaoning 110034, China
(Received Dec. 22, 2018; Revised and Accepted June 24, 2019; First Online Feb. 26, 2020)

Abstract

The traditional Logistic chaotic sequence is easy to be reconstructed and when using it to encrypt data, it is easily to leak sensitive information. Therefore, an external key is introduced to encrypt the initial value and parameters of the Logistic equation. Then we conduct sensitive and diffusion process for feedback discrete Hopfield neural network based on the Logistic sequence sensitivity to the initial value. And by updating the control parameters, it produces good sequence key with random chaos iterative operation. The final algorithm analysis and simulation experiments show that the key of the algorithm has a good sensitivity, the generated random sequence has good randomness, which satisfies the requirement of cryptography. The pseudo-random sequences constructed by the algorithm are characterized by good randomness and complexity.

Keywords: Feedback Discrete Hopfield Neural Network; Logistic Chaotic Sequence; Sensitive and Diffusion Process

1 Introduction

With the rapid development of computer technology and multimedia technology, multimedia communication has gradually become an important way for people communicating with each other [2, 3, 9]. Information security has been an important issue that is closely related to our lives [7, 8, 17]. The most important way to protect information security is data encryption. The discrete Hopfield neural network was proposed by Liu [10], so the combination of chaos and neural networks for data encryption has been continuously developed. At present, the main research methods are as follows:

1) Using a known chaotic sequence to train the neural network model, so that the neural network can approximate the same chaotic sequence, then use the known chaotic sequence as the public key, and the trained neural network weight and threshold parameters are as the private key to implement data encryption;

2) Using the chaotic attraction of the discrete Hopfield network and the unidirectional mapping of the initial state and the attractor, the stable attractor of the discrete Hopfield network can be encrypted as a key.

The neural network mutual learning model and the chaotic system were mutually interfered, and a new composite stream cipher was proposed in reference [15]. A novel chaos-based hybrid encryption algorithm design for secure and effective image encryption was presented. To design the algorithm, the Zhongtang chaotic system had been selected because of its rich dynamic features and its dynamical analysis was performed. On the base of this system, a new chaos-based random number generator (RNG) was developed and usefulness of the designed RNG in an encryption process was shown over NIST 800-22 randomness tests in reference [21]. In reference [11], the Hermite orthogonal polynomial was introduced into the neural network excitation layer, and the "one-time-one-density" asynchronous encryption algorithm was realized. Pareek [12] proposed a chaotic encryption scheme by combining the chaotic attraction of Hopfield network with the linear feedback shift register. The most important thing to apply chaos and neural network to data encryption was that it used the chaotic characteristics of chaotic sequences. However, Zhang [19] also pointed out that there are defects in the sequence reconstruction of chaotic sequences in data encryption. Moreover, the reference [4] completely reconstructed the four-point and sixteen-point sequence fragments of the Logistic equation. Therefore, the simple chaotic sequence encryption method is the risk of being cracked. Aiming at this problem, this paper introduces an external key to encrypt the Logistic equation, and proposes a piecewise discrete Hopfield neural network model. Through a sensitive and diffusion

process, the chaotic network model also has good sensitivity and good random generation sequence.

The rests of the paper are organized as follows. Section 2 introduces the Logistic mapping. Discrete Hopfield neural network Model is illustrated in Section 3. Section 4 and Section 5 outline the quantitative processing and new algorithm analysis. Section 6 finally concludes this paper.

2 Logistic Mapping

Logistic mapping [5,16,18,20] is a classic chaotic sequence mapping. Because of its simple implementation method, it is easy to be reconstructed by the thief using phase space to construct the form of chaotic equation. Therefore, in order to enhance the confidentiality of the Logistic mapping, this paper introduces an external key to encrypt the initial values and parameters of the Logistic map.

The 24-bit binary number K_1 is used as an external key to initialize the initial values and parameters of the Logistic map. Let K_1 be expressed in hexadecimal as $K1 = k_1k_2k_3k_4k_5k_6$, and "$k_1k_2k_3k_4k_5k_6$" is an external key. Where "$k_1k_2k_3$" is used to generate the input control parameter r of the Logistic mapping. "$k_4k_5k_6$" is used to generate the initial value X_0 of the Logistic mapping.

In this article, the Logistic mapping is as follows:

$$X_{n+1} = r \times X_n(1 - X_n), X_n \in (0, 1).$$

Where r is the input control parameter and X_0 is the initial value of the Logistic mapping. They are generated by an operation of an external key. During the operation, "$k_1k_2k_3k_4k_5k_6$" is converted into a corresponding decimal number for operation. Therefore, we can get,

$$\begin{aligned} r &= 4 + \frac{k_1/16^2 + k_2/16 + k_3}{16^2}. \\ X_0 &= \frac{k_4/16^2 + k_5/16 + k_6}{16^3}. \end{aligned} \quad (1)$$

3 Discrete Hopfield Neural Network Model

Assuming that each Neuron state is 0 or 1, the next state $S_i(t+1)$ depends on current state $S_i(t)$. So

$$S_i(t+1) = \sigma(\Sigma_{j=0}^{N-1} T_{i,j} S_i(t) + \theta_j), i = 0, 1, \cdots, N-1,$$

where the threshold of neuron i is θ_j, and the connection weight between neuron j and i is $T_{i,j}$. $\sigma(x)$ is any nonlinear function, is set as the unit step function. Then the energy function of the system at time t is:

$$E(t) = 0.5 \sum_{i,j} T_{i,j} S_i(t) S_j(t).$$

Hopfield has proved that Equation (1) decreases monotonically with the evolution of system state, and eventually it will reach a stable state, namely chaos attractor. And there is an unpredictable relationship between the state messages contained in its attraction domain. If the join weight matrix T is changed, the attractor and its corresponding attraction domain will change too. After the introduction of random transformation matrix H, the initial state S and attractor S_μ will be updated by $\hat{S} = SH$ and $\hat{S}_\mu = S_\mu H$ and get new initial state S and attractor $\hat{S}_\mu$, and this process is unilateral, irreversible [14].

4 Quantitative Processing

In order to apply the random sequence generated by discrete Hopfield neural network into data encryption, this paper introduces the conversion function $T(x)$ to convert the generated random sequence into a $0-1$ random sequence. $T(x)$ is defined as follows:

$$T(x) = 0, x \in [2n/N, (2n+1)/N].$$

Where N is an integer in interval $[10, +\infty]$, n is an integer in interval $[0, \tilde{N}]$, $\tilde{N} = \lfloor N/2 \rfloor$. Since the random value generated by the network is in the interval $(0, 1)$, this paper divides the interval $(0, 1)$ into N equal parts. The larger the N value is, the finer the interval division is and the data precision is higher.

4.1 Generating a $0-1$ Random Sequence

1) Input the key K_1 to initialize the Logistic mapping and iteratively calculate the Logistic mapping 200 times. Let $X = [x_{101}x_{102}x_{103}, \cdots, x_{199}x_{200}]$ be used to store the next 100 chaotic values.

2) Use X to initialize the weight, threshold of the diffusion matrix W and the discrete Hopfield neural network. W is a 4×4 matrix, W_1 is a 2×4 matrix, B_1 is a 2×1 matrix, W_2 is a 1×2 matrix, and B_2 is a 1×1 matrix.Extract elements from X to initialize W, W_1, B_1, W_2, and B_2, respectively.

3) Input the key K_2 and obtain $\tilde{D}_1$ through initial grouping and transformation.

4) Input $\tilde{D}_1$ to the discrete Hopfield neural network, and a random value D_3 is obtained through network operation; then the values of the control parameters Q_1 and Q_2 are continuously updated until a random sequence of the desired length is obtained.

5) The generated random sequence is converted to a corresponding binary random sequence by a quantization function $T(x)$.

5 New Algorithm Analysis

This paper analyzes the key space size of the new algorithm, the number of 0/1 statistics, the correlation of random sequences and the sensitivity of the key through theoretical analysis and experimental simulation, and draws

Table 1: Statistical analysis results

Testing parameter	Sequence 1	Sequence 2
n_0	128	130
n_1	129	127
r	129	131
y_1	$5.37e^{-4}$	$5.98e^{-5}$
y_2	1.3214	-6.7892
y_3	-0.1179	0.1225

Table 2: Statistical analysis result of LET

Testing parameter	Sequence 1	Sequence 2
n_0	127	126
n_1	129	132
r	130	128
y_1	$6.18e^{-5}$	$2.51e^{-4}$
y_2	-0.972	-0.864
y_3	0.259	-0.397

corresponding conclusions. The simulation experiment data is as follows: $n_0 = 120$, $n_1 = 8$, $n_2 = 10$, $\alpha = 2$, $N = 128$, $Q_1 = [0.5, 0.5]^T$, $Q_2 = 0.5$.

5.1 Key Space Analysis

In this paper, the encryption key consists of K_1 and K_2, and the key space is determined by the length of K_1 and K_2. Let their lengths be L_1 and L_2, respectively. The key space is $2^{(L_1+L_2)}$. The larger $L_1 + L_2$ is, the larger the key space is. In this paper, $L_1 = 24$, $L_2 = 16$, and the key space size is 2^{40}. Obviously, the length of L_2 is variable. Increasing the length of L_2 can increase the size of the key space. However, when the length of the key L_2 is increased, the number of inputs of the discrete Hopfield neural network will also increase. The number of corresponding discrete Hopfield neural network layers will increase too, the time overhead of the network iterative operation will increase. When the generated random sequence is very large, the time overhead of running the entire network will be very large too.

5.2 Statistical Analysis

A valid binary random sequence must satisfy the 0/1 ratio. Therefore, the purpose of this test is to determine if the ratio of 0/1 in the sequence is approximately equal to the ratio of 0/1 in the true random sequence. At the same time, this paper refers to the method of [13] for the frequency test, sequence test and run test of the generated random sequence. Therefore, two random sequences of length 256 are randomly selected for statistical analysis. The analysis results are shown in Table 1.

In Table 3, n_0 denotes a number of 0, n_1 is number of 1, r is total number of run test, y_1 is frequency test value, y_2 is sequence test value, y_3 is run test value.

It can be seen from Table 1 that the number of "0" and "1" in sequence 1 and sequence 2 are nearly equal, satisfying the requirements of random sequence. At the same time, the frequency test value y_1 is less than 3.84, which can pass the frequency test. The sequence test value y_2 is less than 5.99, which can pass the sequence test. The run test value y_3 is much less than 1.96, which can pass the run test. Therefore, the generated random sequence has good randomness.

Table 3: Statistical analysis result of RBC

Testing parameter	Sequence 1	Sequence 2
n_0	135	140
n_1	121	116
r	140	135
y_1	$5.34e^{-3}$	$1.23e^{-2}$
y_2	1.467	2.132
y_3	1.792	1.235

In order to further verify the randomness of the generated sequences, the above two randomly selected random sequences are compared with the statistical analysis results in the reference LET [6] and RBC [1]. The statistical analysis results of the LET and RBC are shown in Table 2.

From Table 2, it can be seen that the difference between "0" and "1" in the random sequence generated by new algorithm is smaller than the difference between "0" and "1" in the literature [13]. Therefore, the 0/1 sequence generated in this paper is more random. At the same time, the frequency test value y_1, the sequence test value y_2 and the run test value y_3 are all smaller than the corresponding values in [6]. Therefore, the 0/1 sequence cant better pass the frequency test, sequence test and run test. Compared with the literature [6], the difference between "0" and "1" in the random sequence in this paper is close to the difference between "0" and "1" in the literature [6]. And, the frequency test value y_1 and the run test value y_3 in this paper are smaller than the corresponding values in the literature [6] in some cases, which indicates that the 0/1 sequence can better pass the frequency test and the run test in some cases. However, the sequence test value y_2 in this paper is larger than the corresponding value in [6], which indicates that the 0/1 sequence generated in [6] can pass the sequence test better. In general, the proposed algorithm outperforms the literature in frequency test, sequence test and run test [1], and in some cases is superior to the literature [6].

5.3 Correlation Analysis

The change in the autocorrelation function of the sequence is smaller, the better the randomness of the sequence is. The cross-correlation function of the sequence

is close to zero, the more unrelated the two sequences are. Figure 1 is the autocorrelation function of the 0/1 sequence generated when the initial keys are K_1 and K_2. Figure 2 is a comparison of the 0/1 sequence generated when the key K_1 or K_2 is randomly changed by one bit. The solid line represents the original sequence and the dotted line represents the new sequence. Figure 3 is a cross-correlation function diagram of two sequences.

5.4 Key Sensitivity Analysis

The chaotic sequence generated by the Logistic map has good sensitivity to the initial value. Since the key K_1 is used to generate the initial value of the Logistic equation, K_1 also has good sensitivity. At the same time, this paper uses Logistic map to sensitive and diffuse the key K_2, which makes K_2 also have good sensitivity. In order to verify the sensitivities of the keys K_1 and K_2, one of the keys is changed, and the percentage of the difference between the new random sequence and the original random sequence is counted. Let i denote the position number corresponding to each of K_1 and K_2, then $1 \leq i \leq \vartheta$, ϑ is the sum of the lengths of the keys K_1 and K_2. In this paper, $\vartheta = 40$. NP represents the percentage of the new random sequence and the original random sequence as the percentage of the total number of sequences. The formula of NP is:

$$NP(i) = (\sum_{n=1}^{NK}(A(n)))/(NK) \times 100\%.$$

$$A(n) = \begin{cases} 0 & D(n) = D'(n) \\ 1 & D(n) \neq D'(n). \end{cases}$$

Where $D(n)$ and $D'(n)$ represent the original random sequence and the new random sequence, respectively. NK represents the total number of bits of the sequence $D(n)$. $NP(i)$ represents the percentage corresponding to the change of the $i-th$ bit.

Depending on the strict avalanche criterion in the block cipher measure, changing any bit in the key should result in a change of approximately 50% of the bits in the ciphertext. Figure 4 shows the percentage statistics obtained for the key length $\vartheta = 40$ and the sequence length $NK = 20000$.

It can be seen that the percentage of the sequence generated when the key changes by one bit is close to 50%, which satisfies the strict avalanche criterion.Therefore, both keys K1 and K2 are sensitive.

In summary, the 0/1 sequence generated by the algorithm has good randomness and the key has strong sensitivity.In addition, this paper uses a 24-bit external key to generate chaotic initial values and control parameters. Compared with the literature [16] directly using chaotic initial values and control parameters as keys, the key of this paper is more convenient to manage. At the same time, it solves the problem that the Logistic sequence proposed in [15] is easy to be reconstructed, which makes the security of the key better.

5.5 Anti-Matrix Analysis and Difference Analysis

If using a public key encryption system, from the orthogonal decomposition, the singular value decomposition and triangular decomposition, they demonstrate that the HNN network's safety is reliable. Because the entire password system is irregular in the process of encryption, even if the same clear sequence encrypted, the obtained ciphertext sequence cannot be the same. Moreover, in the decryption process using the attracting method, the differential cryptanalysis for the algorithm is invalid.

6 Conclusion

In this paper, we propose a sequential cipher algorithm based on feedback discrete Hopfield neural network and logistic chaotic sequence. The key is processed with sensitive and diffused to enhance its sensitivity. After experiment demonstration, the new algorithm has reliable security and high efficiency than other methods.

Acknowledgments

This work is supported by Science & Technology Planning Project of Gansu Province under Grant No. 1610RJZE135.

References

[1] S. A. K. Albermany, F. R. Hamade, G. A. Safdar, "New random block cipher algorithm," in *International Conference on Current Research in Computer Science & Information Technology*, 2017. DOI: 10.1109/CRCSIT.2017.7965555.

[2] J. Gao, P. Li, Z. Chen, "A canonical polyadic deep convolutional computation model for big data feature learning in Internet of Things," *Future Generation Computer Systems*, vol. 99, pp. 508-516, Oct. 2019.

