IJNS
International Journal of Network Security

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

Vol. 25, No. 5 (September 2023)

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: 25, No: 5 (September 1, 2023)

International Journal of Network Security

Security Situation Prediction Method of Industrial Control System Based on Self-Attention and GRU Neural Network

Peng-Shou Xie, Shuai Wang, Ying-Wen Zhao, Wan-Jun Shao, Wei Li, and Tao Feng

(Corresponding author: Shuai Wang)

School of Computer and Communications, Lanzhou University of Technology
No. 36 Peng Jia-ping Road, Lanzhou, Gansu 730050, China
Email:627858493@qq.com

(Received Aug. 14, 2022; Revised and Accepted June 20, 2023; First Online July 1, 2023)

Abstract

Given the current industrial control system security situation prediction method accuracy is insufficient, the model is challenging to build. Aiming at the above problems, this paper proposes a security situation prediction method for industrial control systems based on the self-attention mechanism and GRU neural network. Firstly, the self-attention mechanism generates attention weight combined with security situation data. Secondly, The data with attention weight is input to the gated cyclic unit to mine the correlation between the safety data of the industrial control system. Finally, the trained model is used to predict the security situation of the industrial control system, and the final predicted security situation value is output. Experimental results show that the proposed method has faster convergence speed and higher accuracy than existing network security situation prediction methods.

Keywords: Gated Recurrent Unit; Industrial Control System; Neural Network; Security Situation Prediction; Self-Attention Mechanism

1 Introduction

Industrial control system refers to the induction, analysis and arrangement of all kinds of information collected by sensors and LAN through industrial control computer, to realize the integration of information management and automatic control, and can ensure the security of information through authority authentication [4, 6]. It usually includes supervisory control and data acquisition system, distributed control system, process control system and programmable logic controller [14].

In recent years, there have been a lot of organized, hidden, long-lasting and large-scale cyber attacks against Industrial control system, many of which are mixed with national background, such as "Stuxnet" virus, Ukraine power grid virus incident, etc [15].Traditional security measures, such as access control and firewall, only analyze the system from their own point of view, and do not consider the correlation between information, lack of a certain systematic, and the traditional defense measures are also relatively passive, difficult to deal with Industrial control system attacks [7]. It is different from traditional security measures, industrial control system security situational awareness can get information combined with other factors in the system, and the safety of the system state is given, and the macro security situation, and values according to the current safety situation of system to predict the security situation in the future, so as to realize the early warning, make the process more active and defense has targeted [13].

Security situation awareness is usually divided into three stages: security situation recognition, security situation understanding and security situation prediction.At present, there are many methods for security situation awareness. Shi [11] proposed a security situation assessment method of industrial control system based on improved probabilistic neural network, which used the improved Drosophila algorithm to optimize the parameters of the probabilistic neural network, and used the optimized probabilistic neural network for training, and achieved good evaluation results. Zhu [12]combined the information entropy with the improved LSTM, which effectively removed the noise redundancy, reduced the computation amount and improved the prediction accuracy. Yang [17] applied the self-correcting coefficient smoothing method to network security situation prediction. The initial prediction value was obtained by static smoothing adaptive solution, and the time-varying weighted Markov chain was used to modify the initial prediction result of network security situation, which has high adaptability and prediction accuracy.

Through the above analysis, this paper studies and analyzes the industrial Internet security situation predic-

tion method, and combines the security situation characteristics of industrial control system and the attack threat probability distribution in the industrial control vulnerability sub-library of the china national vulnerablility database to obtain the security situation assessment values at different times. Then, a security situation prediction method combining self-attention mechanism and Gated Recurrent Unit (GRU) is proposed, which improves the prediction accuracy and provides support for the research work of industrial control system security situation prediction.

2 Security Situation Prediction Model Method Based on Self-Attention-GRU

2.1 Self-Attention Mechanism

By referring to the attention of human brain, the self-attention mechanism enhances the attention degree of key information to improve the contribution of key information to the results [5]. The neural network with self-attention mechanism can help the model to better learn the relationship between multiple content modalities, avoiding the neural network to treat different predictors equally and ignore the special attention to important information [3].The structure of the self-attention mechanism is shown in Figure 1.

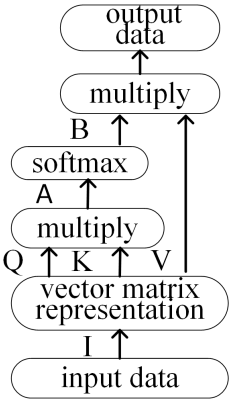


Figure 1: Structure diagram of self-attention mechanism

Self-attention models often adopt the query-key-value (QKV) model, where Q, K, and V in the self-attention mechanism come from the same input [10]. Given the input matrix I, matrices Q, K and V are obtained after different matrix transformations.The matrices Q, K and V are obtained after different matrix transformations. Firstly, the similarity of the query matrix Q and the key matrix K is calculated by transpose multiplication, and the correlation matrix A is obtained.Secondly, the correlation matrix A is normalized to obtain B by softmax operation. Finally, the normalized correlation matrix B is multiplied by the value matrix V to obtain the result of the self-attention layer. The calculation of self-attention mechanism is shown in Equation (1):

$$f_{Self-Att(Q,K,V)} = softmax(\frac{QK^T}{\sqrt{d_K}})V. \quad (1)$$

In the above equation, $\boldsymbol{Q} \in \boldsymbol{R}^{n\times d_k}$, $\boldsymbol{K} \in \boldsymbol{R}^{n\times d_k}$,$\boldsymbol{V} \in \boldsymbol{R}^{n\times d_v}$, T is the transpose of the matrix, d_k,d_v are the dimensions of the vectors k, v. In essence, the self-attention mechanism is to re-encode an $n\times d_k$ matrix into an $n\times d_v$ matrix by considering the global information through matrix operation, and assign different weights to different data.

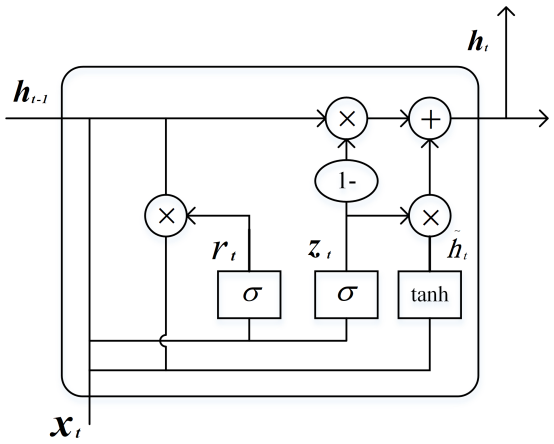


Figure 2: GRU structure diagram

2.2 Gated Recurrent Unit

In order to solve the problem of gradient disappearance and short-term memory in recurrent neural networks, its variants LSTM and GRU introduce the internal mechanism of "gates" to regulate information flow [20]. Compared with LSTM, GRU has simpler structure and calculation process, and fewer parameters, so it converges faster than LSTM [19]. Therefore, the GRU model is selected to extract data features in this paper.

The unit state is removed from the GRU structure, and the hidden state is used to transmit information. Its core structure can be divided into two parts to parse, namely the update gate z_t and the reset gate r_t, whose structure is shown in Figure 2.