[3] J. Gao, J. Li and Y. Li, "Approximate event detection over multi-modal sensing data," *Journal of Combinatorial Optimization*, vol. 32, pp. 1002-1016, 2016.

[4] J. S. Gao, B. Y. Sun, W. Han, "Construction of the control orbit function based on the chaos theory," *Electric Machines & Control*, 2002-02, 2002.

[5] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves," *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 73-84, Mar. 2004.

[6] R. Karmakar, S. Chatopadhyay, R. Kapur, "Encrypt flip-flop: A novel logic encryption technique for sequential circuits," *Computer Science*, 2018. (`https://arxiv.org/pdf/1801.04961.pdf`)

[7] S. Khatoon, T. Thakur, B. Singh, "A provable secure and escrow-able authenticated group key agreement

protocol without NAXOS trick," *International Journal of Computer Applications*, vol. 171, no. 3, pp. 1-8, 2017.

[8] M. Liu and F. Long, "Stream cipher algorithm based on piecewise linear chaotic networks," *Computer Applications and Software*, vol. 33, no. 9, pp. 306-309, 2016.

[9] J. Liu, S. L. Yin, H. Li and L. Teng, "A density-based clustering method for k-anonymity privacy protection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 1, pp. 12-18, Jan. 2017.

[10] Z. Liu, L. Zhang, L. V. Xue, *et al.*, "Evaluation method about bus scheduling based on discrete hopfield neural network," *Journal of Transportation Systems Engineering & Information Technology*, vol. 11, no. 2, pp. 77-83, 2011.

[11] Q. Meng, S. Yu, H. Liu, *et al.*, "A novel blind detection algorithm based on improved compound sine chaotic neural networks," in *IEEE International Conference on Communication Technology*, 2016. DOI: 10.1109/ICCT.2015.7399973.

[12] N. K. Pareek, "Design and analysis of a novel digital image encryption scheme," *International Journal of Network Security & Its Applications*, vol. 4, no. 2, 2012.

[13] L. Teng, H. Li, J. Liu and S. Yin, "An efficient and secure cipher-text retrieval scheme based on mixed homomorphic encryption and multi-attribute sorting method," *International Journal of Network Security*, vol. 20, no. 5, pp. 872-878, 2018.

[14] L. Teng, H. Li, S. Yin, "A multi-keyword search algorithm based on polynomial function and safety inner-product method in secure cloud environment," *International Journal of Network Security*, vol. 8, no. 2, 2017.

[15] C. Tie-Ming, J. Rong-Rong, "New hybrid stream cipher based on chaos and neural networks," *Acta Physica Sinica*, vol. 62, no. 4, pp. 191-201, 2013.

[16] S. F. Tzeng, M. S. Hwang, "Digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem," *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 61-71, Mar. 2004.

[17] S. L. Yin and J. Liu, "A k-means approach for map-reduce model and social network privacy protection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 6, pp. 1215-1221, Nov. 2016.

[18] S. Yin, L. Teng, J. Liu, "Distributed searchable asymmetric encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 4, no. 3, pp. 684-694, 2016.

[19] X. Zhang, F. Han, Y. Niu, "Chaotic image encryption algorithm based on bit permutation and dynamic DNA encoding," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 11, 2017. (`https://doi.org/10.1155/2017/6919675`)

[20] Q. Zhang, L. T. Yang, X. Liu, Z. Chen, and P. Li, "A tucker deep computation model for mobile multimedia feature learning," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 13, no. 3, pp. 1-39:18, 2017.

[21] U. Cavu?o?lu, S. Kacar, A. Zengin, I. Pehlivan, "A novel hybrid encryption algorithm based on chaos and S-AES algorithm," *Nonlinear Dynamics*, vol. 92, no. 4, pp. 1745-1759, 2018.

Biography

Shoulin Yin received the B.Eng. and M.Eng. degree from Shenyang Normal University, Shenyang, Liaoning province, China in 2016 and 2013 respectively. Now, he is a doctor in Harbin Institute of Technology. His research interests include Multimedia Security, Network Security, Filter Algorithm, image processing and Data Mining. Email:352720214@qq.com.

Jie Liu is a full professor in Software College, Shenyang Normal University. He received his B.S. and M.S. degrees from Harbin Institute of Technology. He is also a master's supervisor. He has research interests in wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Professor Liu had published more than 30 international journal papers (SCI or EI journals) on the above research fields. Email:ljnan127@163.com.

Lin Teng received the B.Eng. degree from Shenyang Normal University, Shenyang , Liaoning province, China in 2016. Now, she is a laboratory assistant in Software College, Shenyang Normal University. Her research interests include Multimedia Security, Network Security, Filter Algorithm and Data Mining. Email:910675024@qq.com.

A Novel Blockchain-based Anonymous Handover Authentication Scheme in Mobile Networks

ChenCheng Hu, Dong Zheng, Rui Guo, AXin Wu, Liang Wang, and ShiYao Gao
(Corresponding author: ChenCheng Hu)

School of Cyberspace Security, Xi'an University of Posts and Telecommunications
National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications
Xi'an 710121, China
(Email: 13772485758@163.com)

(Received Apr. 23, 2019; Revised and Accepted Nov. 16, 2019; First Online Jan. 29, 2020)

Abstract

In the wireless mobile network (WMN), the handover authentication scheme is the key to ensure fast and secure handoff of mobile nodes among several access points. However, it is difficult to design an appropriate handover authentication protocol for the inherent drawbacks in WMN. For example, the resources of the MN are limited and the mobile nodes have low capability in computing. Therefore, the traditional authentication schemes are unsuited to be applied in WMN for the reason that they have low efficiency. To construct a fast handover authentication protocol, in this paper, we design an anonymous handover authentication protocol with high efficiency by considering the distributed storage, collective maintenance and tamper-resistance. Then, the security and efficiency of the proposal is analyzed, and it is concluded that ours uses chameleon hash with blockchain to achieve robust security and high efficiency. Meanwhile, our scheme satisfies the property of user anonymity, conditional privacy protection and robust key agreement as well.

Keywords: Anonymity; Blockchain; Chameleon Hash; Handover Authentication; Wireless Mobile Network

1 Introduction

With the rapid development of wireless mobile network (WMN), various mobile Internet applications have been utilized in the different fields of life. Due to the portability and mobility of mobile devices, the demand for multimedia services by mobile nodes has exploded. Therefore, providing secure and fast real-time services for mobile users will become an inevitable trend in the future. When a service provider offers these real-time services to mobile terminals in the wireless network, the mobile users often need a handoff in the different access points (or base stations) for the limited signal range. In detail, a new connection is established between the mobile terminal and the new access point depending on the handover authentication.

The handover authentication technology realizes the interconnection, intercommunication and mutual confidence between the mobile node and access points, which provides a guarantee for the secure communication in the mobile internet. As shown in Figure 1, a typical handover authentication scenario consists of three entities: The mobile nodes (MNs), access points (APs) and authentication server (AS). For a secure handover authentication, when the MN moves from the current node (AP1) to the new node (AP2), the AP2 needs to authenticate the MN to prevent the illegal users, and the MN also needs to authenticate the AP2 to prevent an attacker from disguising the AP2. In addition, the MN should also establish a session key with the AP2 to protect the security of user's data. Based on this framework, the handover authentication is employed in the mobile communication and real-time services such as 5G, Voice over Internet Phone(VoIP), Video-Phone, mobile TV, Video Conference, and online games. However, the time-delay in these services affects user experience seriously. Thus, it is crucial to reduce the time-delay and the energy consumption in the process of handover authentication that improves the service quality.

To design an efficient and secure handover authentication protocol, there are two issues should be considered. First, for the limited computing power of the MN, the protocol should be lightweight in computation and communication costs. Second, because of the openness of wireless network, the protocol should have robust security to protect the privacy of MN and prevent the system from various attacks. In order to achieve the requirements above, many handover authentication protocols have been put forward in the last several years.

1.1 Related Works

In IEEE 802.11i [3], it proposed a four-way handshake to create a Pairwise Transient Key (PTK) and then distributed a Group Transient Key (GTK) for broadcast

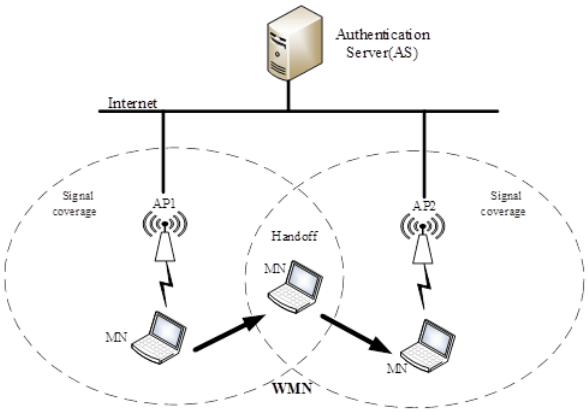


Figure 1: Handover authentication overview

communication. However, the time cost of full handover authentication is unacceptable for real-time traffic in nowadays. In order to improve the performance of handover authentication scheme, the authentication, authorization, and accounting (AAA) based schemes [21,22] were proposed. These schemes use AAA servers to ensure security handovers, and they adopt pre-authentication and proactive key distribution methods to increase the efficiency of authentication. However, they need to establish trust relationship and generate a large amount of authentication traffic between network nodes, which increases the complexity of the entire system. Different from the AAA scheme, there is an alternative protocol without communicating with the AAA server which is called the Security Context Transfer (SCT) schemes [4,27,30] were proposed. The SCT scheme need not to establish communication between AAA and AP. Nevertheless, these solutions are based on the assumption that the APs are mutually trustworthy. Thus, in actual application scenarios, the APs cannot be trusted totally, which brings some security risks to these schemes.

In order to solve the problems in handover authentication above, some works [2, 12–15, 18, 19, 28, 29] were proposed. In [15], an identity-based handover authentication scheme was proposed, which can be implemented only by ID between the MN and AP. This scheme has better efficiency than AAA scheme because there is no need to make communication between the MN and AP, and it reduces the overall system complexity compared to the AAA-based and SCT-based schemes. Unfortunately, since there is a PKG to issue a private key, this solution has the problem of key escrow.

The study in [2] used low-cost functions to achieve security and efficiency, it also used nonce instead of timestamps to avoid the clock synchronization problem. However, Youn *et al.* [29] identified that the scheme of [2] cannot achieve the anonymity under four attack strategies, and it is not efficient in password authentication. Liao and Wang [19] presented a dynamic ID-based remote user authentication scheme for multi-server environment, the scheme of [19] uses simple hash function to enhance efficiency and it can preserve user's anonymity. Later on, Hsiang and Shih [14] showed that the scheme of [19] is vulnerable to insider's attack. He *et al.* [12] proposed a strong user authentication scheme with smart cards for wireless communications.

The scheme of [12] is suitable for the low-power and resource-limited mobile devices since it only performs a symmetric encryption/decryption operation. However, [18] showed that He *et al.*'s scheme is unfairness in key agreement. Then, He *et al.* [13] summarized the basic security requirements of handover authentication protocols and proposed a novel batch verification AHA protocol. However, their implementation calls for complex and time-consuming operation, such as bilinear pairing operations and point multiplication. After that, Xie *et al.* [28] proposed an improved AHA protocol using ECC. Unfortunately, this scheme does not support batch verification and are not suitable for practical applications.

Ramadan *et al.* [23] proposed a user-to-user mutual authentication and key agreement scheme, which is more compatible with the LTE security architecture. In recent years, the idea of proxy signature has been utilized to design handoff authentication schemes [7–9, 20]. The essential idea of these schemes lies in that the authentication server issues its delegation power to the MN, which grants the MN the ability to generate a proxy signature on behalf of the authentication server. Then, the new AP trusts the MN due to the proxy signature on behalf of the authentication server. However, these schemes are vulnerable to various security issues.

Different from the above schemes, the schemes in [5, 10, 11] proposed a handover authentication scheme based on the chameleon hash function. These schemes used the collision of chameleon hash function for authentication to avoid certificate management problems. These protocols are lightweight authentication schemes with high efficiency, but there are still some shortcomings in them, such as key escrow, privacy preservation, redirection attack, high communication and computation overhead.

In the handover authentication, if the legal identity of the MN can be securely broadcast to all APs, the efficiency of authentication process can be greatly improved. In order to make this property can be applied in the handover authentication, we use blockchain technology to achieve our goals. Nowadays, blockchain technology has been applied in many fields [6, 17, 24, 25]. Regarding the application of blockchain in the field of identity authentication. In the literature [6], based on the shortcomings of traditional authentication relying on third-party centers and vulnerable to man-in-the-middle attacks, a blockchain PKI scheme based on privacy protection was proposed. The scheme of [17] describes the concept of blockchain PKI and shows that it has significant advantages over traditional PKI and implements PKI authentication based on Ethereum. However, none of these schemes solves the handover authentication problem.

1.2 Our Contribution

For the above problems in the current handover authentication, we propose a secure anonymous handover authentication scheme based on chameleon hash function and blockchain technology, and design a blockchain certificate model. We summarize our main research contributions as follows:

1) In order to improve the efficiency of authentication process, our scheme uses the distributed and difficult-to-tamper features of the blockchain, and it does not require an extra interaction between the AP node and registration node.

2) We use a blockchain certificate and chameleon hash function to solve the problem of certificate management.

3) To achieve the robust security, we use pseudonyms to provide user anonymity, conditional privacy protection, and updatable key agreement.

4) Finally, we analyze the performance of our scheme and compare its performance with some existing schemes. From the analysis results, our scheme is more efficient than them.

1.3 Organization

The rest of the paper is organized as follows: The Section 2 introduces some preliminaries, such as the knowledge of blockchain techniques, chameleon hash functions and the requirements for an ideal handover authentication. The anonymous handover authentication based on blockchain scheme is presented in Section 3. The security and performance analysis of the related schemes is discussed in Section 4. Finally, Section 5 concludes the paper.

2 Preliminaries

2.1 Blockchain

Blockchain is a new application mode of computer technology such as distributed database, point-to-point transmission, consensus protocols, and encryption algorithm [24]. It records all transaction information occurring on the node. The process is highly transparent and the data is highly secure. The data structure of the blockchain can be described from three levels: Chain, block and transaction. All transactions in the same time period form a block, and the blocks are linked in chronological order to form a blockchain. When several transactions are packaged into a block, data in all nodes can be updated. Each block is composed of a block header and a block body.

Each block header contains the hash value of the previous block, the timestamp, the total hash value of the transaction data (Merkle root). In this way, the chain structure is formed by the interlocking of the hash values of each block. Because of these properties, blockchain has some important characteristics such as tamper resistance, data synchronization, traceability.

2.2 Chameleon Hash Function

The chameleon hash function was first proposed by Krawczyk and Rabin as a one-way hash function with trapdoors [16]. A chameleon hash function is associated with a set of public and private keys, which are also known as trapdoors. For the participants who do not grasp the trapdoor information, it is only a one-way function that is strongly collision-resistance. But for the users who have mastered the trapdoor information, he can easily calculate the collision of the chameleon hash function.

Definition 1. *A chameleon hash function based on the single trapdoor information) [1]*

Generation of public and private key pairs: Let p be a safe prime number of bitlength τ, and satisfies $p = 2q + 1$, where q is a sufficiently large prime number. Let Z_p^* be a group, g is the generator of Z_p^*, g has order q. The user chooses random number $x \in Z_q^*$ as a private key CK_R, and the corresponding public key HK_R is computed as $y = g^x mod p$. Assume that the length of q is λ. Let H be a collision-resistant hash function, mapping arbitrary-length bitstrings to strings of fixed length λ, $H : \{0,1\}^* \rightarrow H : \{0,1\}^\lambda$.

Construction of chameleon hash function: To commit to a message m, and $m \in Z_q^*$. Define the chameleon hash function as: $CHAM - HASH(m, r, s) = r - (y^e g^s mod p) mod q$, the random values (r, s) are choose from $Z_q^* \times Z_q^*$, where $e = H(m||r)$.

Collision finding: Let C be the output of Chameleon hash function $C = CHAM - HASH(m, r, s)$, the user chooses a new random message m' and a random value $k \in Z_q^*$, then computes $r' = C + (g^k mod p) mod q$, $e' = h(m'||r')$, $s' = k - e' x mod q$. Then, we can get the equation:

$$\begin{aligned} C &= CHAM - HASH(m, r, s) \\ &= CHAM - HASH(m', r', s'). \end{aligned}$$

Computational Diffie-Hellman (CDH) problem: Given $x \cdot P, y \cdot P(g^x, g^y)$, the task of the CDH problem is computing $x \cdot y \cdot P(g^{x \cdot y})$, where $x, y \in Z_q^*$ are two unknown numbers.

2.3 Requirements of Handover Authentication

In wireless networks, an ideal handover authentication scheme should satisfy the following requirements:

1) **Mutual authentication:** The AP should authenticate the identity of the MN to determine that the MN is a legitimate user. At the same time, although the AP is trusted, the MN should also authenticate the AP, in order to prevent the attacker from impersonating the AP.

2) **Conditional privacy protection of user:** The identity of the user should not be made public, except for the initial registration node, even if the AP does not know the true identity of the user. However, in some special cases, the AP can send a request to the registration node to obtain the true identity of the MN.

3) **Key agreement:** After mutual authentication is completed, a session key should be established between the MN and the AP to ensure the security of communication afterwards.

4) **Robust security property:** The handover authentication protocols should provide robust security attributes to defend against various attacks on the wireless network (such as eavesdropping, replay attacks, man-in-the-middle attacks, *etc.*).

5) **Perfect forward secrecy:** To protect the security of the session key, a handover authentication protocol should be able to provide perfect forward secrecy, i.e., the adversary cannot extract the session key produced in previous session even he/she gets both private keys of the MN and the AP.

6) **Efficiency:** Since the computing power and storage capacity of mobile nodes in mobile networks are limited, the energy consumption of the authentication process should be as small as possible and the delay should be as low as possible.

The scheme we proposed in this paper satisfies the security attributes required for the above handover authentication.

3 Anonymous Handover Authentication Based on Blockchain

3.1 Blockchain Certificate

In this section, we designed a blockchain certificate based on the X.509 digital certificate and blockchain structure. When the MN completes registration at the registration node AS, the AS will generate a unique blockchain certificate and upload it to the blockchain for the next handover authentication. Our bolckchain certificate structure is shown in Figure 2.

According to [26], the write interface of the blockchain is defined as put(action, data), and the query interface of the blockchain is defined as get(condition). The registration node and the authentication server have the right to write and query. The valid users only have the right to query. The parameter action of the interface written here indicates the user's data processing intent, which can be the state of "issue" or "revoke". The parameter action of the interface written here indicates the user's data processing intent, which can be the state of "issue" or "revoke". Since the blockchain cannot change the data already stored in the blockchain, the issue and revoke here do not directly operate on the data, but record the operation of this data in the blockchain, and then generate a new block and add it to the blockchain. Our handover authentication model is shown in Figure 3.

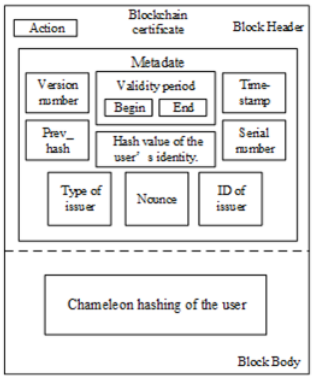


Figure 2: Blockchain certificate

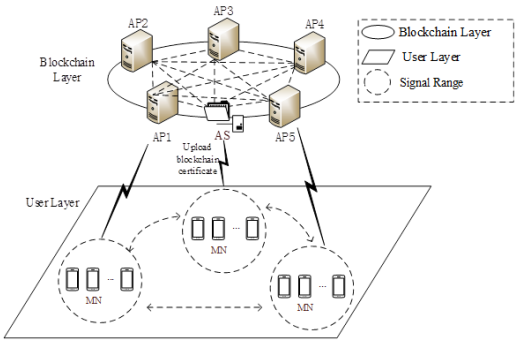


Figure 3: Our handover authentication system model

3.2 Our Handover Authentication Scheme

This section introduces our proposed anonymous handover authentication scheme based on blockchain technology. The notations used in the protocol are shown in Table 1.

Where i represents the different stages of the calculation, the specific process diagram of the protocol flow is shown in Figure 4 and Figure 5:

3.2.1 Initial Authentication Phase

The Initial authentication phase is shown as Figure 4. In the Initial Authentication phase, the mobile node MN needs to register to the AS node with his real identity. If the MN is valid, the AS generates a blockchain certificate

Table 1: The notations used in protocol

Notations	Meanings
$h_1() : \{0,1\}^* \rightarrow Z_q^*$ $h_2() : \{0,1\}^* \rightarrow \{0,1\}^\lambda$	One way and collision-resistance hash function
ID_x	The identity of x
N_i	Random number
$C()_x$	The blockchain certificate of x
T_{Exp}, T_{Curr}	Expiration and current time
(X_x, Y_x)	The chameleon hash trap-door key pair of x
$CHAM(m,r,s)$	Chameleon hash function $CHAM(m,r,s) = r-(Y_x^e g^s modp)modq$
$r(i)_x$	The parameter of Chameleon hash function where $r(i)_x = CHAM(m,r,s)+(g^k modq)$
$s(i)_x$	The parameter of Chameleon hash function where $s(i)_x = k - eX_x modq$
m_x	A message choose by x where $m_x \in Z_q^*$
e	$e$is a required parameter to calculates$(i)_x$,where $e = h_2(m,r)$

of MN, and uploads it to the blockchain. Similarly, for each AP node, their own blockchain certificates are also recorded in the blockchain.

System Parameters: Our scheme specifies two random prime number p and q,q is a big prime number, where $p = 2q+1$.g is selected as a generator of order q from Z_q^*. $h_1() : \{0,1\}^* \rightarrow Z_q^*$ and $h_2() : \{0,1\}^* \rightarrow \{0,1\}^\lambda$ are two safe and collision-resistant hash functions.

1) MN $\rightarrow$ AS: $h_2(ID_{MN})$ The mobile node MN sends the hash value of his identity $h_2(ID_{MN})$ to the AS. Upon receiving the parameter from the MN. The AS verifies the validity of the MN identity based on the stored identity hash value. If the identity is invalid, the MN's access is denied, otherwise, the authentication proceed to the next step.

2) AS $\rightarrow$ MN: PID After confirming the identity of the MN, AS chooses a pseudo-name set $PID = pid_1, pid_2$...in which the elements are unlinkable, and sends it to the MN.

3) MN $\rightarrow$ AS: $CHAM(m_{MN}, r(0)_{MN}, s(0)_{MN})$ After the MN receives the feedback from the AS, the MN randomly chooses $X_{MN} \in Z_q^*$ as his private Chameleon hash key CK_R, and Y_{MN} is public Chameleon hash key. Then the MN chooses the random values $(r(0)_{MN}, s(0)_{MN})$ from $Z_q^* * Z_q^*$,and computes the value of Chameleon hash function:

$$\begin{aligned} C &= CHAM(m_{MN}, r(0)_{MN}, s(0)_{MN}) \\ &= r(0)_{MN} - (Y_{MN}^{e_{MN}} g^{s(0)_{MN}} \bmod p) \bmod q. \end{aligned}$$

Then, the MN sends the value to the AS, where $e_{MN} = h_2(m_{MN}, r(0)_{MN})$.

4) AS $\rightarrow$ Blockchain: $C(0)_{MN}$ Upon receiving the chameleon hash value sent by MN, the AS generates a blockchain certificate of the MN and uploads it to the blockchain.

5) AS $\rightarrow$ MN: T_{EXP} After the AS generates the blockchain certificate of the MN, it returns the time when the certificate expires to the MN. At the same time, the AS opens the query interface of the blockchain to the MN, so that the MN can query the data on the blockchain.

3.2.2 Handover Authentication Phase

The Handover Authentication phase is shown as Figure 5. When the MN arrives at the new AP2, the AP2 needs to

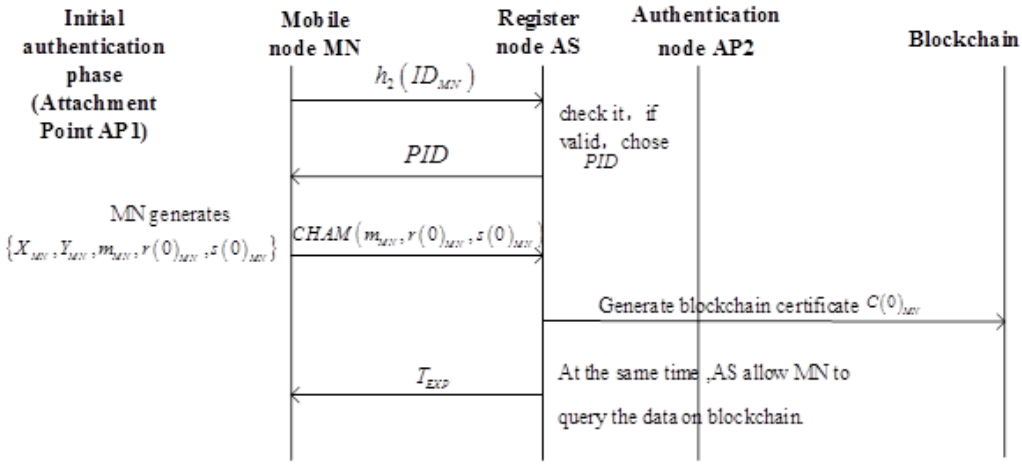


Figure 4: Initial authentication phase

authenticate the legal identity of MN to decide whether to provide services for the MN. Similarly, the MN also needs to authenticates the AP2.

1) MN $\rightarrow$ AP2: $Cert_{MN} \parallel m'_{MN} \parallel r(1)_{MN} \parallel s(1)_{MN} \parallel g^{h_1(pid_j+N_1)} \parallel T_{Curr}$

The MN chooses an unused pid_j from PID, a new random message m'_{MN}, and computes $g^{h_1(pid_j+N_1)}$, then the MN sends $Cert_{MN} \parallel m'_{MN} \parallel r(1)_{MN} \parallel s(1)_{MN} \parallel g^{h_1(pid_j+N_1)} \parallel T_{Curr}$ to AP2, where

$$\begin{aligned} Cert_{MN} &= (pid_j, g^{X_{MN}}) \\ r(1)_{MN} &= CHAM(m_{MN}, r(0)_{MN}, s(0)_{MN}) \\ &\quad +(g^{h_1(pid_j+N_1)} \bmod q) \\ s(1)_{MN} &= h_1(pid_j + N_1) - e'_{MN} X_{MN} \bmod q \\ e'_{MN} &= h_2(m'_{MN}, r(1)_{MN}). \end{aligned}$$

2) AP2 $\leftarrow$ Blockchain: $C(0)_{MN}$ Upon receiving the parameters from the MN, the AP2 uses these parameters to find the MN's blockchain certificate. Then, the AP2 queries the status of the MN's blockchain certificate. If the certificate status is "revoke", the MN's access is denied. Otherwise, the AP2 computes:

$$\begin{aligned} & CHAM(m'_{MN}, r(1)_{MN}, s(1)_{MN}) \\ = \; & r(1)_{MN} - (Y_{MN}^{e'_{MN}} g^{s(1)_{MN}} \bmod p) \bmod q, \end{aligned}$$

and compares with the Chameleon hash value of MN's blockchain certificate $C(0)_{MN}$ to verify whether

$$\begin{aligned} & CHAM(m_{MN}, r(0)_{MN}, s(0)_{MN}) \\ = \; & CHAM(m'_{MN}, r(1)_{MN}, s(1)_{MN}) \end{aligned}$$

is established. If the equation does not hold. The MN is determined to be an illegal user, otherwise, the AP2 authenticates the MN as a legitimate user, and the AP2 sends its own parameters to the MN, so that the MN can authenticates the identity of the AP2.

3) AP2 $\rightarrow$ MN:

$Cert_{AP2} \parallel m'_{AP2} \parallel r(1)_{AP2} \parallel s(1)_{AP2} \parallel g^{k'} \parallel T_{Curr}$

$Cert_{AP2} = (ID_{AP2}, g^{X_{AP2}})$, $g^{X_{AP2}}$ is the public Chameleon hash key of AP2. The AP2 chooses random value $m'_{AP2} \in Z_q^*$ and $k' \in Z_q^*$, and computes $r(1)_{AP2}$, $s(1)_{AP2}$ and $g^{k'}$. Then the AP2 uses the value $g^{X_{MN}}$ and $g^{h_1(pid_j+N_1)}$ from the MN, together with his own parameters, to calculate the session K for communicating with MN. The AP2 can get:

$$K = (g^{h_1(pid_j+N_1)})^{X_{AP2}} (g^{X_{MN}})^{k'}.$$

4) MN $\leftarrow$ Blockchain: $C(0)_{AP2}$

AP2 computes:

$$\begin{aligned} & CHAM(m'_{MN}, r(1)_{MN}, s(1)_{MN}) \\ & = r(1)_{AP2} - (Y_{AP2}^{e'_{AP2}} g^{s(1)_{AP2}} \bmod p) \bmod q \end{aligned}$$

and compares with the Chameleon hash value of AP2's blockchain certificate $C(0)_{AP2}$ to verify whether

$$\begin{aligned} & CHAM(m_{AP2}, r(0)_{AP2}, s(0)_{AP2}) \\ = \; & CHAM(m'_{AP2}, r(1)_{AP2}, s(1)_{AP2}) \end{aligned}$$

to authenticate the AP2. If the authentication is successful, the MN uses the parameters of the AP2 to calculate the session key for communicating with the AP2. The MN can get:

$$K = (g^{k'})^{X_{MN}} (g^{X_{AP2}})^{h_1(pid_j+N_1)}.$$

In the end, the session key shared between MN and AP2 is:

$$K = g^{h_1(pid_j+N_1)X_{AP2}} g^{k'X_{MN}}.$$

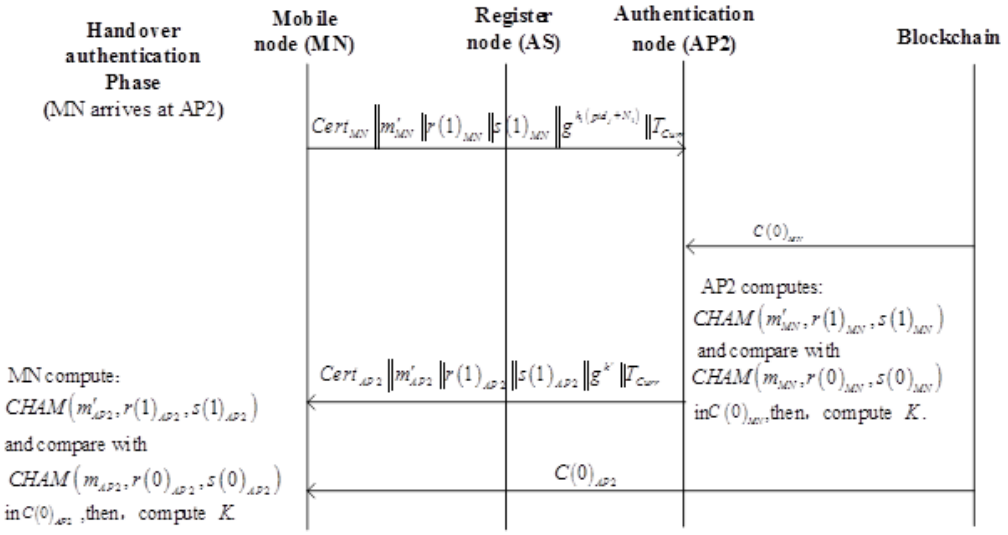


Figure 5: Handover authentication phase

Finally, when the user no longer has a handover authentication request or needs to quit the system, the current AP node generates the user's revocation block and uploads it to the blockchain. In this way, the revocation operation of the users' blockchain certificate is achieved.