The update gate z_t decides to retain information and add information to control the influence of the output hidden state at the previous time on the current time. The reset gate r_t decides to discard information, controlling how much information is ignored in the previous moment. At time t, the update gate and reset gate are firstly calculated.Then the new memory content will then use the reset gate r_t to store past related information. Finally, the network calculates the hidden state h_t, which retains the information of the current unit and passes it to the next

unit. GRU updates parameters through Equations (2) to (5) :

$$\boldsymbol{z}_t = \sigma\left(\boldsymbol{W}_z \cdot \boldsymbol{x}_t + \boldsymbol{U}_z \cdot \boldsymbol{h}_{t-1}\right) \quad (2)$$

$$\boldsymbol{r}_t = \sigma\left(\boldsymbol{W}_r \cdot \boldsymbol{x}_t + \boldsymbol{U}_r \cdot \boldsymbol{h}_{t-1}\right) \quad (3)$$

$$\tilde{\boldsymbol{h}}_t = \tanh\left(\tilde{\boldsymbol{W}} \cdot \boldsymbol{x}_t + \tilde{\boldsymbol{U}} \cdot \left(\boldsymbol{r}_t \circ \boldsymbol{h}_{t-1}\right)\right) \quad (4)$$

$$\boldsymbol{h}_t = \left(1 - \boldsymbol{z}_t\right) \cdot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \cdot \tilde{\boldsymbol{h}}_t \quad (5)$$

In the above equations W_z, W_r, $\tilde{W}$, U_z, U_r, $\tilde{U}$ are the weight matrix of the GRU. x_t is the input vector at time t, h_{t-1} denotes the state information at time t-1, $\tilde{h}_t$ denotes the candidate hidden stateis, h_t denotes the hidden state, $\circ$ stands for the element multiplication operation, σ stands for the sigmoid function.

2.3 Prediction Value Solving Process

Different data in industrial control systems have different importance. To pay more attention to the important information in the industrial control system security data, this paper proposes a security situation prediction method for industrial control system based on self-attention mechanism and GRU neural network. Using the proposed method for security situation prediction can better mine the relationship between the original security data, ensure that important information is not lost, and improve the prediction accuracy [21]. As shown in Figure 3, the network model of this method is mainly composed of the input layer, the self-attention layer, the GRU layer, and the output layer.The model is described as follows.

1) Input layer: the industrial control system security situation data is used as the input of the model. The data sequence can be expressed as $x_1, \cdots, x_n$, and x_i is the security situation value at different time points.The industrial Internet security situation data is converted into a tensor suitable for model processing, and the shape of the input tensor is (number of samples sent, number of loop kernel time expansion steps, number of input features per time step) : In the first dimension, the number of samples is the total amount of data sent to the input layer. In the second dimension, the number of loop kernel time expansion steps is how much historical data is input in a batch at a time. The third dimension is the dimension of the input feature at each time step.

2) Self-attention layer: The self-attention mechanism can assign the information of all input items in the sequence to each item, that is, each input item can use the information of the whole sequence to reason [2]. The self-attention layer improves the important information contribution by iteratively updating the weight of input features. The data sequence S is changed to the data sequence Y by assigning different importance weights to different data through the self-attention layer.

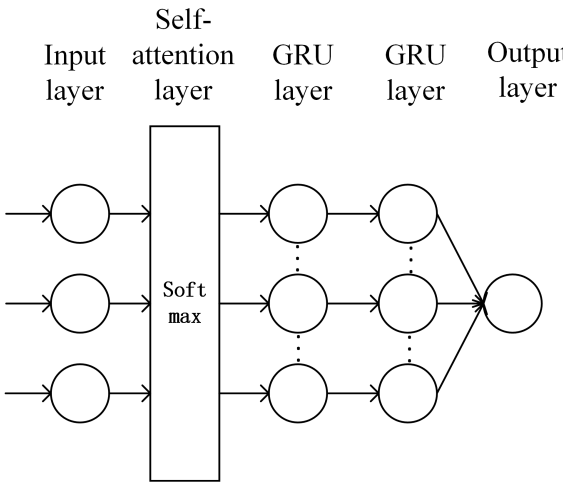


Figure 3: Structure of Self-attention-GRU model

3) GRU layer: A two-layer GRU neural network is used to receive new inputs from the self-attention layer for learning [16]. The first layer of GRU neural network is used to preliminarily predict the security situation data and capture the internal variation law of the sequence. The second GRU makes more accurate predictions based on the processed data of the first layer.Let the output of the GRU layer be h:

$$h = GRU\left(h_{n-1}, y_n\right). \quad (6)$$

4) Output layer: The output layer calculates the output prediction results through the fully connected layers (FCN), and the output is expressed as follows:

$$Y_{predict} = FCN(h). \quad (7)$$

2.4 Loss Function

In the training phase of the network, the mean square error function (MSE) is used as the loss function of the model, which is mathematically defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \quad (8)$$

In the above equation, n is the number of samples; $y_i$is the actual value; $\hat{y}_i$ is the model output value.

Since the whole prediction process is smooth and differentiable, end-to-end learning can be used to jointly optimize the network parameters directly through the forward propagation of the neural network and the back propagation of the gradient.

The Adam optimization algorithm is used to optimize the model parameters [18]. In the optimization process, the weight matrix and bias of neurons are iteratively updated by minimizing the loss function to minimize the output value of the loss function.

3 Experimental Scheme and Analysis of Results

3.1 Experimental Scheme

In order to verify the accuracy of the proposed prediction model, the natural gas dataset publicly available from Mississippi State University is used as the experimental basis. In total, the dataset contains records of 274,627 Modbus network packets, of which 60,048 packets are related to cyber attacks. The natural gas dataset contains normal data and seven different types of attack data, as shown in Table 1.

At present, there is no clear definition of security situation assessment indicators for industrial control systems, and most of them consider three elements: threat, asset and vulnerability [1]. The threat is evaluated according to the probability distribution of each attack threat in the industrial control vulnerability sub-library of the china national vulnerablility database. Since this experimental data set does not provide relevant data of assets and vulnerabilities, the assets and vulnerabilities elements can be simulated by using values in line with the Poisson distribution according to the trend of data change and the characteristics of Poisson distribution [8]. However, considering that the experimental environment is unchanged and the assets and vulnerabilities are fixed values, the threat value when an attack occurs can be used as the security situation value of the industrial control system. Each data is normalized to the industrial system security situation value [9]. The normalization formula is as follows:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{9}$$

In the above equation, x is the data before processing, $min(x)$ and $max(x)$ are the minimum and maximum values in the dataset before processing, x' is the data value after normalization. After experiments, it is found that part of the natural gas data set has a periodicity with a period of 4, so the average value of every 4 normalized values is taken as the security situation value of the industrial control system. The first 1000 industrial control system security situation values are shown in Figure 4. The first 90% of 68656 security situation data is selected as the training set and the last 10% is used as the test set. The training set data is input to the model to obtain the training output, and Equation (8) is used to measure the loss value between the training output and the training label. The model's learnable parameter vector is continuously optimized according to the Adam back propagation algorithm.The trained optimal model is saved and input into the test data set to obtain the corresponding prediction value.

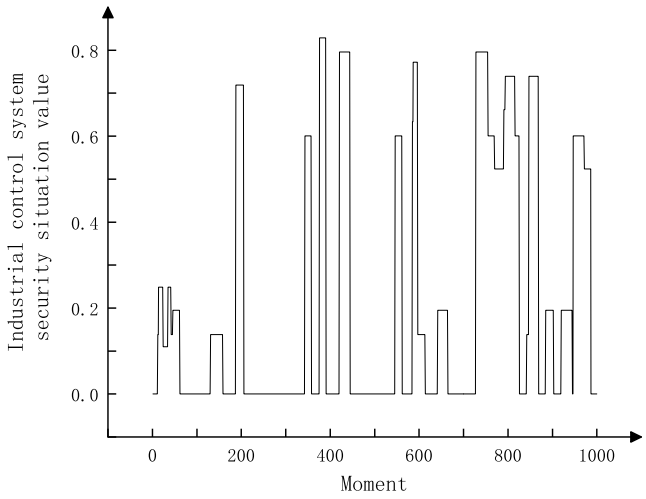


Figure 4: Industrial control system security situation value

3.2 Analysis of Experimental Results

The experimental environment of this paper is: Windows 10 operating system, Python3.6 environment using Keras deep learning framework based on TensorFlow.The hardware configuration is: 64-bit operating system, processor is AMD Ryzen 7 5800H with Radeon Graphics CPU 3.20GHz

3.2.1 Evaluate Metric Selection and Parameter Settings

In this paper, two measurement methods are selected to evaluate the proposed prediction model: mean absolute error (MAE) and root mean square error (RMSE).The specific formula is defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y_i} - y_i| \tag{10}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2} \tag{11}$$

In the above equation, n is the number of samples; y_i is the actual value; $\hat{y_i}$ is the model output value. After data preprocessing in Section 3.1, the optimal parameters of the combined model are obtained through experiments: the number of GRU layer memories is 64 for each layer, the time step is 20, the Batch_size is 256, the parameters of Adma optimizer are 0.001, and epochs is 20.

3.2.2 Prediction Accuracy Analysis

In order to evaluate the performance of the proposed model in security situation prediction, comparative exper-

Table 1: Attack categories and corresponding abbreviations

Category	Description	Label
Normal	Normal data	0
NMRI	Simple malicious response injection attack	1
CMRI	Complex malicious response injection attacks	2
MSCI	Malicious status command injection attack	3
MPCI	Malicious parameter command injection attack	4
MFCI	Malicious function command injection attacks	5
DoS	Denial of service attack	6
Recon	Reconnaissance attacks	7

iments are carried out with other deep learning methods, including GRU, LSTM, RNN, Self-Attention-LSTM,and Self-Attention-RNN. The performance of each model is shown in Table 2.

Table 2: Comparison of model prediction error indicators

Prediction model	RMSE	MAE
Proposed	0.069598	0.013859
Self-Attention-LSTM	0.069715	0.016482
GRU	0.070083	0.024133
LSTM	0.071150	0.033075
RNN	0.069416	0.023168
Self-Attention-RNN	0.069329	0.020466

In this paper, two evaluation indexes, mean absolute error and root mean square error, are selected to measure the accuracy of the prediction results. The two evaluation indexes respectively represent the deviation between the predicted value and the true value and the fitting accuracy, and the smaller the value, the better the prediction effect. According to Table 2, the proposed method, The MAE of Self-attention-LSTM prediction results are 0.013859 and 0.016482. Compared with the MAE of GRU, LSTM and RNN prediction results are 0.024133, 0.033075 and 0.023168 respectively, it can be seen that the single type prediction method cannot effectively predict random fluctuations security situation data. The results show that the self-attention mechanism has a positive effect on the weight redistribution, which indicates that the prediction method of security situation value proposed in this paper is effective. In order to show the change trend of network security situation more clearly, Figure 5 shows the network security situation prediction curves under different methods.

In order to facilitate observation, this paper selects the prediction results of the security situation in the period of 5500-5600 when the security situation value changes frequently to show the effect. As can be seen from Figure 5, we can intuitively see that all prediction models have certain prediction ability,but the predicted value of the proposed method has the highest fitting degree with the real

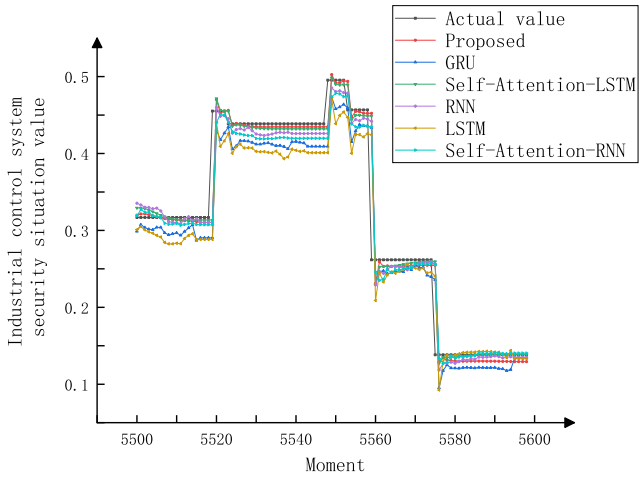


Figure 5: Comparison of security situation values of different prediction models

value. It can be seen from Table 2 that the proposed method is not as effective as RNN in terms of RMSE value, but the MAE value of RNN is far from that of the proposed method. It can be seen intuitively from Figure 5 that RNN is not as effective as the proposed method. The main reason is that although RNN can capture the change law between time series data, it is difficult to capture the long-term dependence between data, resulting in large fluctuations in the predicted data and high MAE values.

3.2.3 Convergence Analysis

Figure 6 shows the variation of the training error of different models with the number of iterations. It can be observed from the figure that all models reach the inflection point at almost the same time during training, but the proposed method has advantages in convergence speed and convergence accuracy, indicating that the model has good learning efficiency.

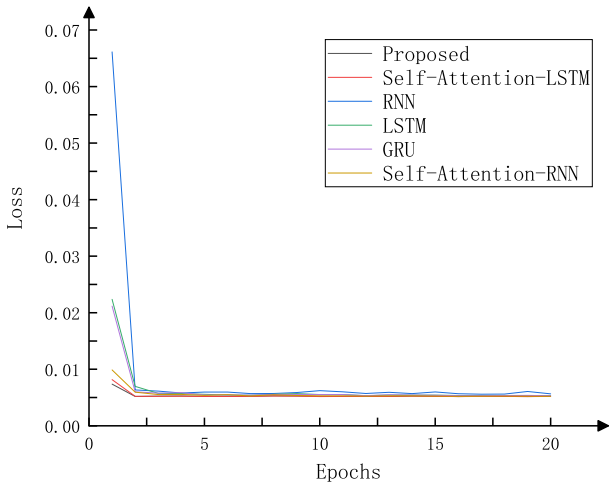


Figure 6: Curve of training error as a function of iteration number

3.2.4 Execution Time Analysis

The execution time referred to in this paper includes the model training time and the test time. The execution time of the proposed method is closely related to the number of iterations of the algorithm, the advantages and disadvantages of the hardware system and other reasons. The execution time of each model is recorded through experiments, as shown in Table 3.