4 Security Analysis and Performance Evaluation

4.1 Security Analysis

4.1.1 Mutual Authentication

Our scheme ensures only the legitimate user to access the wireless network. After the MN is successfully registered, the AS allows the MN to query the data on the blockchain. In a handover authentication phase, after the AP authenticates the identity of the MN, the MN calculates $CHAM(m'_{AP_i}, r(1)_{AP_i}, s(1)_{AP_i})$ by using the value$r(1)_{AP_i}$,$s(1)_{AP_i}$ and m'_{AP_i} provided by the AP, and then the MN authenticates the identity of the AP by querying the blockchain certificate $C(0)_{AP_i}$ on the blockchain. According to this, the mutual authentication between MN and AP is completed.

4.1.2 Conditional Privacy Preservation

The MN obtained the pseudonym set PID from the AS during the Initial Authentication phase. The MN uses different *pid* instead of the real identity during different handover authentication phase. Since the elements in the PID are unlinkable to each other, when the MN reaches the new AP and uses a new pseudonym, there is no way for APs to collude to trace the MN according to the connection between the pseudonyms. However, in some special cases, the AP can send a request with the *pid* provided by MN to AS. Upon receiving the request, the AS finds the pseudonym set to which it belongs according to the *pid* provided by the AP, and the true identity of the MN can be found. Based on this, the conditional privacy protection of user can be achieved.

4.1.3 Key Agreement

During the authentication process, the MN uses his own chameleon hash function public key $g^{X_{MN}}$ and the parameter $g^{h_1(pid_j+N_1)}$ generated by the random number and *pid*; the AP2 uses his own chameleon hash function public key $g^{X_{AP2}}$ and the parameter $g^{k'}$ generated by the newly selected element to establish a shared session key. In our construction,K can be shared by the MN and AP2, which satisfies $K = g^{h_1(pid_j+N_1)X_{AP2}} g^{k'X_{MN}}$. Moreover, whenever a new round of handover authentication is performed, the MN must choose a new *pid* to protect its privacy. At the same time, according to key agreement process, since the session key contains the parameter *pid*, the session key is updated with each new round of handover authentication.

4.1.4 Resistance to Replay Attack

In the process of handoff authentication, an adversary may record the message that the MN send to the AP and replay it. Our scheme uses timestamps, random numbers and PID to prevent replay of previous messages. Since the MN updates his own pid_j at every new round of handover authentication. When the MN performs a verification, the timestamp T_{Curr} will be sent, and the parameter $g^{h_1(pid_j+N_1)}$ also contains a random number N_1. Therefore, if an attacker replays a message to try to enter the system, the AP can detect the replay attack regardless of whether he detects the pid_j, the timestamp T_{Curr}, or the value of $g^{h_1(pid_j+N_1)}$. Accordingly, it is worthless for an adversary to replay messages.

4.1.5 Resistance to Man-in-the-Middle attack

In the process of key agreement between MN and AP2, an adversary may replace the parameters with his generated parameters to obtain information. The attack process implemented by the attacker can be described as Figure 6.

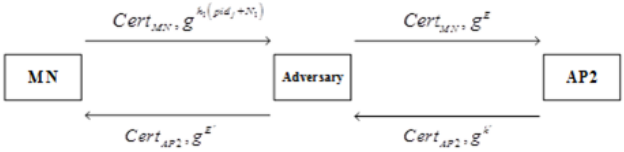


Figure 6: Man-in-the-Middle Attack model

As shown in Figure 6, the adversary E may replace the key parameters to get the session key between the MN and the AP. Upon receiving the parameters, the MN and the AP2 respectively calculate the session key. The key calculated by the MN is $K_1 = g^{h_1(pid_j+N_1)X_{AP2}+E'X_{MN}}$, and the key calculated by the AP2 is $K2 = g^{EX_{AP2}+k'X_{MN}}$. This is not the shared key value they expect, so the MN and the AP2 can not communicates with each other. The adversary will be discovered by the MN and AP2. However, since the attacker does not know the private indexes X_{MN} and X_{AP2} of the MN and the AP2, the attacker cannot calculate either of K_1 or K_2 . It ensures that the MN and the AP2 can be confident that only themselves can calculate the key value shared between them.

4.1.6 Resistance to Passive Eavesdropping Attack

During the process of the handoff authentication, the information that the attacker most desires is the identity of the MN and the the session key K. Firstly, the identity that the MN sends to the AS during the Initial Authentication phase is hashed, which is not available to the eavesdroppers. During the different handover authentication phases, the pid_j in the PID are unlinkable with each other, so the attacker cannot associate the user MN with different *pid* appearing in different handover authentications. As a result, obtaining the identity of the MN is

Table 2: Security comparisons

protocols	Mutual Authentication	Key Agreement	Conditional Privacy Preservation	Replay Attack	Man-in-the-Middle attack	Passive avesdropping Attack	Perfect forward secrecy
[15]	YES	YES	NO	YES	NO	NO	NO
[13]	YES	YES	YES	YES	YES	YES	YES
[28]	YES	YES	NO	YES	YES	YES	YES
[7]	YES	YES	NO	YES	YES	YES	YES
[11]	YES	YES	NO	YES	NO	NO	NO
Ours	YES	YES	YES	YES	YES	YES	YES

difficult for the attacker. Secondly, if the adversary wants to get the private key X_{MN} and X_{AP2}, then the problem of getting X_{MN} and X_{AP2} from $g^{X_{MN}}$ and $g^{X_{AP2}}$ can be reduced to solve the discrete logarithm problem, and it is difficult to be solved. Therefore, our construction can resist against passive eavesdropping.

4.1.7 Perfect Forward Secrecy

To get the session key $K = g^{h_1(pid_j+N_1)X_{AP2}}g^{k'X_{MN}}$ the adversary has to extract $g^{h_1(pid_j+N_1)X_{AP2}}$ from $g^{h_1(pid_j+N_1)}$ and $g^{X_{AP2}}$, the adversary has to address the CDH problem. Because the CDH problem is hard, the proposed protocol can support the perfect forward secrecy.

4.2 Security Comparisons

According to the requirements of handover authentication in section 2.3 and section 4.1, the comparisons of security properties are listed in Table 2.

4.3 Performance Analysis

In this section, the performance of the proposed protocol is analyzed with some existing schemes. Then, we obtained some conclusions about the efficiency of our scheme.

4.3.1 Computation Overhead

The notations we used in this section are shown in Table 3.

Table 3: The notations used in Efficiency calculation

T_E	Time for executing a modular exponentiation in G_T
T_P	Time for executing a bilinear map operation
T_{ECSM}	Time for executing a scalar multiplication operation
T_H	Time for executing a general hash function

Since the AS node only plays the role of registering and uploading the MN's blockchain certificate to the blockchain in our scheme, we only consider the operations and computational cost required by the MN and AP nodes in the efficiency analysis. In order to prove the efficiency of our scheme, we implement the above operations on a Laptop (Lenovo with Intel I5-3320M 2.60GHz processor, 4G bytes memory and the Windows 7 operating system) using the JPBC library. The time cost of the primitive cryptography operations shown in Table 4.

In the Handover authentication phase of our scheme, the MN and AP2 authenticates each other and negotiates a session key. During this phase, the computation cost of MN is: $5T_E+3T_H \approx 1.038ms$, the computation cost of AP is: $5T_E + T_H \approx 0.946ms$. Furthermore, the computation cost among schemes [7, 13, 15, 28] and ours is analyzed in Table 5 and is compared in Figure 9.

Table 4: Time cost of cryptography operations

	T_E	T_P	T_{ECSM}	T_H
Times(ms)	0.18	8.45	2.013	0.046

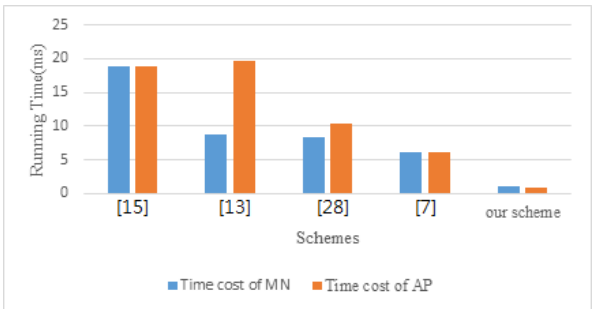


Figure 7: Comparison of the computation cost

And Table 6 shows the energy consumption at the MN (E_{MN}), the energy consumption can be calculated as $E = T_{MN} \times P$, where E is the energy consumption, T_{MN} is the total computation time for handover authentication of MN, and P is the CPU maximum power (35W).

Table 5: Comparison of the computation cost

	MN operations	AP operations
[15]	$T_{ECSM} + 2T_P \approx 18.913ms$	$T_{ECSM} + 2T_P \approx 18.913ms$
[13]	$4T_{ECSM} + 3T_E + 5T_H \approx 8.822ms$	$2T_P + T_{ECSM} + 3T_E + 5T_H \approx 19.683ms$
[28]	$4T_{ECSM} + 5T_H \approx 8.282ms$	$5T_{ECSM} + 5T_H \approx 10.295ms$
[7]	$3T_{ECSM} + 4T_H \approx 6.223ms$	$3T_{ECSM} + 4T_H \approx 6.223ms$
Ours	$5T_E + 3T_H \approx 1.038ms$	$5T_E + T_H \approx 0.946ms$

Table 6: Energy consumption of MN

	[15]	[13]	[28]	[7]	Ours
$E_{MN}(mJ)$	661.955	308.77	289.87	217.805	36.33

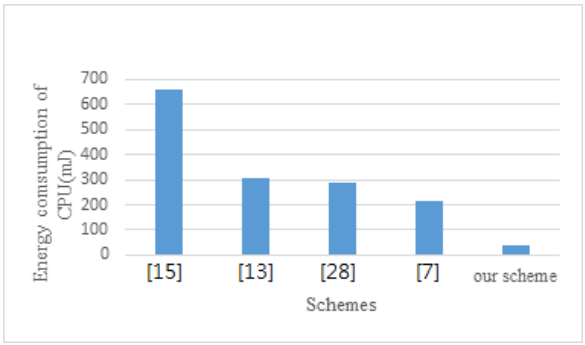


Figure 8: Energy consumption of CPU

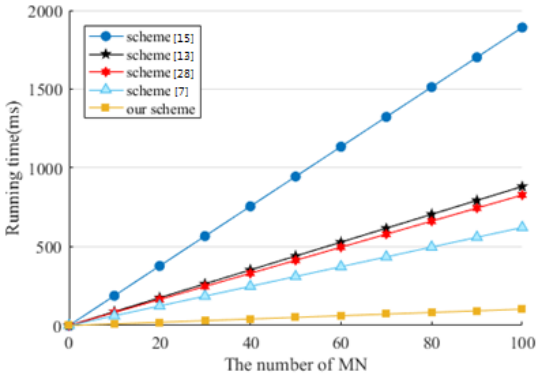


Figure 9: Comparison of handover authentication

4.3.2 Transmission Overhead

For the transmission overhead, it is assumed that the expected authentication message delivery cost between the AP2 and the AAA server is e unit and that between the MN and the AP2 is δ unit, respectively. In our scheme, since we only view the data on the blockchain and do not need it to send us data, we only consider the time consumption between the MN and the AP2. The comparison of the transmission overhead as shown in table 7.

Table 7: Comparison transmission overhead

	[15]	[13]	[9]	Ours
T_{MN-AP2}	3δ	2δ	3δ	2δ
$T_{AP1-AP2}$	0	0	0	0
$T_{AP2-AAA}$	0	0	0	0
T_{tot}	3δ	2δ	3δ	2δ

* T_{MN-AP2} :The transmission cost between the MN and the AP2.
* $T_{AP1-AP2}$:The transmission cost between APs, i.e., AP1 and AP2.
* $T_{AP2-AAA}$:The transmission cost between the AP2 and the AAA server.
* $T_{T_{tot}}$:The total transmission cost.

4.3.3 Communication Overhead

In the proposed handover protocol, two messages correspondence is required for obtaining the handover authentication. In the protocol, the MN transmits $Cert_{MN} \parallel m'_{MN} \parallel r(1)_{MN} \parallel s(1)_{MN} \parallel g^{h_1(pid_j+N_1)} \parallel T_{Curr}$ to the AP2. Hence, the communication overhead incurred from the MN is $(2|p| + 3|q| + l_{id} + l_{time})bits$. The AP2 transmits $Cert_{AP2} \parallel m'_{AP2} \parallel r(1)_{AP2} \parallel s(1)_{AP2} \parallel g^{k'} \parallel T_{Curr}$ to the MN. Hence the generated communication overhead from the AP2 is $(2|p| + 3|q| + l_{id} + l_{time})bits$. According to [13], we know that the proposed protocol increases the communication cost. The reason for the increases is that the MN and the AP2 send $g^{h_1(pid_j+N_1)}$ and $g^{k'}$ to each other for achieving the perfect forward secrecy. It is worthy to achieve the important security attribute at the cost of increasing computation cost only.

Based on the above comparative analysis, it can be seen that our scheme consumes less computation. Our scheme also provides user anonymity and conditional privacy protection. Therefore, our scheme is more suitable for practical application scenarios.

5 Conclusion

In the handover authentication of wireless networks, secure and efficient handover authentication has been the focus of widespread attention. In this paper, we propose a anonymous handover authentication scheme based on chameleon hash function and blockchain technology. The main idea of our scheme is to generate a blockchain certificate for the user by a registration node AS. When the handover authentication occurs, the AP compares the chameleon hash value provided by the user with the blockchain certificate to verify the legal identity of the user. Our scheme provides anonymity and conditional privacy protection.The AP can request the true identity of the MN from the AS when some accidents occurs during the handover authentication phase. When the user no longer has a handover authentication request or needs to log off, the current AP node generates the user's revocation block and uploads it to the blockchain. In this way, the revocation operation of the users' blockchain certificate is achieved. Finally, when the MN performs a new handover authentication to choose a new *pid*, the session key for the secure communication with AP is also updated at the same time. After the analysis of performance, our scheme has the ideal efficiency.

Acknowledgments

This work was supported by the Natural Science Foundation of China under Grants 61802303 and 61772418, the Innovation Ability Support Program in Shaanxi Province of China under Grant 2017KJXX-47, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2016JM6033 and 2018JZ6001.

References

[1] G. Ateniese and B. D. Medeiros, "On the key exposure problem in chameleon hashes," in *International Conference on Security in Communication Networks*, pp. 165–179, Sep. 2004.

[2] C. C. Chang, C. Y. Lee, and Y. C. Chiu, "Enhanced authentication scheme with anonymity for roaming service in global mobility networks," *Computer Communications*, vol. 32, no. 4, pp. 611–618, 2009.

[3] C. Chaplin, E. Qi, H. Ptasinski, J. Walker, and S. Li, *802.11i overview, IEEE.802.11–04/0123r1*, 2005. (`http://www.drizzle.com/~aboba/IEEE`)

[4] J. Choi and S. Jung, "A secure and efficient handover authentication based on lightweight diffie-hellman on mobile node in FMIPv6," *IEICE Transactions on Communications*, vol. 91, no. 2, pp. 605–608, 2008.