Table 3: Comparison of execution time of different models

Prediction model	Execution time /s
Proposed	136.27159
Self-Attention-LSTM	183.55769
GRU	126.25753
LSTM	143.94127
RNN	52.80577
Self-Attention-RNN	60.30961

Table 3 shows that the execution time of the proposed method is better than that of LSTM and Self-attention-LSTM, which is due to the fact that the GRU unit has one less gate function and the number of parameters is less than that of LSTM. So the training speed of this paper is better than the other two. RNN has the best execution time performance, but RNN is a neuron with a simple activation function inside, and the optimization parameters are few, the prediction error is large, and the network security situation value cannot be accurately predicted.

4 Conclusions

This paper proposes a security situation prediction method based on self-attention mechanism and GRU neural network. This method solves the problem of insufficient accuracy of the existing industrial control system security situation prediction methods. The contribution of different data is extracted by the self-attention mechanism, and then the features between time series data are mined by GRU to predict the security situation. Experimental results show that compared with the existing methods, the proposed method has higher prediction accuracy. In the field of industrial control network security, this method can help network administrators have more time and preparation to deal with the possible coming threats, and reasonably allocate network resources to take correct and effective defense measures for the network.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grants No.61862040 and No.61762059. The authors gratefully acknowledge the anonymous reviewers for their helpful comments and suggestions.

References

[1] R. H. Dong, C. Shu, and Q. Y. Zhang, "Security situation assessment algorithm for industrial control network nodes based on improved text simhash," *International Journal of Network Security*, vol. 23, no. 6, pp. 973–984, 2021.

[2] R. H. Dong, C. Shu, Q. Y. Zhang, and Y. Y. Mo, "Security situation prediction method for industrial control network based on adaptive grey verhulst model and gru network," *International Journal of Network Security*, vol. 24, no. 1, pp. 49–61, 2022.

[3] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3146–3154, 2019.

[4] M. Gidlund, G. P. Hancke Jr, M. Eldefrawy, and J. Åkerberg, "Guest editorial: Security, privacy, and trust for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 625–628, 2020.

[5] W. Hong and Y. a. H. y. Li, "chaotic time series prediction based on hybrid neural network and attention mechanism," *Acta physica Sinica*, vol. 70, no. 1, pp. 235–243, 2021.

[6] X. Hong, "Network security situation prediction based on grey relational analysis and support vector machine algorithm," *International Journal of Network Security*, vol.22, no.1, pp. 177-182, 2020.

[7] F. Kuang, "Network security model for multi-parallel wireless communication based on bmns," *International Journal of Network Security*, vol. 22, no. 6, pp. 997–1003, 2020.

[8] P. Lin and Y. Chen, "Network security situation assessment based on text simhash in big data environment." *International Journal of Network Security*, vol. 21, no. 4, pp. 699–708, 2019.

[9] C. Ma, H. You, L. Wang, and J. Zhang, "Intelligent cybersecurity situational awareness model based on deep neural network," in *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC'20)*, pp. 76–83, 2020.

[10] A. Shewalkar, "Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, 2019.

[11] L. Shi, X. Xu, L. Weihao, and L. jia, "an improved probabilistic neural network security situation assessment method for industrial control system," *International Journal of Network Security*, vol. 21, no. 3, pp. 15–25, 2021.

[12] L. Shi, H. Zhu, L. Weihao, and L. jia, "Industrial control system intrusion detection based on correlation information entropy and cnn-bilstm," *Journal of Computer Research and Development*, vol. 56, no. 11, pp. 2330–2338, 2019.

[13] W. Wang, Z. Wang, Z. Zhou, H. Deng, W. Zhao, C. Wang, and Y. Guo, "Anomaly detection of industrial control systems based on transfer learning," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 821–832, 2021.

[14] G. Wu and J. Sun, "Optimal switching integrity attacks on sensors in industrial control systems," *Journal of Systems Science and Complexity*, vol. 32, no. 5, pp. 1290–1305, 2019.

[15] Y. Xue, "Research on network security intrusion detection with an extreme learning machine algorithm," *International Journal of Network Security*, vol. 24, no. 1, pp. 29–35, 2022.

[16] H. Yang, L. Cheng, and M. C. Chuah, "Deep-learning-based network intrusion detection for scada systems," in *IEEE Conference on Communications and Network Security (CNS'19)*, pp. 1–7, 2019.

[17] H. Yang and Z. Xugao, "network security situation prediction based on self-correcting coefficient smoothing method," *Journal on Communications*, vol. 41, no. 5, pp. 196–204, 2020.

[18] Z. Yifan, "Application of machine learning in network security situational awareness," in *World Conference on Computing and Communication Technologies (WCCCT'21)*, pp. 39–46, 2021.

[19] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[20] H. Zhao, Y. Chang, and W. Wang, "Research on robustness of deep neural networks based data preprocessing techniques," *International Journal of Network Security*, vol. 24, no. 2, pp. 243–252, 2022.

[21] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, "Resnest: Split-attention networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2736–2746.

Biography

Peng-shou Xie was born in Jan.1972. He is a professor and a supervisor of master student at Lanzhou University of Technology. His major research field is Security on Internet of Things. E-mail: xiepsh lut@163. com

Shuai Wang was born in Aug.1994. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 627858493@qq. com

Ying-wen Zhao was born in Feb.1996. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 1376144882@qq. com

Wan-jun Shao was born in Mar.1998. She is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 2443404684@qq. com

Wei Li was born in Mar.1998. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: 2304118232@qq. com

Tao Feng was born in Dec.1970. He is a professor and a supervisor of Doctoral student at Lanzhou University of Technology. His major research field is modern cryptography theory, network and information security technology. E-mail: fengt@lut. cn

Trust and Risk Based on Access Control Model in Social Internet of Things

Hongbin Zhang[1,2], Jian Liu[1], Dongmei Zhao[2], Bin Liu[3,4], Yanmei Wang[5] and Fan Fan[1]
(Corresponding author: Dongmei Zhao)

School of Information Science and Engineering, Hebei University of Science and Technology[1]
Shijiazhuang 050000, China
Hebei Key Laboratory of Network and Information Security, Hebei Normal University[2]
Shijiazhuang 050024, China
School of Economics and Management, Hebei University of Science and Technology[3]
Research Center of Big Data and Social Computing, Hebei University of Scienc and Technology[4]
Hebei Geological Worker's University[5]
Shijiazhuang 050000, China
Email: zhaodongmei666@126.com
(Received Dec. 27, 2022; Revised and Accepted June 12, 2023; First Online July 1, 2023)

Abstract

Integrating social networks and the Internet of Things has formed an emerging field of Social Internet of Things (IoT). Aiming to address the shortcomings of traditional TBAC (Trust based on Access Control) and ABAC (Attribute based on Access Control) in dynamic authorization and security, this paper proposes a hybrid access control model based on trust and risk for SIoT. The model supports social relationship-based trust assessment and risk prediction between SIoT nodes and integrates trust and risk values into SIoT security values through reasonable weight allocation. Furthermore, by simulating the safety requirements of different scenarios, it was verified that the model was efficient and more secure than the current models.

Keywords: Access Control; Attribute Integration; Social Internet of Things; Trust and Risk; Security

1 Introduction

The rapid development of Internet technology offers tremendous potential for the Internet of Things based on social networks. The Social Internet of Things (SIoT) [1], which combines social networks with the Internet of Things, contains the features of social networks and the Internet of Things [2]. It has been progressively extended to new areas of research, such as Smart Home [3,4], Smart Medical [5] and Smart Transportation [6, 7]. Build a social IoT model and analyze the social network structure based on IoT objects, so as to expand the model of human social network into various fields based on things-things, things-people, people-things, and people-people [8]. The Social Internet of Things defines the social relationships between objects within the Internet of Things by adding social network attributes to the Internet of Things. By integrating the characteristics of social networks and the Internet of Things, SIoT has more attributes [9]. Combining new SIoT attributes with traditional access control models and technologies, the implementation of SIoT-based access control mechanism has become an important research direction. Current research has mostly focused on authentication as well as encryption to enforce security controls on network security mechanisms and access authorisation methods [10]. However, whether focusing on encryption to enable authentication is more applicable than an effective access control model in the social IoT still needs further discussion.

The application of social network technology to the field of the Internet of Things has attracted more and more attention, which makes it a new research idea to establish social relations between devices based on the interaction history. In the traditional Internet of Things-oriented access control studies, most of them choose to build trust or risk models, which only authorize access based on trust or risk, instead of combining the two attributes, and implement access control based on the integrated attributes. The trust attribute of SIoT represents the reliability of the service provided by the smart device, and the smart object can establish relationships with the objects it trusts and provide the required services to those people and devices with which they have established good relationships. At the same time, the access between nodes in the traditional Internet of Things is often accompanied by access risks, such as IP address, access time and access location, etc, which need to be considered when object

system implements authorization.

The traditional trust evaluation methods in the field of Internet of Things mainly include: trust fuzzy evaluation based on user behavior and comprehensive trust evaluation based on trust attributes [11–14]. At present, most of the trust evaluations in SIoT are based on the aggregation of direct trust and indirect trust to calculate the overall trust, or to evaluate and aggregate trust through custom methods or protocols, and use the research results for service recommendation or access control [15, 16].

This paper conducts trust and risk assessment based on SIoT social relations and risk factors, which combines the attribute-based access control framework structure of traditional Internet of Things to aggregate trust and risk into SIoT security attributes. As is known to us, the security status of a node plays a significant impact on authorization decisions. After implementing an effective analysis of the security state of the SIoT, we finally implemented an access control mechanism based on SIoT attributes to enhance the security of the SIoT system.

The rest of this paper includes: Section 2 describes the relationship work; Section 3 mainly introduces the main modules of the access control model based on set attributes, including the calculation and quantification process of each attribute value. In Section 4, we verify the model's validity by establishing a reasonable assessment function. The feasibility of the model is illustrated experimentally from the level of interactive success. Finally, We conclude the whole paper in Section 5.

2 Related Work

At present, researches have reviewed the security issues of SIoT, while there is still little specific studies focusing on access control. This paper has extensively summarized and combed the access control research of SIoT in recent years. In current studies, literature [17] adopted Bayesian decision theory and proposes a Bayesian decision-based authorization method for the security problem of authorization in access control. However, without sufficiently considering dynamic and after authorization, authorization security is slightly insufficient. Among other studies oriented towards access decision methods, literature [18] proposed a multi-attribute decision approach that improves the adaptability of the model and the security of authorization through dynamic evaluation of trust, but the model is not adaptable in complex and lightweight IoT environments. In addition, for the identification of malicious nodes and malicious behaviour. Literature [19] used attribute encryption and reliable attribute authorities to improve the effectiveness of the model. In the study of access control based on risk and security attributes, literature [20] combined the attribute-based XACML framework to implement effective risk prediction and security assessment focusing on access interactions of data resources. However, it lacked the combination of risk and trust attributes to achieve more effective system or data resource security management. In the current research for access control in SIoT, Literature [21] focused on fine-grained cross-domain access control mechanism in SIoT, including the trust model of certificate authorization, user login protocol, cross-domain security authentication protocol and cross-domain fine-grained access control, which provides a reference for the security integration of the Internet of Things and social networks. Literature [22] conducted preliminary authorization based on the labeling similarity of SIoT account, uses game theory technology to realize adaptive access control, and realizes fine-grained authorization method based on labeling technology, but the security of the model still needs to be improved. Literature [23] proposed a fine-grained access control scheme based on attributes in Internet of Vehicles, by evaluating multiple trust attributes, comprehensively calculating trust value, which determined whether to authorize according to the threshold, implements fine-granularity authorization based on SIoT trust, and improves system security through trust computing.

In order to improve the security of legal node interaction in the SIoT environment and protect resources or data from theft and destruction by malicious nodes, previous studies have carried out research on access control. In the traditional access control model, RBAC [24] granted permissions to users who have legally obtained roles based on the role-permission mapping relationships. But the model lacked dynamic adjustment of permission granting; TRBAC [25] was an extension of the RBAC model, which proposed a method of trust attribute. The security of the model was been improved and inherited the advantages of the fine-grained authorization of the RBAC [24], however after neglecting to gain high levels of trust through illegal means, permissions are still available, there are still defects in safety; ABAC [26] was based on whether the attribute set of visitors complies with the current access authorization policy. Nervertheless, decide whether to allow or refuse access depending on the main empowerment factors, which is not applicable in the Social Internet of Things. In addition, they met the needs of reasonable authorization and fine-grained degree, while ignored the adjustment of trust and the assessment of risk factors. At present, how to carry out effective trust assessment in the social Internet of Things to prevent malicious nodes from obtaining high trust has become the focus of current research.

The above studies have designed and implemented access control schemes in SIoT or IoT, but lack of analysis and quantification of SIoT attributes between smart objects in SIoT, mostly based on the formulation of SIoT models and access control protocols, or implemented an access control mechanism considering a single attribute. Based on the traditional attribute-based access control framework, we quantifies the multi-social relationships between nodes into two attributes of trust and risk. According to the characteristics of the two attributes, the real-time security value between the subject and the object is calculated. And dynamicly setting a reasonably security

threshold realize access control in different SIoT scenarios. The security attribute, which combines trust and risk factors, represents the security value of the current access environment. In a high trust and low risk environment, the access control mechanism makes authorization decisions based on the security value. It is an extension of the trust based access control and risk-based access control models, and has the advantages of the two models in attribute security.

3 SIoT Trust and Risk Based on Access Control Model

3.1 Access Control Model Design

The access control model based on SIoT attributes in this paper is shown in Figure 1, which consists of three main modules: SIoT trust assessment module, SIoT risk prediction module and security integration module. The trust assessment module mainly aims at the real-time trust of the access initiated by the SIoT node, and obtains a reliable trust value after calculation. Similarity, the risk assessment module coordinates and standardizes the risk factors between SIoT nodes, and obtains the SIoT risk value by calculating the Euclidean distance in the multi-dimensional space. The above modules are all part of the preparation for authorisation decisions. After quantifying the trust and risk values, we integrate trust attributes and risk attributes into security attributes, and implement access authorization in the security integration module. In addition, other modules of the model include audit, security policy, and authorization. Among them, the audit module is used to record and track access operations, the authorisation section includes security classification and security policies, and the authorisation module is implemented in accordance with predefined security access policies. Security attribute integration module is mainly implemented for the two attributes of trust and risk, high trust and low risk will improve the security of the system, otherwise it will reduce the security value between subjects and objects.