[5] J. Choi and S. Jung, "A handover authentication using credentials based on chameleon hashing," *IEEE communications letters*, vol. 14, no. 1, pp. 54–56, 2009.

[6] C. Fromknecht, D. Velicanu, and S. Yakoubov, "Certcoin: A namecoin based decentralized authentication system 6.857 class project," *Unpublished Class Project*, 2014. (`https://courses.csail.mit.edu/6.857/2014/files/19-fromknecht-velicann-yakoubov-certcoin.pdf`)

[7] S. Gupta, B. L. Parne, and N. S. Chaudhari, "A lightweight handover authentication protocol based on proxy signature for wireless networks," in *The 14th IEEE India Council International Conference (INDICON'17)*, pp. 1–6, July 2017.

[8] S. Gupta, B. L. Parne, and N. S. Chaudhari, "A proxy signature based efficient and robust handover AKA protocol for LTE/LTE-A networks," *Wireless Personal Communications*, vol. 103, no. 3, pp. 2317–2352, 2018.

[9] S. Gupta, B. L. Parne, and N. S. Chaudhari, "Pseh: A provably secure and efficient handover AKA protocol in LTE/LTE-A network," *Peer-to-Peer Networking and Applications*, vol. 12, no. 4, pp. 989–1011, 2018.

[10] S. Gupta, B. L. Parne, and N. S. Chaudhari, "An efficient handover aka protocol for wireless network using chameleon hash function," in *The 4th International Conference on Recent Advances in Information Technology (RAIT'18)*, pp. 1–7, June 2018.

[11] Q. Han, Y. Zhang, X. Chen, H. Li, and J. Quan, "Efficient and robust Identity-Based handoff authentication in wireless networks," in *International Conference on Network and System Security*, pp. 180–191, 2012.

[12] D. He, M. Ma, Y. Zhang, C. Chen, and J. Bu, "A strong user authentication scheme with smart cards for wireless communications," *Computer Communications*, vol. 34, no. 3, pp. 367–374, 2011.

[13] D. He, D. Wang, Q. Xie, and K. F. Chen, "Anonymous handover authentication protocol for mobile wireless networks with conditional privacy preservation," *Science China Information Sciences*, vol. 60, no. 5, pp. 052104, 2017.

[14] H. C. Hsiang and W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.

[15] Y. Kim, W. Ren, J. Jo, M. Yang, Y. Jiang, and J. Zheng, "SFRIC: A secure fast roaming scheme in wireless lan using ID-based cryptography," in *IEEE International Conference on Communications*, pp. 1570–1575, June 2007.

[16] H. Krawczyk and T. Rabin, *Chameleon Hashing and Signatures*, Aug. 2000. US Patent 6,108,783.

[17] K. Lewison and F. Corella, *Backing Rich Credentials with a Blockchain PKI*, 2016. (`https://pomcor.com/techreports/BlockchainPKI.pdf`)

[18] C. T. Li and C. C. Lee, "A novel user authentication and privacy preserving scheme with smart cards for

wireless communications," *Mathematical and Computer Modelling*, vol. 55, no. 1-2, pp. 35–44, 2012.

[19] Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.

[20] C. Ma, K. Xue, and P. Hong, "A proxy signature based re-authentication scheme for secure fast handoff in wireless mesh networks," *International Journal Network Security*, vol. 15, no. 2, pp. 122–132, 2013.

[21] A. Mishra, M. H. Shin, and W. A. Arbaugh, "Proactive key distribution using neighbor graphs," *IEEE Wireless Communication Magazine*, vol. 11, no. 1, pp. 26–36, 2004.

[22] S. Pack and Y. Choi, "Fast handoff scheme based on mobility prediction in public wireless lan systems," *IEE Proceedings of Communications*, vol. 151, no. 5, pp. 489–495, 2004.

[23] M. Ramadan, F. Li, C. X. Xu, and A. Mohamed, "User-to-user mutual authentication and key agreement scheme for LTE cellular system," *International Journal Network Security*, vol. 18, no. 4, pp. 769–781, 2016.

[24] N. Satoshi, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. (`https://bitcoin.org/bitcoin.pdf`)

[25] C. Tang and L. Gao, "Multi-parties key agreement protocol in block chain," *Netinfo Security (in Chinese)*, vol. 12, no. 9, pp. 19, 2017.

[26] W. T. Tsai, L. Yu, and R. Wang, "Blockchain application development techniques," *Journal of Software (in Chinese)*, vol. 28, no. 6, pp. 1474–1487, 2017.

[27] H. Wang and A. R. Prasad, "Fast authentication for inter-domain handover," in *Telecommunications and Networking (ICT'04)*, pp. 973–982, Aug. 2004.

[28] Y. Xie, L. Wu, N. Kumar, and J. Shen, "Analysis and improvement of a privacy-aware handover authentication scheme for wireless network," *Wireless Personal Communications*, vol. 93, no. 2, pp. 523–541, 2017.

[29] T. Y. Youn, Y. H. Park, and J. Lim, "Weaknesses in an anonymous authentication scheme for roaming service in global mobility networks," *IEEE Communications Letters*, vol. 13, no. 7, pp. 471–473, 2009.

[30] C. Zhang, R. Lu, P. Ho, and A. Chen, "A location privacy preserving authentication scheme in vehicular networks," in *IEEE Wireless Communications and Networking Conference*, Apr. 2008. DOI: 10.1109/WCNC.2008.447.

Biography

ChenCheng Hu received the B.Eng. degree from the Xi'an University of Posts and Telecommunications, in 2016, where he is currently pursuing the M.S. degree. He is doing research at the National Engineering Laboratory for Wireless Security. His current research interests include blockchain technology, user authentication, and information security.

Dong Zheng received the Ph.D. degree from Xidian University, in 1999. He joined the School of Information Security Engineering, Shanghai Jiao Tong University. He is currently a Professor with the Xi'an University of Posts and Telecommunications, China. His research interests include information theory, cryptography, and information security. He is also a Senior Member of the Chinese Association for Cryptologic Research and a member of the Chinese Communication Society.

Rui Guo received the Ph.D. degree from the State Key Laboratory of Networking and Switch Technology, Beijing University of Posts and Telecommunications, China, in 2014. He is currently a Lecturer with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications. His current research interests include attribute-based cryptograph, cloud computing, and blockchain technology.

Axin Wu received B.S. degree from Zhengzhou University of Light Industry in 2016. Since 2016, he is currently in M.Eng program in Xi'an University of Post and Telecommunications, Xi'an, China. His research interests include cloud security and wireless network security.

Liang Wang received the B.S. degree from the Institute of Information Technology, GUET, in 2016. He is currently pursuing the M.S. degree with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, China. His research interests include anonymous authentication, vehicular ad hoc networks, and blockchain.

ShiYao Gao received the B.S. degree from the Xi'an University of Posts and Telecommunications, in 2017. She is currently pursuing the M.S. degree with the Xi'an University of Posts and Telecommunications, China. She is doing research at the National Engineering Laboratory for Wireless Security. Her research interests include blockchain technology, electronic voting, and information security.

New Publicly Verifiable Data Deletion Supporting Efficient Tracking for Cloud Storage

Changsong Yang[1], Xiaoling Tao[1], and Qiyu Chen[2]
(Corresponding author: Changsong Yang and Xiaoling Tao)
School of Computer Science and Information Security, Guilin University of Electronic Technology[1]
No. 1 Jinji Road, Guilin, China, 541004
College of Information and Computer Engineering, Northeast Forestry University[2]
No. 26 HeXing Road, Harbin, Heilongjiang, China
(Email: csyang02@163.com and txl@guet.deu.cn)
(Received Apr. 26, 2019; Revised and Accepted Nov. 16, 2019; First Online Feb. 15, 2020)

Abstract

With the rapid development of cloud storage, an increasing number of data owners are willing to outsource their data to cloud server to greatly reduce local storage overhead. However, in cloud storage, the ownership of the outsourced data is disconnected from the management, which makes the outsourced data deletion become a crucial security challenge: the cloud server might reserve the data maliciously for economic interests, and return wrong deletion results to cheat the data owners. To solve this problem, we design a novel outsourced data deletion scheme. If the cloud server does not execute deletion command honestly, the data owner can detect the dishonest data reservation by checking the returned deletion evidence. Additionally, we adopt Merkle hash tree to achieve public verifiability in outsourced data deletion without requiring any trusted third party. Meanwhile, the proposed scheme is able to achieve efficient data leakage source tracking to prevent the data owner and the cloud server from slandering each other. Finally, we prove that our scheme can satisfy the desired security requirements.

Keywords: Cloud Storage; Data Deletion; Efficient Tracking; Merkle Hash Tree; Public Verifiability

1 Introduction

Cloud computing, a newly-developing and promising computing paradigm, can connect large-scale computing resources, network resources and storage resources together through the Internet [7]. Thanks to the rapid development of computer software and hardware technology, cloud computing can utilize its plenty of resources to provide many attractive services, for instance, data storage and sharing service, outsourcing service, verifiable databases service, and so on. These services have been widely applied by the public, especially for cloud storage service. The cloud storage service provider can offer on-demand data storage service to the tenants [9]. By employing the high-quality cloud storage service, all the resource-constraint data owners could upload their personal data to remote cloud server for saving heavy local storage overhead. Because of the attractive advantages, an increasing number of data owners, including individuals and corporations prefer to embrace cloud storage service.

Despite a number of advantages, cloud storage service inescapably suffers from a few novel security challenges [3,6]. First of all, the outsourced file might contain some data owner's privacy information, which should be kept secret. Therefore, data confidentiality has become a particularly austere security challenge for cloud storage. Generally speaking, the traditional encryption technique can be seen as a solution to this issue. However, it can only offer a partial solution because it is very difficult to execute significative operations over the ciphertext. The fully homomorphic encryption algorithm seems a potential solution, but the existing protocols are not efficient and practical. Secondly, both the data owner and the cloud server may be dishonest. Both of them might expose the data maliciously to slander each other. Therefore, how to precisely trace the data leakage source is a challenge which requires to be solved. Last but not least, the data owner cannot execute any operation over the outsourced data directly because he will lose the direct control over the data. All the operations over the outsourced data, such as data deletion operation, might be executed by the cloud server. However, the cloud server might not remove the data sincerely for economic interests. Hence, how to securely remove outsoutced data is also a security threat.

Although plenty of solutions have been proposed to realize data deletion, there are still some security challenges in processing the outsourced data deletion. Firstly, plenty of existing data deletion schemes reach deletion by

overwriting the physical disks [8, 13, 18]. To be specific, they use some random data to overwrite the data which needs to be deleted. To make the data deletion operation more secure, some researchers suggest that the disk should be overwritten more than one times. Although overwriting the disk can theoretically solve the problem of data deletion, it still has some inherent limitations. On the one hand, overwriting the physical medium isn't efficient for real applications. Especially for distributed storage system, it is very difficult to overwrite every disk which maintains the data copy. On the other hand, there may be some physical remanence of the overwritten data left on the disk. The attacker who equips with advanced microsoping tools can recover the overwritten data with the physical remanence [5]. Therefore, it is desired to improve the efficiency and the security of the data deletion schemes.

Boneh and Lipton [2] firstly utilized cryptography technique to delete the data instead of protecting them, which can make the data deletion operation more secure and efficient, and resulting in plenty of follow-up schemes [10, 14, 16]. To be specific, these schemes should firstly use encryption key to encrypt the data before storing. Then they destroy a very short decryption key to make a large amount of related ciphertext unavailable, and return an erasure outcome to the data owner. This approach can efficiently delete the digital data. However, a lot of existing cryptography-based data deletion schemes are not able to achieve verifiability. That is, the data owner must trust the returned deletion result since he cannot verify it conveniently. However, the storage server may reserve the data maliciously and return a wrong outcome to cheat the data owner. Therefore, the requirement of deletion result verifiability should be introduced into the data deletion schemes.

Last but not least, some schemes have been put forwarded to provide the data owner with the ability to verify the data deletion result conveniently [11, 20, 22, 25, 26]. They delete the outsourced data and then return a related data deletion proof. The data owner can check the deletion result by verifying the returned deletion proof. However, these schemes all assume that the data owner is honestly. If the deleted data is later discovered, the cloud server is deemed to reserve the data maliciously, and the data owner should be entitled to compensation. However, there are some dishonest data owners in real-world, and they may expose the data maliciously to slander the cloud server to obtain compensation. All the existing schemes are not able to judge the data leakage source under the dishonest data owner and cloud server model because both of the two entities can obtain the same data backup. Therefore, we should offer the ability to track the data leakage source if the data is exposed.

Although various schemes have been put forward to deal with the problem of data deletion, most of them have a few deficiencies. First of all, in a lot of existing solutions, the data owner must trust the returned outcome since he cannot verify it. However, the cloud server might reserve the data backup maliciously and return a wrong outcome to cheat the data owner, but the data owner cannot detect the malicious behavior. Secondly, although some existing deletion schemes provide the data owner with the ability to verify the deletion outcome, they cannot achieve traceability. When the data is leaked, they cannot trace the data leakage source. To the best of our knowledge, it seems that there is not research work on publicly verifiable data deletion scheme that supports data leakage source tracking. Hence, we design a new scheme to delete the outsourced data and trace the data leakage source.

1.1 Our Contributions

In this paper, we put forward a novel publicly verifiable data deletion scheme for cloud storage, which can simultaneously achieve data leakage source tracking. The main contributions of our proposed scheme are as follow:

- We put forward a new Merkle hash tree-based publicly verifiable outsourced data deletion scheme, which can simultaneously achieve data leakage source tracking. To be specific, after executing data deletion operation, the cloud server can utilize Merkle hash tree to generate a deletion proof. If the cloud server reserves the data backup maliciously, the data owner can detect the cloud server's dishonest behavior by verifying the proof.

- Our proposed scheme can satisfy the property of traceability, which is different from the previous solutions. That is, if the data is leaked, the proposed scheme can trace the data leakage source, which can prevent data owner and cloud server from exposing the data maliciously to slander each other. Additionally, the proposed scheme is also very efficient in computation and communication.

This paper is an extension of our previous work that was presented at SICBS [24]. In the following, we show the main differences between this paper and the conference version. Firstly, we demonstrate the related work more detailedly in Section 1. Secondly, we put forward a more detailed scheme, and add the high description of the proposed scheme in Section 4. We also identify the main security properties for the proposed scheme in Section 3.3, and we prove that our new scheme can satisfy these design goals in Section 5.1. Finally, we will add the experimental simulation and performance comparison between our scheme and two previous schemes in Section 5.3.

1.2 Related Work

Data deletion has been studied for a long time. Perlman *et al.* [12] utilized a trusted third party (TTP) to solve the data deletion problem. First of all, they encrypt the data with a data key, then the TTP further encrypts the data key with a control key. When the data owner will not need the file anymore, the TTP will make

the data key unavailable by destroying the control key, thus the corresponding ciphertext cannot be decrypted anymore. In 2010, Tang *et al.* [15] designed a practical and implementable file assured deletion (FADE) system. They firstly use a data key to encrypt the file. After that the data key will be encrypted with a control key which associated with a policy. Besides, one or multiple TTPs maintain the policies together. Finally, when they want to delete the file, they can remove the related policy, and instruct the TTP to delete the corresponding key.

To offer the data owner the ability to verify the deletion outcome, Hao *et al.* [5] presented a novel data deletion protocol. In their protocol, they store the private key in a trusted platform module's protected memory. Then they reach data deletion by destroying the private key and finally return a signature as an evidence. In 2016, Luo *et al.* [8] presented a permutation-based data deletion scheme. They suppose that the cloud server could merely maintain the latest version of the data. Additionally, all the backups will be consistent when they are updated. Then they reach deletion by updating them with random data. Finally, the data owner is able to verify the deletion outcome through a challenge-response protocol. In 2018, Yang *et al.* [21] used blockchain to design a novel scheme to achieve publicly verifiably data deletion. In their scheme, they utilize blockchain to reach public verifiability without requiring any TTP, which is quite different from a lot of the previous schemes. After executing deletion operation, the cloud server can generate a deletion proof, which will be published on the blockchain. Finally, the data owner can check the deletion result by verifying the proof.