We select SIoT social relationship as the evaluation factor of SIoT trust and risk. According to the definition of social relationship in literature [3] and improvement, this paper obtains several basic social relationships as shown in Table 1. In addition, mainly implements SIoT-based system trust assessment for IoT devices, and quantifies the social intimacy and interaction intimacy of devices. Trust assessments are based on two main dimensions: intimacy and social similarity [13, 14].

Definition 1. *Transaction Intimacy(TI).The intimacy-based mutual trust of node i is oriented to node j, and the mutual trust value is obtained by calculating the ratio of the number of successful interactions of nodes i and j to the total number of interactions of nodes. In the SIoT trust evaluation based on social relationship, the social relationship of nodes mainly has two aspects: friendship similarity and group similarity.*

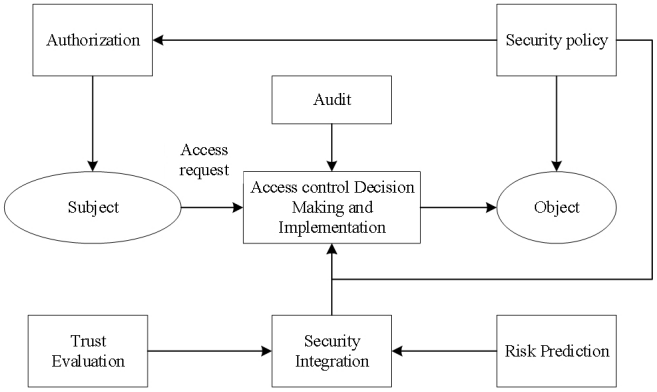


Figure 1: Trust and Risk Based on Access Control Model

Definition 2. *(Friendship Similarity: FS). It refers to the ratio of the common friends between nodes to the number of all friends of the node. The higher the similarity value between nodes, the higher the FS value will be, which also means that their friendship level is higher.*

Definition 3. *(Interest Similarity: IS). It refers to the ratio of the common interests between nodes in SIoT to the number of all interests of the nodes. Similar to FS, the higher the degree of interest similarity, the higher the IS value and the higher the likelihood of providing interest recommendations.*

This paper implements trust assessment based on FS and IS. In the FS calculation, the presence of all basic social relationships has an impact on FS, while the relationship between POR and CLOR is mainly obtained by superposition in the IS calculation. In the rest of this section, the three main modules of the model are explained and illustrated.

3.2 SIoT Trust Evaluation Module

The following is an explanation of the social relationships in Table 1, we describe SIoT trust attributes by fine-grained social relationship types, including POR, OOR, CLOR and CTOR [9, 13].The model evaluates the intimacy trust of nodes to nodes through direct observation. The calculation formula is as follows shown in Equation (1):

$$STrust_{ij}^{FS} = \frac{|\, OR_i \cap OR_j \,|}{|\, OR_i \cup OR_j \,|}. \quad (1)$$

OR represents the social relationship set of node, which contains the following four basic sets of social relations:

$$OR_i = \{POR_i, OOR_i, CLOR_i, CTOR_i, SOR_i\}. \quad (2)$$

The similarity of social interests between node i and node j forms social interest trust.And calculation formula of

Table 1: The social relationship of SIoT

Social relationship type	Descriptions
Parent Object Relationship (POR)	Create objects by the same manufacturer during the same period
Co-location Object Relation (CLOR)	Create between objects (homogeneous or heterogeneous) used in the same location
Ownership Object Relation (OOR)	Create between objects with the same owner
Co-temporal Object Relation (CTOR)	Create between objects at the same period

social interest trust is as follows in Equation (3):

$$STrust_{ij}^{IS} = \frac{\mid IS_i \cap IS_j \mid}{\mid IS_i \cup IS_j \mid}. \tag{3}$$

IS represents the social interest set of node.The calculation formula of IS is as follows in Equation (4):

$$IS_i = POR_i \cdot *CLOR_i. \tag{4}$$

The FS Trust and IS Trust is linearly combined to form SIoT Trust, which is calculated as follows in Equation (5):

$$STrust_{ij}^{Total} = \alpha \times STrust_{ij}^{FS} + (1-\alpha) \times STrust_{ij}^{IS}. \tag{5}$$

where the parameter is an adjustment factor of the total trust of the node.

In addition, the interaction success trust is generated during the historical access interaction of nodes. We believe that interaction trust needs to be initialized before the number of interactions between nodes reaches the threshold, so as to avoid the small number of node interactions and the impact of the interaction trust evaluation mechanism.Considering the instability of interaction trust at the early stage of interaction, the historical interaction trust between nodes is calculated as follows in Equation (6):

$$STrust_{ij}^{HA} = \begin{cases} \beta, N < \pi \\ \frac{A}{A+B}, N \geq \pi \end{cases} \tag{6}$$

A represents the number of successful interactions between nodes, B represents the number of successful interactions between nodes. N is the number of access between nodes.

The trust formed by the historical records between nodes is regarded as the historical efficiency of cooperation between nodes. When the number of interactions is below a certain threshold, no cooperation condition exists; while the trust between nodes is determined by the success rate of inter-node interactions when the number of interactions between nodes reaches a certain level.

In this paper, the evaluation of SIoT comprehensive trust needs to be adjusted according to the mutual trust between nodes. Considering the contribution of interaction trust to total trust throughout the process, high interaction success rates have a positive effect on overall trust, while low historical success rates have a negative effect on trust. Using interaction trust as a moderator, a dynamic moderation of overall trust is implemented. The final comprehensive trust calculation formula is as follows in Equation (7):

$$STrust_{ij}^{Total} = \gamma * STrust_{ij}^{Total} * STrust_{ij}^{HA}. \tag{7}$$

Different from traditional attribute-based trust assessment mechanisms, the initial SIoT trust is built based on social attributes and trust is dynamically adjusted through historical interaction implementation. The resulting composite trust value is more reasonable in the SIoT scenario.

3.3 SIoT Risk Prediction Module

In the SIoT system, the access control entity not only includes smart device objects, but user [1]. Therefore the spatiotemporal attributes of device users are incorporated into the access authorization decision-making factors. This module describes SIoT risk attributes in terms of social attributes and other attributes. In the SIoT environment, moving nodes are spatially variable, and the access of SIoT nodes is accompanied by location and temporality.

The Euclidean distance between the location and temporality of SIoT nodes and the average center point of this attribute in the authorization policy can be taken as the SIoT risk value. The risk value can be specifically interpreted as the degree of deviation between the node's current risk factor and the historical average risk factor. The selection of SIoT risk factors is diverse according to different SIoT scenarios. Since the selection of risk factors is not the focus of this article, this article will not repeat the selection of risk factors.

The number of evaluation factors for SIoT risk is set as n, k is the number of risk predictors and the calculation formula is as follows in Equation (8):

$$SRisk_{ij}^{n} = \sqrt{\sum_{k-1}^{n} (x_{ik} - x_{jk})^2}. \tag{8}$$

The selection of risk factors needs to combine different SIoT scenarios. The space-time factor is one of the most common factors. In other scenarios, equipment power consumption and network conditions are also important risk factors. The evaluation method of these risk factors

needs to be adjusted according to the type of factors. We do not focus on these aspects in detail.

3.4 Security Integration Module

In this module, the SIoT security attribute values between nodes are calculated based on the SIoT trust value and the SIoT risk value. We set an appropriate authorisation threshold based on the security requirements between devices in a given scenario. When the SIoT security attribute value is greater than or equal to when equal to the authorisation threshold, access is granted to the node.