Xue *et al.* [19] put forward a verifiable data deletion method, which could also achieve provable data transfer and data integrity verification. Their scheme gives the data owner the ability to move the outsourced data between two different clouds. Moreover, the data owner is able to check the transferred data integrity on the target cloud through provable data possession (PDP) protocol. Then the original cloud deletes the transferred data blocks, and utilizes Rank-based Merkle hash tree (RMHT) to generate a deletion proof. Wang *et al.* [17] presented a similar method in 2018. Recently, Yang *et al.* [23] put forward a novel verifiable outsoutced data transfer and deletion scheme. In their scheme, the data owner is able to migrate the outsourced data between two different clouds, and then delete the transferred data from the original cloud server. Additionally, they utilize the primitive of vector commitment (VC) to realize public verifiability without requiring any TTP.

1.3 Organization

The remainder of this paper is organized as follows: We describe the preliminary of Merkle hash tree in Section 2. In Section 3, we describe the problem statement in detail. To be specific, we firstly formalize the system model of our novel scheme. Then we present the main security challenges. Finally, the security goals will be identified. In Section 4, we put forward our new publicly verifiable data deletion scheme in detail. A brief analysis of the proposed scheme, and the performance evaluation are presented in Section 5. Finally, we will conclude the proposed scheme in Section 6.

2 Merkle Hash Tree

Merkle hash tree (MHT), a specific binary tree, is always used to authenticate digital data [1, 4]. By using MHT, the communication and computation overhead during the verification process will be decreased greatly. In MHT, each leaf node maintains a hash value of the data block which needs to be authenticated, and every internal node keeps a hash value of the concatenation of its two children. For example, if we want to authenticate data set $D = \{d_1, d_2, d_3, d_4\}$, the MHT is illustrated as Figure 1, $h_{2\_i} = H(d_i)$, where $i \in [1, 4]$, $h_{1\_2} = H(h_{2\_1}||h_{2\_2})$, H is a secure one-way hash function. Finally, the public key signature technique is used to sign the root node.

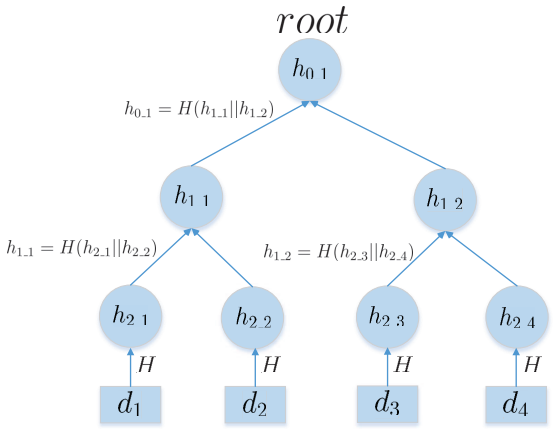


Figure 1: An example of MHT

The verifier can verify any subset of D by utilizing the verification object Φ, which is a set of all sibling nodes on the path from the authenticated leaf node to the root node. For instance, to verify d_3, Φ contains h_{1_1} and h_{2_4}. The verifier computes $h'_{0\_1} = H(h_{1\_1}||H(h_{2\_3}||h_{2\_4}))$ firstly. Then he checks that whether $h'_{0\_1} = h_{0\_1}$ holds, and verifies the validity of the signature. If both the verifications pass, it means that d_3 is valid; otherwise, it means that d_3 has been tampered with maliciously.

3 Problem Statement

3.1 System Model

In the following, we formalize the system model of our new scheme, which involves three entities: a data owner O, a cloud server S and a trusted agency TA, as illustrated in Figure 2.

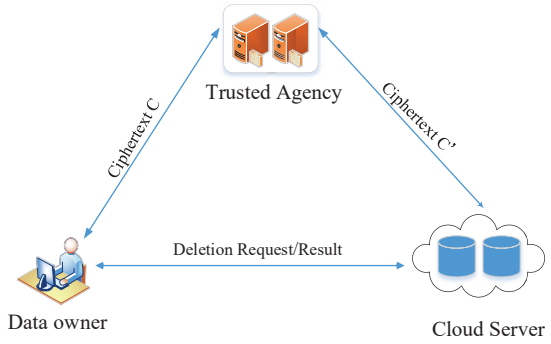


Figure 2: The system model

- **The data owner O.** It is a resource-constraint entry, who prefers to outsource his personal data to the cloud server for saving local storage overhead. When O will not need the data anymore, he sends a data deletion command to the cloud server to delete the outsourced data permanently. Finally, O can check the deletion result by verifying the returned deletion proof.
- **The cloud server S.** It refers to an entry which has large-scale computing resources, network resources and storage resources, thus, can maintain a large amount of data for resource-constraint O. When O will not need the outsourced data anymore, S will be required to delete the outsourced data from the disk. After that S may generate a deletion evidence for O to check the deletion outcome.
- **The trusted agency TA.** It is a trusted third party, which is absolutely righteous. That is to say, TA will never collude with O or S to cheat the other maliciously. TA always behaves honestly and righteously, therefore, both S and O fully trust TA unconditionally.

3.2 Security Threats

We assume that the cloud server S is "semi-honest-but-curious". As a result, S may not follow the protocol honestly for economic interests. Besides, the attackers, such as hackers or malicious users may try their best to access the outsourced data illegally. Therefore, we seriously consider the following two types of attacks: the internal attacks and the external attacks. The internal attacks are launched by the internal attackers, such as the dishonest cloud administrators, who would try to dig some sensitive information from the outsourced data. Furthermore, the dishonest S may share the outsourced data with others for financial incentives. The external attackers (*e.g.*, hackers and illegal users) might try to access the outsourced file and dig privacy data. Therefore, we should consider the following three security challenges.

- **Data privacy disclosure**. Privacy disclosure is a very common and serious security threat in cloud storage. On the one hand, the internal attackers are so curious that they may dig some sensitive information from the outsourced data. Moreover, the selfish cloud server moves the outsourced data to other subcontractors for saving storage overhead, or shares them with some other corporators for economic interests. On the other hand, the external attackers may try their best to access the file to find some privacy data.
- **Data corruption**. The outsourced data may be polluted for the following reasons. First of all, the manager performs erroneous operations, software or hardware malfunctions all may cause data loss. Secondly, the external attackers (*e.g.*, hackers) may modify or delete the data arbitrarily. Last but not least, when the data owner downloads the file, the cloud server sends part of the data for saving bandwidth, or delivers some unrelated data to cheat the data owner.
- **Malicious data reservation**. When the data owner will not need the data anymore, he will send a deletion command to the cloud server to delete the data permanently. However, the selfish cloud server might not execute the data deletion operation honestly for the following factors: (1) it needs some computational cost to delete the data from the physical medium; (2) the cloud server might try to reserve the data to dig some privacy data.

3.3 Design Goals

In our new scheme, we aim to achieve publicly verifiable data deletion in cloud storage. Meanwhile, when the data is leaked, we can trace the data leakage source efficiently. Therefore, our scheme should realize the following four goals.

- **Data confidentiality**. To ensure the outsourced data confidentiality, it should prevent the attackers from accessing the data directly because the data may contain some privacy information. That is, it is necessary to use cryptography algorithm to encrypt the file before uploading it to the cloud server. Moreover, the corresponding decryption key should be maintained secretly.
- **Data integrity**. To prevent the outsourced data from being polluted, the data owner should be given the ability to verify the data integrity and availability. If the outsourced data has been polluted, the data owner should be able to detect the malicious manipulation.
- **Verifiable data deletion**. To make the cloud server delete the data from physical medium sincerely, the data owner should be given the ability to verify the deletion result. If the cloud server reserves the data dishonestly, the data owner can detect the malicious data reservation by verifying the returned proof.

- **Accountable traceability**. To prevent the data owner and the cloud server from slandering each other, it should satisfy the property of accountable traceability. To be specific, we can trace the data leakage source precisely when the data is leaked. Additionally, upon the data owner and the cloud server executing some operations, they cannot deny their performances anymore.

4 Our Construction

4.1 High Description

In this paper, we study the problem of publicly verifiable cloud data deletion with efficient tracking under the commercial mode, which is very similar to schemes [5,19]. In our system model, there is a trust problem between the cloud server S and the data owner O: both of them might not fully believe each other. On the one hand, O might not believe that S will execute data deletion operation honestly. On the other hand, S thinks that O may reserve the data deliberately, and expose them to slander that S did not sincerely delete the data after deletion. Now plenty of data deletion methods have been proposed, but all of them assume that O is honest, thus O will not slander S maliciously. However, this assumption is not realistic. Therefore, we propose a novel scheme, which aims to make the data deletion publicly verifiable and the data leakage source traceable.

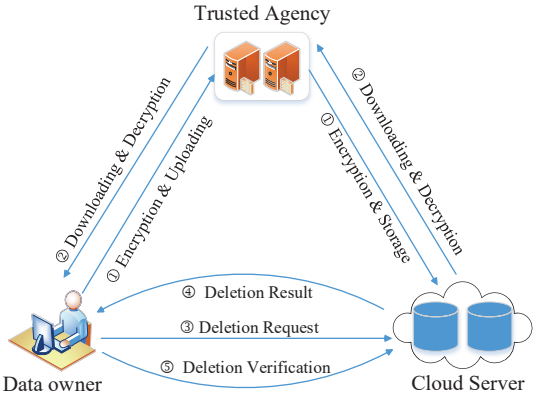


Figure 3: The main processes of our scheme

Our proposed scheme not only can achieve publicly verifiable data deletion but also can realize efficient data leakage source tracking, and Figure 3 describes the main steps of the proposed scheme. First of all, O encrypts the file to protect the privacy, and then sends the ephemeral ciphertext to TA. Then TA further encrypts the received ephemeral ciphertext and sends the final ciphertext to S. After that TA verifies the storage result, and O deletes the local backup. When O wants the outsourced file, he downloads the corresponding ciphertext and decrypts it to obtain the plaintext. If O will not need the file anymore, he is willing to send a deletion command to delete the data from S. Upon receiving the deletion request, S deletes the related data and returns a deletion evidence to O. Finally, O can check the data deletion result by verifying the proof. In our scheme, we utilize MHT to realize public verifiability, and the verification process does not need any TTP.

4.2 The Concrete Construction

In this part, we put forward our new scheme in detail. First of all, we define a few notations. Before embracing cloud storage service, the data owner O must pass the authentication of cloud server S. For simplicity, we assume that O has been authenticated and become a legal user of S. Then O can set a unique identity id, which is maintained by O and TA secretly. Besides, we suppose that O, S, TA respectively has a ECDSA key pair (pk_o, sk_o), (pk_s, sk_s) and (pk_t, sk_t). $H_1(\cdot)$ and $H_2(\cdot)$ are two secure hash functions. Furthermore, we assume that every file is named with a secret and unique name, and the name is so secure that it can resist brute-force attack. Without loss of generality, we can assume that O wants to upload file F to S, and n_f is the name of F.

- *Encrypt*. To guarantee data confidentiality, the data owner O should encrypt the file F before uploading, and the detailed processes are as follow.
  - First of all, O encrypts the file F: $C_o = Enc_{k_o}(F)$, where $k_o = H_1(sk_o||id||n_f)$, and Enc is an IND-CPA secure symmetric encryption algorithm. Then O computes a file tag $tag_f = H_1(n_f)$ and a hash value $h_o = H_1(C_o||id||tag_f)$. Finally, O sends the ephemeral ciphertext C_{f_o} to TA, where $C_{f_o} = (C_o, tag_f, h_o)$.
  - Upon receiving C_{f_o}, TA verifies that whether the equation $h_o = H_1(C_o||id||tag_f)$ holds. If $h_o \neq H_1(C_o||id||tag_f)$, TA aborts and returns failure; otherwise, TA further encrypts C_{f_o}: $C_t = Enc_{k_t}(C_{f_o})$, where $k_t = H_1(sk_t||id||tag_f)$. Then TA computes a hash value $h_t = H_1(C_t||id||tag_f)$. Finally, TA sends the final ciphertext C_{f_t} to S, where $C_{f_t} = (C_t, tag_f, h_t)$.
- *StoreCheck*. On receipt of C_{f_t}, the cloud server S maintains the data and returns a storage proof. For simplicity, assume that m files are stored in MHT.
  - Upon receipt of C_{f_t}, S stores the data in the leaf node of MHT. Here we can take $m = 8$ for example, and C_{f_t} is stored in the leaf node 6, as illustrated in Figure 4. Then S computes a signature on the hash value of the root node: $sig_r = Sign_{sk_s}(h_{0\_1})$, where $Sign$ is a ECDSA signature generation algorithm. Finally, S returns storage proof $\lambda = (sig_r, \Phi)$ to TA, where Φ is the verification object $(h_{1\_1}, h_{2\_4}, h_{3\_5})$.

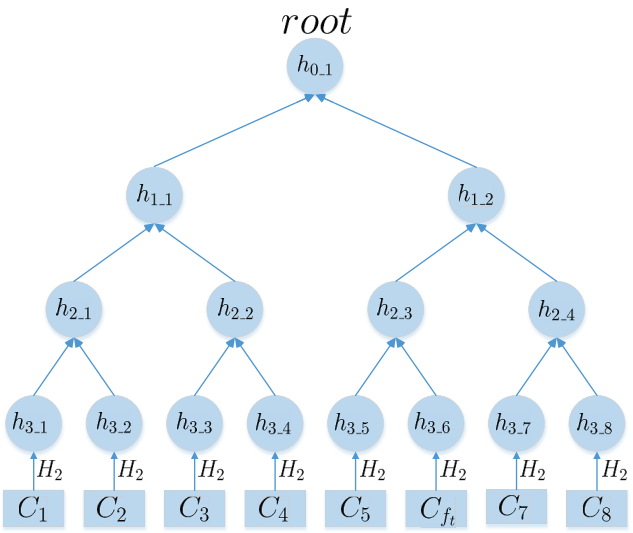


Figure 4: MHT for storage proof

– On receiving λ, TA checks that whether S stores the file and generates the storage proof honestly. To be specific, TA computes the following equations:

$$h'_{3\_6} = H_2(C_{f_t});$$
$$h'_{2\_3} = H_2(h_{3\_5}||h'_{3\_6});$$
$$h'_{1\_2} = H_2(h'_{2\_3}||h_{2\_4});$$
$$h'_{0\_1} = H_2(h_{1\_1}||h'_{1\_2});$$

Then TA checks that whether the equation $h'_{0\_1} = h_{0\_1}$ holds. If $h'_{0\_1} \neq h_{0\_1}$, TA quits and outputs failure; otherwise, TA verifies that if sig_r is a valid signature on h_{0_1}. If sig_r is not valid, TA quits and outputs failure; otherwise, TA trusts that S stores the data honestly, and sends λ to O, then O deletes the local backup.

- *Decrypt*. When data owner O needs the file F, he should download the ciphertext from the cloud server S and decrypt it to obtain the plaintext.

  – In order to download the ciphertext, O needs to generate a download request R_d. First of all, O computes a signature $sig_d = Sign_{sk_o}(download||tag_f||T_d)$, where T_d is a timestamp. Then O sends download request R_d to S, where $R_d = (download, tag_f, T_d, sig_d)$. Upon receiving R_d, S checks the validity of R_d through signature verification. If R_d is not valid, S quits and outputs failure; otherwise, S sends $C_{f_t} = (C_t, tag_f, h_t)$ and R_d to TA.