In a trust and risk-based SIoT environment, trust serves as the primary factor in assessing the current security state of a system, while risk serves as a valid attribute for assessing the privacy and security of a user or device. Therefore, we will obtain the security attributes of the current SIoT system based on two main attributes, trust and risk, using an attribute integration approach. The formula for calculating the SIoT security attribute value is as follows in Equation (9):

$$SRT_{ij} = \mu STrust_{ij}^{Total} + (1-\mu) SRisk_{ij}^{n}. \tag{9}$$

There are many attributes in SIoT, among which trust and risk are particularly important. Changes in their attribute values will greatly affect the security status of the current SIoT system. As shown in Figure 2, the two main sources of the SIoT node security attribute quantification model are SIoT trust and SIoT risk.

Among them, the trust attribute is mainly based on the social relationship between the current nodes, and the risk attribute needs to be obtained by aggregating multiple influencing factors in specific scenarios. The trust attribute and risk attribute are used as the evaluation factors of SIoT security attribute, and finally the SIoT security attribute value is obtained.

3.5 Model Implementation

The smart home environment is used as the application scenario to illustrate the implementation of the model in SIoT. In the trust evaluation module, the access is mainly based on the interaction between devices, and the OR is spontaneously formed between devices, and used to guide the object's judgment of the trust degree of the subject node, mainly based on the current OR matrix between them to calculate the initial trust of the device. In addition, historical interactions are a major factor in trust. In addition, the risk prediction between devices is calculated by evaluating risk factors such as geographic location, working hours, and IP addresses between devices. And the authorization of the target node to the subject node is determined by the current security value of the subject node. If some node wants to access objects or resources with high security requirements, it must achieve the corresponding access control security level, and to achieve a certain security level, it needs to work hard from two aspects: high trust and low risk.

4 Experiment and Analysis

The dataset [27] is based on real IoT objects provided by the city of Santander, which classified according to the object types.And data models introduced in the FIWARE data model to construct a realistic small-world based model of a smart city in social IoT movement. The simulation environment is implemented in matlab 2016a.

To observe the relationship between the SIoT security thresholds set by visitors in the SIoT attribute-based access control model and the node access success rate. We selected 329 devices as access initiators, which have their own device ID, device model, device owner, device brand and device type. At the same time, devices can be connected at all times and in all locations. In order to balance the differences of individual devices, 100 devices were randomly selected as subjects and objects among 329 devices, including 99 visitors and one objects. 99 devices respectively initiated 99 requests, and 10 of the 99 devices are randomly selected to observe the final results.

4.1 Validation and Dynamic Adaptability

To simplify the experimental process, the experiment evaluates the initial trust of SIoT nodes based on the three social relationships of OOR, POR and CLOR. In SIoT trust calculation, the intimacy trust and social interest trust values are obtained mainly according to Equation (1)(3)(4),and the historical interaction record between nodes is obtained through Equation (6). The interactive trust value is obtained based on the interaction success rate, which is used as an adjustment factor to dynamically adjust the combined trust value between nodes. We set the initial trust values for OOR and POR to 0.7 and 0.8 respectively.

Also, there are only two risk assessment factors in the two-bit risk space, determine their coordinates in the spatial coordinate system and calculate the multidimensional Euclidean distance between the nodes. In this paper, the risk factors selected are time and position in the 2D scenario as a risk assessment. Euclidian temporal and localization coordinates were obtained from CTOR and CLOR.

In the authorization stage, authority to grant access control is determined by the relationship between the security threshold value set by the system and the current security value of the host and guest nodes. When the security threshold value is greater than the real-time security value, the system denies the node's current access request; otherwise, the node's access request is granted.

The difference between the average security value of multiple accesses to the device, which is calculated as follows in Equation (10). H is the default security threshold and the system threshold is defined as the p.

$$p = \frac{1}{L}\sum_{1}^{L}(SRT_{ij} - H). \tag{10}$$

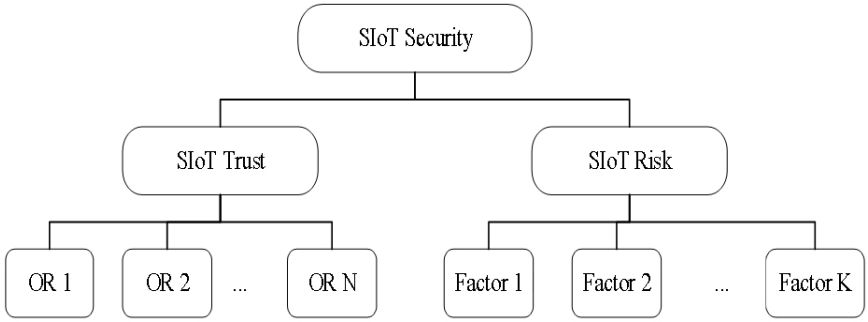


Figure 2: SIoT security attribute quantification model

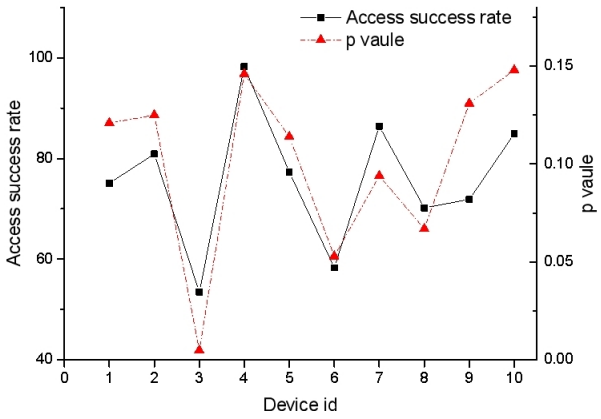


Figure 3: p value and access success rate of the model in the scenario where H=0.2

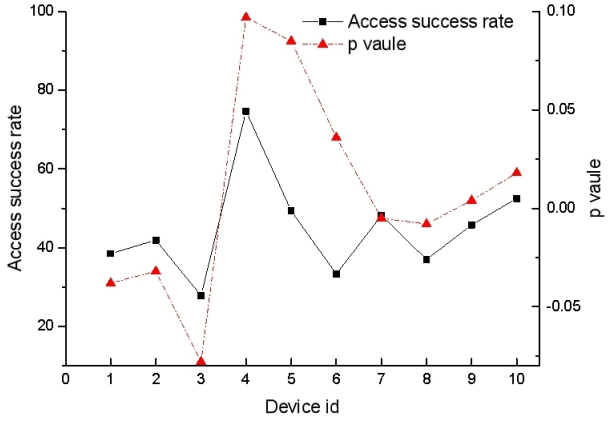


Figure 4: p value and access success rate of the model in the scenario where H=0.3

As shown in Table 2, this paper divides the security status of the accessed system into four levels: low, middle, high and very high. As is generally believed that it is difficult to achieve a very high security state between nodes. The security status among most nodes is mostly low, middle or high. By implementing trust assessment and risk prediction, trust and risk values are obtained and used as a basis to quantify the value of security attributes between nodes as the current node security state value.

When the security threshold set by the system defaults to 0.2, the system security level is required to be lower. At this time, as the device node initiates an access request, the simulation experiment can obtain the access success rate of the device, the average security value of the node, the simulation diagram of the p and and the relationship between them as shown in Figure 3.