  – Upon receipt of C_{f_t} and R_d, TA firstly verifies the validity of R_d. If R_d is not valid, TA quits and outputs failure; otherwise, TA checks that if $h_t = H_1(C_t||id||tag_f)$ holds. If $h_t \neq H_1(C_t||id||tag_f)$, TA quits and outputs failure; otherwise, TA decrypts C_t to obtain the ephemeral ciphertext $C_{f_o} = Dec_{k_t}(C_t)$, where Dec represents a traditional symmetric decryption algorithm, and $k_t = H_1(sk_t||id||tag_f)$. Finally, TA sends $C_{f_o} = (C_o, tag_f, h_o)$ to O.

  – Upon receiving $C_{f_o} = (C_o, tag_f, h_o)$, O checks that if the equation $h_o = H_1(C_o||id||tag_f)$ holds. If $h_o \neq H_1(C_o||id||tag_f)$, O quits and outputs failure; otherwise, O executes decryption operation to obtain the corresponding plaintext $F = Dec_{k_o}(C_o)$, where $k_o = H_1(sk_o||id||n_f)$.

- *Delete*. When data owner O will not need the file F anymore, he wants to permanently delete the file from the cloud server S.

  – To delete the outsourced data, O needs to generate a deletion request R_e. Firstly, O computes a signature $sig_e = Sign_{sk_o}(delete||tag_f||T_e)$, where T_e is a timestamp. Then O generates a deletion request $R_e = (delete, tag_f, T_e, sig_e)$, and sends R_e to S.

  – Upon receipt of R_e, S firstly checks the validity of R_e. If R_e is not valid, S aborts and returns failure; otherwise, S deletes the data and computes a signature $sig_s = Sign_{sk_s}(delete||tag_f||T_e||R_e)$. Then S utilizes sig_s to replace C_{f_t} to re-construct the MHT, as illustrated in Figure 5. Finally, S computes a new signature on the hash value of the new root node $sig^*_r = Sign_{sk_s}(h^*_{0\_1})$, and returns a deletion proof τ to O, where $\tau = (R_e, sig_s, sig^*_r, \Phi)$, and $\Phi = (h_{1\_1}, h_{2\_4}, h_{3\_5})$.

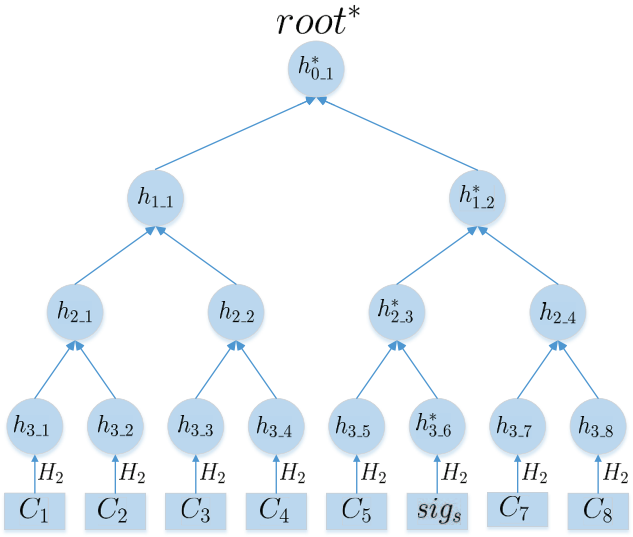


Figure 5: The MHT for deletion proofs

- *DelCheck*. After receiving τ, the data owner O can check the deletion result by verifying τ.

  – Firstly, O checks that whether the signature sig_s is valid. If sig_s is not valid, O quits and

outputs failure; otherwise, O computes the following equations:

$$h^{*'}_{3\_6} = H_2(sig_s);$$
$$h^{*'}_{2\_3} = H_2(h_{3\_5}||h^{*'}_{3\_6});$$
$$h^{*'}_{1\_2} = H_2(h^{*'}_{2\_3}||h_{2\_4});$$
$$h^{*'}_{0\_1} = H_2(h_{1\_1}||h^{*'}_{1\_2});$$

– Then O checks that whether the equation $h^{*'}_{0\_1} = h^{*}_{0\_1}$ holds. If $h*'_{0\_1} \neq h^{*}_{0\_1}$, O quits and outputs failure; otherwise, O verifies that whether the signature sig^{*}_{r} is a valid signature on $h^{*}_{0\_1}$. If sig^{*}_{r} cannot pass the verification, O quits and outputs failure; otherwise, O can trust that the deletion proof τ is valid. If the deleted data discovered again, based on the evidence, O should be entitled to compensation.

5 Scheme Analysis and Implementation

In the following section, we give a brief analysis of the proposed scheme. Firstly, we analyze the proposed scheme's security properties in detail. Secondly, we present the comparison among our scheme and some previous schemes in theory. Finally, we evaluate the performance through the simulation experiments.

5.1 Security Analysis

In this part, we analyze the security of our proposed scheme, including the data confidentiality, data integrity, public verifiability, traceability and non-repudiation.

5.1.1 Data Confidentiality

To protect the sensitive information, the data owner uses IND-CPA secure AES algorithm to encrypt the file before uploading. Additionally, the data owner keeps the encryption key and decryption key secret. That is, any attacker cannot acquire the encryption key and decryption key maliciously. In other word, the malicious attacker cannot obtain any plaintext information from the ciphertext. Hence, our novel scheme can reach data confidentiality.

5.1.2 Data Integrity

Our proposed scheme is able to ensure the outsourced data integrity. In the decryption process, the cloud server S will firstly send the final ciphertext C_{f_t} and the download request R_d to the trusted agency TA, where $C_{f_t} = (C_t, tag_f, h_t)$ and $R_d = (download, tag_f, T_d, sig_d)$. On receipt of C_{f_t} and R_d, TA will check R_d and C_{f_t} before decrypting. If R_d is not valid, it means that O does not require to download the file, and TA aborts; otherwise, TA checks C_{f_t}. To be specific, TA checks that whether the equation $h_t \stackrel{?}{=} H_1(C_t||id||tag_f)$ holds. The id is kept secret by O and TA. Therefore, S cannot forge a new C'_t to make equation $h_t = H_1(C'_t||id||tag_f)$ hold. That is, if and only if C_t is intact can the verifications pass, and TA decrypts C_t to obtain C_{f_o}. Therefore, TA always can detect the malicious operation if S falsifies C_t.

Besides, upon receiving $C_{f_o} = (C_o, tag_f, h_o)$ from TA, O will check it before executing decryption operation. To be specific, O verifies that whether the equation $h_o \stackrel{?}{=} H_1(C_o||id||tag_f)$ holds. The attacker cannot falsify a new C'_o to make equation $h'_o = H_1(C'_o||id||tag_f)$ hold because the id is maintained secretly by O and TA. That is, the attacker cannot forge C'_o to cheat O successfully. Therefore, if and only if C_{f_o} is intact can the verification pass.

As the analysis described above, the proposed scheme can achieve data integrity.

5.1.3 Public Verifiability

Our new scheme can reach publicly verifiable data deletion in cloud storage. After executing data deletion operation, the cloud server S generates a deletion proof τ to prove that he has performed data deletion honestly. Note that $\tau = (R_e, sig_s, sig^{*}_{r}, \Phi)$, where $\Phi = (h_{1\_1}, h_{2\_4}, h_{3\_5})$. Then anyone who given τ (called verifier) can check the data deletion result by verifying the evidence τ. Firstly, the verifier checks the validity of the deletion request R_e. If R_e is not valid, the verifier aborts and returns failure; otherwise, the verifier checks the validity of the signature sig_s. If sig_s is not valid, the verifier aborts and returns failure; otherwise, the verifier utilizes $H_2(sig_s)$ and Φ to re-compute $h^{*}_{0\_1}$. Finally, the verifier checks that whether the signature sig^{*}_{r} is a valid signature on $h^{*}_{0\_1}$. If and only if all verifications pass will the verifier trust that the deletion proof τ is valid. Note that the verification phases do not involve any private information, and any verifier can check the deletion outcome. Therefore, we think that our scheme is able to realize the property of public verifiability.

5.1.4 Traceability

The proposed scheme can trace the data leakage source precisely when the data is leaked. In our scheme, the data owner O owns the plaintext F and the ephemeral ciphertext C_{f_o}. The trusted agency TA further encrypts the ephemeral ciphertext C_{f_o} to obtain the final ciphertext C_{f_t}. Then the cloud server S maintains the final ciphertext C_{f_t}. Besides, O cannot access to the final ciphertext C_{f_t}, and S cannot access to the plaintext F and the ephemeral ciphertext C_{f_o}. That is, only TA and O can obtain the ephemeral ciphertext C_{f_o}, and only TA and S can obtain the final ciphertext C_{f_t}. Note that TA is absolutely impartial, and it will never collude with O (or S) to cheat S (or O). Hence, on the one hand, the data leakage source must be O if C_{f_o} is exposed. That is,

Table 1: Functionality comparison among three schemes

Scheme	Scheme [5]	Scheme [22]	Our Scheme
Trusted Third Party	Yes	Yes	Yes
Public Verifiability	Yes	Yes	Yes
Data Confidentiality	Yes	Yes	Yes
Data Integrity	No	Yes	Yes
Non-repudiation	No	Yes	Yes
Traceability	No	No	Yes

Table 2: Computational complexity comparison

Scheme	Scheme [5]	Scheme [22]	Our Scheme
(Encrypt)	$2\mathcal{E}+4\mathcal{H}$	$1\mathcal{E}+m\mathcal{H}$	$2\mathcal{E}+6\mathcal{H}$
(Decrypt)	$1\mathcal{E}+1\mathcal{D}+3\mathcal{H}$	-	$1\mathcal{S}+1\mathcal{V}+2\mathcal{D}+4\mathcal{H}$
(Store)	-	$1\mathcal{S}+1\mathcal{V}+46m\mathcal{H}$	$1\mathcal{S}+1\mathcal{V}+(2^{n+1}+n)\mathcal{H}$
(Delete)	$1\mathcal{S}$	$2\mathcal{S}+1\mathcal{V}+23\mathcal{H}$	$3\mathcal{S}+1\mathcal{V}+(n+1)\mathcal{H}$
(DelCheck)	$1\mathcal{V}$	$1\mathcal{V}+20\mathcal{H}$	$2\mathcal{V}+(n+1)\mathcal{H}$

the dishonest O cannot reserve C_{f_o} and then expose it to successfully slander that S did not delete the data honestly. On the other hand, the data leakage source must be S if C_{f_t} is leaked. Therefore, if S reserves C_{f_t} maliciously and resulting in data leakage, S cannot deny his dishonest data reservation. That is, the proposed scheme can reach the data leakage source traceability.

5.1.5 Non-repudiation

In our scheme, we assume that both the data owner O and the cloud server S may deny their behaviors thus slander the other. Without loss of generality, we analyze the non-repudiation when S is malicious and O is dishonest, respectively.

Case 1: Malicious cloud server S. The malicious cloud server S may slander the data owner O. First of all, the malicious S deletes the outsourced data arbitrarily to save storage overhead, and then slanders that he performed the data deletion operation as O's command. For this scenario, O can require S to present the data deletion request $R_e = (delete, tag_f, T_e, sig_e)$, where sig_e is a signature generated by O with private key sk_o. On the one hand, O had never generated and sent R_e to S to delete the data. On the other hand, S cannot forge a valid R_e since S does not has the private key sk_o. Therefore, S cannot present R_e to prove that he deleted the outsourced data as O's command. Secondly, S reserves the data dishonestly and slanders that O had not required him to remove the outsourced data. Here, O can demonstrate the data deletion proof $\tau = (R_e, sig_s, sig_r^*, \Phi)$, where sig_s is a signature generated by S with private key sk_s. The signature sig_s can be seen as a proof, which can prove that O has required S to delete the data, and S has responded to this request. That is, the dishonest S cannot successfully slander O.

Case 2: Dishonest data owner O. The dishonest data owner O denies his behavior and slanders the cloud server S maliciously. Firstly, O had asked S to delete the data, and S had done it honestly. However, O declares that he had never asked S to delete the data, and slanders that S deleted the data arbitrarily. Here, S can show the deletion request $R_e = (delete, tag_f, T_e, sig_e)$, which contains a signature sig_e generated by O. No one else can forge a valid signature sig_e to further forge a valid deletion request R_e. Therefore, R_e can be seen as an evidence which can prove that O had required S to delete the data. Secondly, O had never required S to delete the data. Nevertheless, O declares that he had required S to delete the file and S did not do it sincerely. In this case, S can require O to show the deletion proof $\tau = (R_e, sig_s, sig_r^*, \Phi)$ which generated by S. However, S did not generate τ at all. In addition, O cannot forge a valid τ. Therefore, O cannot slander S successfully.

5.2 Comparison

In the following, we compare the functionality and computational complexity among our new scheme and two previous schemes [5, 22] in theory, then Table 1 and Table 2 demonstrate the comparison results, respectively.

From Table 1 we can have the following findings. Firstly, all of these three schemes need to introduce a trusted third party to achieve publicly verifiable data deletion. Secondly, all of them can guarantee data confidentiality, which can protect the sensitive information that contained in the outsourced file. Thirdly, Hao *et*

al. scheme [5] cannot satisfy the data integrity and non-repudiation, which is different from our scheme and Yang *et al.* scheme [22]. Last but not least, only our new scheme is able to achieve traceability, which can prevent data owner and cloud server from slandering each other. Overall, our new scheme is more attractive than the other two schemes.

Then we compare the performance of the three schemes in theory, and the results are listed as time complexities in Table 2. For simplicity, we use symbols $\mathcal{E}$ and $\mathcal{D}$ to represent a symmetry encryption and decryption, respectively. Moreover, we denote by $\mathcal{H}$ a hash computation, $\mathcal{S}$ a signature generation operation, and $\mathcal{V}$ a signature verification operation. Meanwhile, we assume that the MHT has $m = 2^n$ leaf nodes and the number of data blocks in scheme [22] is m. Finally, we ignore the other overhead, such as multiplication and communication overhead.

5.3 Performance Evaluation

In this part, we simulate our proposed scheme and two previous schemes [5, 22], then provide the performance evaluation. The related algorithms are implemented with PBC library and the OpenSSL library on an Unix machine, which equips with Intel(R) Core(TM) i5-6200U processors running at 2.4 GHz and 8 GB main memory. For simplicity, we simulate all the entities on this Linux machine and ignore the communication overhead.

The outsourced file always contains some sensitive information, which should be kept secret. Therefore, the data owner needs to use secure encryption algorithm to encrypt the file before uploading. We increase the size of the file from 0.125 MB to 1 MB with a step for 0.125 MB, and the number of data blocks in Yang *et al.* scheme [22] is fixed in 1024. Then the approximate time cost is shown in Figure 6. We can find that although the time overhead will increase with the size of the file, the encryption operation is one-time. Moreover, our proposed scheme and Hao *et al.* scheme [5] cost almost the same time cost to encrypt the same size of file. Meanwhile, Yang *et al.* scheme [22] needs more time cost than the other two schemes. Hence, we think our proposed scheme is efficient to encrypt file.

After uploading the file to the cloud server, the data owner wants to check that whether the cloud server maintains the data honestly. The main computation comes from storage proof generation and storage result verification. In our scheme, the cloud server needs to compute $(2m - 1)$ hash values and a ECDSA signature to generate storage proof, where $m = 2^n$. Then the data owner needs to execute $(n + 1)$ hash calculations and a signature verification operation to verify the storage result. In Yang *et al.* scheme [22], the computation consists of a signature generation and a signature verification, and $46m$ hash computations. We increase the number n from 1 to 8 with a step for 1, and then Figure 7 presents the efficiency comparison. From Figure 7 we can realize that although the computational overhead increases with

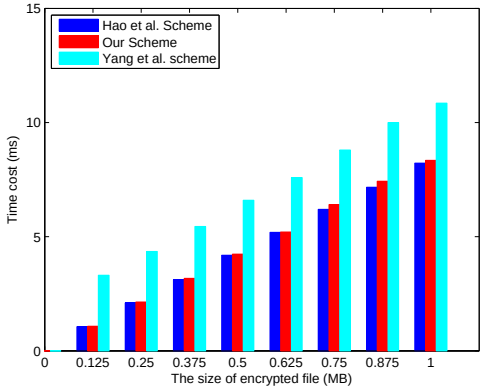


Figure 6: Time cost of encryption

n, the growth rate of our scheme is relatively lower than that of Yang *et al.* scheme [22]. Meanwhile, our proposed scheme costs less time overhead. Therefore, our proposed scheme is more efficient than Yang *et al.* scheme [22].

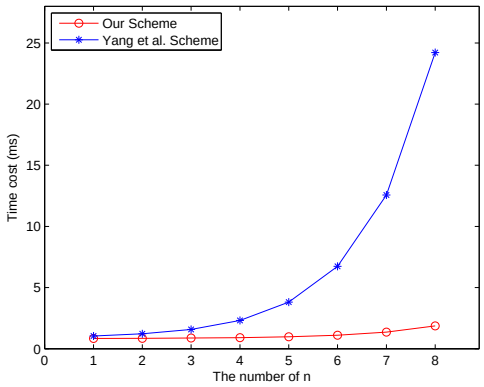


Figure 7: Time cost of storage

For saving storage overhead, the data owner does not maintain any local data backup after outsourcing the file to the remote cloud server. Therefore, when the data owner needs the file, he needs to download the corresponding ciphertext from the cloud server, and then decrypts it to obtain the plaintext. We increase the ciphertext from 0.125 MB to 1 MB with a step for 0.125 MB, and test the approximate time overhead. Then the efficiency comparison of decryption process between the two schemes is shown in Figure 8. From Figure 8 we can realize that the time cost will increase with the size of the decrypted ciphertext, and our scheme's growth rate is lower than that of Hao *et al.* scheme [5]. Moreover, although our scheme will cost a little more time when the ciphertext is less than 0.75 MB, the extra overhead is small and acceptable. Further, when the ciphertext is larger than 0.75 MB, the time cost of our scheme is less than that of Hao *et al.* scheme [5]. In real application, the ciphertext is often larger than 0.75 MB. Therefore, we can think that our scheme is more efficient than Hao *et al.* scheme [5] to

decrypt same size of ciphertext.

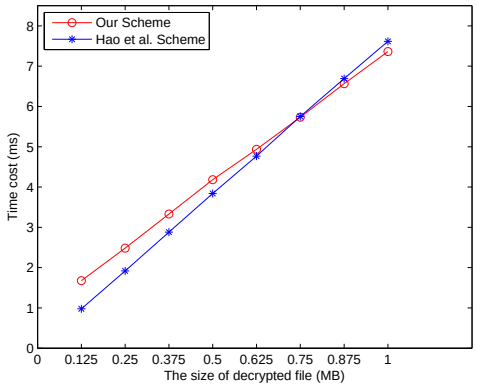


Figure 8: Time cost of decryption

When the data owner will not need the outsourced file anymore, he wants to permanently delete the data from the cloud server. To delete the outsourced file, our scheme needs to execute three signature generation operations and a signature verification computation. Moreover, our scheme also needs to perform $(n+1)$ hash calculations to update the MHT. However, Hao *et al.* scheme [5] merely needs to generate a signature. Meanwhile, Yang *et al.* scheme [22] needs to compute 23 hash values, generate two signatures and perform a signature verification operation. Then the efficiency comparison among the three schemes is shown in Figure 9. We can easily find that the time overhead of our scheme will increase with the number n, but the growth rate is very low. However, the time overhead of the other two schemes is almost constant. Moreover, although our scheme needs a little more time to delete a file, the time cost is very small. For example, when $n = 40$, the time cost is about 1.5 microseconds. Meanwhile, note that the deletion operation is one-time. Therefore, our scheme is still very efficient.

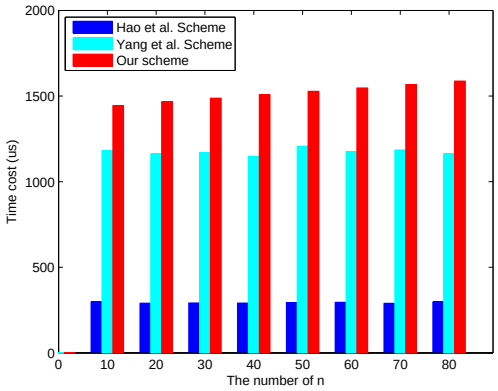


Figure 9: Time cost of deletion

After deletion, the data owner is able to check the data deletion result through verifying the returned deletion evidence. Our scheme needs to execute two signature verification operations and $(n+1)$ hash computations to verify the deletion result. However, Hao *et al.* scheme [5] only needs to verify the validity of a signature. Meanwhile, Yang *et al.* scheme [22] needs to verify a signature and compute 20 hash values. Then the time cost comparison among the tree schemes is demonstrated in Figure 10. We can easily find that the time cost of our scheme will increase with n, while the time of the other two schemes is almost constant. However, the growth rate of our scheme is very low. Additionally, although our scheme costs a litter more time than the other two schemes, the time cost is very small. Meanwhile, note that the verification operation can be finished off-line. Therefore, it will not affect the overall efficiency.

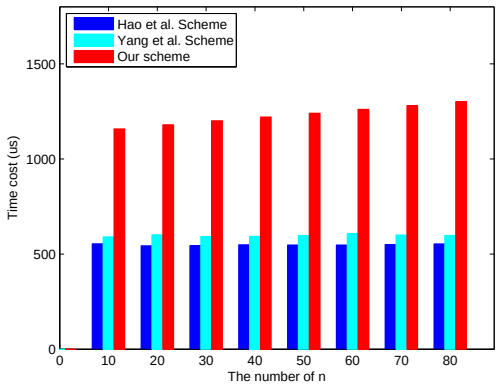


Figure 10: Time cost of deletion verification

6 Conclusions

In this paper, we put forward a novel MHT-based publicly verifiable outsourced data deletion scheme, which also supports efficient data leakage source tracking. In cloud storage, both the data owner and the cloud server might think the other is dishonest. In our new scheme, we use the cryptographic primitive of MHT to deal with this trust problem. To be more specific, the cloud server should use MHT to compute an evidence after deletion. If the cloud server reserves the data maliciously, the data owner is able to easily detect the dishonest data reservation by verifying the deletion proof. In addition, our novel scheme can satisfy the property of data leakage source traceability, which can prevent the data owner and cloud server from exposing the data to slander the other. In the future, we will study how to reach data deletion and leakage source traceability without requiring any TTP.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61962015), the Science and Technology Program of Guangxi (No.AB17195045), and the Natural Science Foundation

of Guangxi (No.2016GXNSFAA380098). Moreover, the authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] G. S. Aujla, R. Chaudhary, N. Kumar, A. K. Das, and J. J. Rodrigues, "Secsva: Secure storage, verification, and auditing of big data in the cloud environment," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 78–85, 2018.

[2] D. Boneh and R. Lipton, "A revocable backup system," in *Proceedings of the 6th Conference on USENIX Security Symposium*, pp. 91–96, pp. 22-25, 1996.

[3] K. Brindha and N. Jeyanthi, "Securing portable document format file using extended visual cryptography to protect cloud data storage," *International Journal of Network Security*, vol. 19, no. 5, pp. 684–693, 2017.

[4] N. Garg and S. Bawa, "Rits-mht: Relative indexed and time stamped merkle hash tree based data auditing protocol for cloud computing," *Journal of Network and Computer Applications*, vol. 84, pp. 1–13, 2017.

[5] F. Hao, D. Clarke, and A. F. Zorzo, "Deleting secret data with public verifiability," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 6, pp. 617–629, 2016.

[6] W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[7] G. Lou and Z. Cai, "A cloud computing oriented neural network for resource demands and management scheduling," *International Journal of Network Security*, vol. 21, no. 3, pp. 477–482, 2019.

[8] Y. Luo, M. Xu, S. Fu, and D. Wang, "Enabling assured deletion in the cloud storage by overwriting," in *Proceedings of the 4th ACM International Workshop on Security in Cloud Computing*, pp. 17–23, May 2016.

[9] H. Ma, X. Han, T. Peng, and L. Zhang, "Secure and efficient cloud data deduplication supporting dynamic data public auditing," *International Journal of Network Security*, vol. 20, no. 6, pp. 1074–1084, 2018.

[10] Z. Mo, Y. Qiao, and S. Chen, "Two-party fine-grained assured deletion of outsourced data in cloud systems," in *Proceedings of the IEEE 34th International Conference on Distributed Computing Systems (ICDCS'14)*, pp. 308–317, July 2014.

[11] J. Ni, X. Lin, K. Zhang, Y. Yu, and X. X. Shen, "Secure outsourced data transfer with integrity verification in cloud storage," in *Proceedings of the IEEE/CIC International Conference on Communications in China (ICCC'16)*, pp. 1–6, July 2016.

[12] R. Perlman, "File system design with assured delete," in *Proceedings of the Third IEEE International Security in Storage Workshop (SISW'05)*, pp. 83–88, Dec. 2005.

[13] J. Reardon, D. Basin, and S. Capkun, "Sok: Secure data deletion," in *Proceedings of the IEEE Symposium on Security and Privacy (SP'13)*, pp. 301–315, May 2013.

[14] J. Reardon, S. Capkun, and D. A. Basin, "Data node encrypted file system: Efficient secure deletion for flash memory," in *Proceedings of the 21st USENIX Conference on Security Symposium*, pp. 333–348, Aug. 2012.

[15] Y. Tang, P. P. Lee, J. C. Lui, and R. Perlman, "Fade: Secure overlay cloud storage with file assured deletion," in *Proceedings of the 6th Iternational ICST Conference on Security and Privacy in Communication Systems*, pp. 380–397, Sep. 2010.

[16] W. L. Wang, Y. H. Chang, P. C. Huang, C. H. Tu, , H. W. Wei, and W. K. Shih, "Relay-based key management to support secure deletion for resource-constrained flash-memory storage devices," in *Proceedings of the 21st Asia and South Pacific Design Automation Conference (ASP-DAC'16)*, pp. 444–449, Jan. 2016.

[17] Y. Wang, X. Tao, J. Ni, and Y. Yu, "Data integrity checking with reliable data transfer for secure cloud storage," *International Journal of Web and Grid Services*, vol. 14, no. 1, pp. 106–121, 2018.

[18] M. Y. C. Wei, L. M. Grupp, F. E. Spada, and S. Swanson, "Reliably erasing data from flash-based solid state drives," in *Proceedings of the 9th USENIX Conference on File and Storage Technologies (FAST'11)*, pp. 105–117, Feb. 2011.

[19] L. Xue, J. Ni, Y. Li, and J. Shen, "Provable data transfer from provable data possession and deletion in cloud storage," *Computer Standards & Interfaces*, vol. 54, pp. 46–54, 2017.

[20] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, "Efficient attribute-based encryption with attribute revocation for assured data deletion," *Information Sciences*, vol. 479, pp. 640–650, 2019.

[21] C. Yang, X. Chen, and Y. Xiang, "Blockchain-based publicly verifiable data deletion scheme for cloud storage," *Journal of Network and Computer Applications*, vol. 103, pp. 185–193, 2018.

[22] C. Yang, X. Tao, F. Zhao, and Y. Wang, "A new outsourced data deletion scheme with public verifiability," in *Proceedings of the 14th International Conference on Wireless Algorithms, Systems, and Applications (WASA'19)*, pp. 631–638, June 2019.

[23] C. Yang, J. Wang, X. Tao, and C. Chen, "Publicly verifiable data transfer and deletion scheme for cloud storage," in *Proceedings of the 20th International Conference on Information and Communications Security (ICICS'18)*, pp. 445–458, Oct. 2018.

[24] C. Ynag and X. Tao, "New publicly verifiable cloud data deletion scheme with efficient tracking," in *Pro-*

ceedings of the 2th International Conference on Security with Intelligent Computing and Big-data Services (SICBS'18), pp. 1–14, Dec.2018.

[25] Y. Yu, J. Ni, W. Wu, and Y. Wang, "Provable data possession supporting secure data transfer for cloud storage," in *Proceedings of the 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA'15)*, pp. 38–42, Nov. 2015.

[26] Y. Yu, L. Xue, Y. Li, X. Du, M. Guizani, and B. Yang, "Assured data deletion with fine-grained access control for fog-based industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4538–4547, 2018.

Biography

Changsong Yang biography. Changsong Yang received the B.S. degree in school of telecommunications engineering from Xidian university, Xi'an, China in 2014 and became a master degree candidate of this university in 2014. And now he is currently undergoing his Ph.D degree at the School of Cyber Engineering at Xidian university. His research interests are network security, cloud computing, verifiable data deletion and blockchian.

Xiaoling Tao biography. Xiaoling Tao received the M.S. degree in computer application technology from Guilin University of Electronic Technology, Guilin, China in 2008 and became the faculty member of this university in 2000. She is currently a professor at the school of computer science and information security, Guilin University of Electronic Technology. And now she is studing for her Ph.D degree in information and communication engineering at Guilin University of Electronic Technology. Her current research interests include network security, cloud computing and computational intelligence.

Qiyu Chen biography. Qiyu Chen is a research scholar in the College of Information and Computer Engineering, Northeast Forestry University.

Guide for Authors
International Journal of Network Security

IJNS will be committed to the timely publication of very high-quality, peer-reviewed, original papers that advance the state-of-the art and applications of network security. Topics will include, but not be limited to, the following: Biometric Security, Communications and Networks Security, Cryptography, Database Security, Electronic Commerce Security, Multimedia Security, System Security, etc.

1. Submission Procedure

Authors are strongly encouraged to submit their papers electronically by using online manuscript submission at http://ijns.jalaxy.com.tw/.

2. General

Articles must be written in good English. Submission of an article implies that the work described has not been published previously, that it is not under consideration for publication elsewhere. It will not be published elsewhere in the same form, in English or in any other language, without the written consent of the Publisher.

2.1 Length Limitation:

All papers should be concisely written and be no longer than 30 double-spaced pages (12-point font, approximately 26 lines/page) including figures.

2.2 Title page

The title page should contain the article title, author(s) names and affiliations, address, an abstract not exceeding 100 words, and a list of three to five keywords.

2.3 Corresponding author

Clearly indicate who is willing to handle correspondence at all stages of refereeing and publication. Ensure that telephone and fax numbers (with country and area code) are provided in addition to the e-mail address and the complete postal address.

2.4 References

References should be listed alphabetically, in the same way as follows:
For a paper in a journal: M. S. Hwang, C. C. Chang, and K. F. Hwang, ``An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445--446, 2002.

For a book: Dorothy E. R. Denning, *Cryptography and Data Security*. Massachusetts: Addison-Wesley, 1982.

For a paper in a proceeding: M. S. Hwang, C. C. Lee, and Y. L. Tang, ``Two simple batch verifying multiple digital signatures," in *The Third International Conference on Information and Communication Security (ICICS2001)*, pp. 13--16, Xian, China, 2001.

In text, references should be indicated by [number].

Subscription Information

Individual subscriptions to IJNS are available at the annual rate of US$ 200.00 or NT 7,000 (Taiwan). The rate is US$1000.00 or NT 30,000 (Taiwan) for institutional subscriptions. Price includes surface postage, packing and handling charges worldwide. Please make your payment payable to "Jalaxy Technique Co., LTD." For detailed information, please refer to http://ijns.jalaxy.com.tw or Email to ijns.publishing@gmail.com.