It can be seen from the Figure 3 that the average security value of 10 randomly selected access devices is higher than 0.2, p is always positive, and the access success rate of the system is close to 100. When p is larger, the access success rate of the node is closer to 100.

The default security threshold set by the system is 0.3, and the security requirements of the system are medium. When the device node initiates an access request, the access success rate of the device, the average security value of the node and the p value are shown in Figure 4.

As can be seen from Figure 5 and Figure 6, there are two situations for p: when it is positive, the device access success rate is greater than 50. At this time, the larger p is, the access success rate is slowly rising on the basis of 50; when it is negative, the device access success rate is lower than 50. At this time, the larger p is, the further the access success rate will decrease.

The above experimental results can prove that the model is reliable to predict the success rate of access through the p value. According to the principle that the access control interception rate corresponds to the system requirements in different scenarios, the validity of the system model will be verified below.

To verify the dynamic flexibility of the access control model under different scenarios, the system security thresholds of 0.1, 0.2, 0.3, 0.4 and 0.5 were selected for experimental simulation due to the different requirements for the system security state under different scenarios.

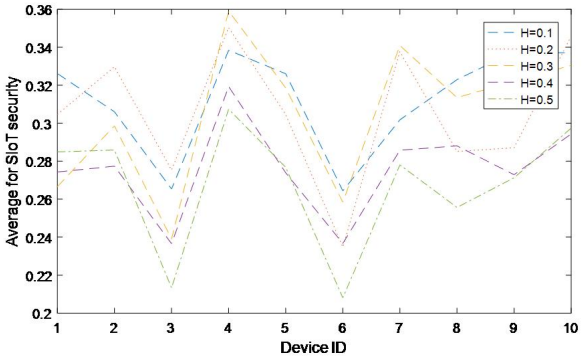


Figure 5: Average value of security attributes of SIoT nodes

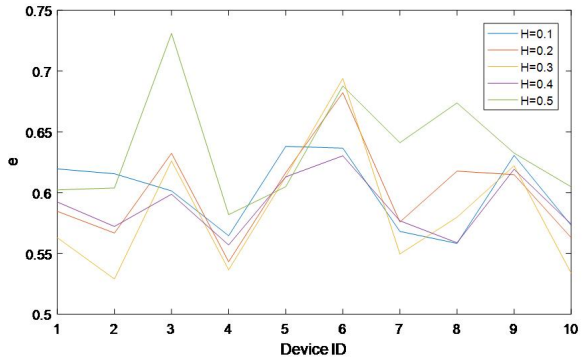


Figure 6: Effectiveness of access control mechanisms

And the effectiveness of the access control of the model is analysed in five scenarios.

This paper calculates the effectiveness of the model mechanism in access control, where p is the average interaction success rate of the nodes and e denotes the effectiveness of access control.The calculation method is as follows in Equation (11):

$$e = \begin{cases} 0.5^{(p-1)} - 1, p \geq 0 \\ malicious, p < 0 \end{cases} \quad (11)$$

It can be seen from Figure 5 and Figure 6 that the random fluctuation and situational distribution of the security value among the nodes of the access control of the system in different scenarios. As the system security level requirement increases, the effective value e of the access control mechanism increases, in line with the principle of high security requirement and high effectiveness. The higher the value of p, the lower the value of e, which can be interpreted as the higher the security between the nodes and the easier the access control mechanism. In addition, there are a large number of malicious nodes with low trust values and high risk values in the IoT environment. Given that malicious nodes tend to be insecure, the model's identification of malicious nodes is mainly based on positive and negative p-values. When the p value of the node is negative, it is determined as a malicious node, and recorded in the decision database.

Table 2: Relationship between safety value and system safety level

SRT	Security status level
[0, 0.2)	Low
[0.2,0.3)	Middle
[0.3,0.5)	High
[0.5,1.0]	Very High

4.2 Comparing with Other Literatures

Our model is compared to other models, as can be seen from Table 3. Considering the two security attributes - trust and risk, the initial trust of the node is evaluated for the social relationship between the nodes in SIoT. Current trust value is adjusted based on the access history interaction results. At the same time, the access control can be dynamically changed by adjusting the authorization threshold in different SIoT scenarios. Compared with other literatures, the security of this model is greatly improved. In view of scenario changes and security requirements, this paper can adjust the authorization threshold to achieve dynamic adaptivity of access control. Compared with the complex algorithms of other models, the method in this paper is more suitable for light weight IoT scenarios.

In this paper, trust and risk are used as the main decision attributes for the authorization of the access control mechanism based on the multiple social relationships of SIoT nodes, which are synthesized by the method of weight analysis to obtain the values of security attributes among nodes. According to the security requirements in the scenario, the authorization threshold is reasonably adjusted to achieve a dynamic, flexible and efficient SIoT attribute-based access control model.

5 Conclusions and Future Work

In SIoT environment, we propose an access control model based on trust and risk. First, we have integrated social relationships and spatial and temporal characteristics into the confidence and risk attributes of site access. Second, we calculated SIoT trust and risk by assessing the attributes. Then, we obtained the security attribute values between SIoT access nodes. Ultimately, security values are treated hierarchically with fuzzy classification and reasonable mapping is determined by the level of safety of the system. Through the experimental form of multiple access nodes accessing the same node, combined with experimental results and practical scenarios, we demonstrate the effectiveness and rationality of the access control model and implement a trust and risk-based access

Table 3: Comparison with other references

Schemes	Trust Property	Risk Property	System Security	Dynamically	Lightweight
Wu et al. [21]	Y	N	N	Y	N
Pan et al. [11]	Y	N	N	N	N
Zheng et al. [28]	N	Y	Y	N	N
Zhang et al. [22]	N	Y	N	Y	N
This paper	Y	Y	Y	Y	Y

control mechanism in SIoT scenarios. In future work, we will explore more SIoT attributes and improve experimental methodologies and scenario simulation more efficiently.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under grant No.61672206 and No.61572170, Central Guide Local Science and Technology Development Fund Project(216Z0701G), Science and Technology Program of Hebei under Grant No.18210109D, No.20310701D, No.20310802D, No.21310101D, National cultural and tourism science and technology innovation project(2020).

References

[1] L. Atzori, A. Iera, and G. Morabito, "Siot: giving a social structure to the internet of things," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1193–1195, 2011.

[2] B. T. Mi, D. Liang, and S. Zhang, "A survey on social internet of things," *Chinese Journal of Computers(In Chinese)*, vol. 41, no. 07, pp. 1448–1475, 2018.

[3] D. O. Kang, J. H. Choi, J. Y. Jung, and et al, "Sdif: social device interaction framework for encounter and play in smart home service," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 01, pp. 85–93, 2016.

[4] D. Diazsanchez, A. Marin, F. Almenarez, and et al, "Social applications in the home network," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 01, pp. 220–225, 2016.

[5] D. A. Gupta, C. Dwith, and B. Ramakanth. "Wireless home automation using social networking websites,", 2014.

[6] S. Shaji, M. V. Ramesh, and V. N. Menon, "Real-time processing and analysis for activity classification to enhance wearable wireless ecg," *Springer India*, 2016.

[7] Q. Yang and H. G. Wang, "Toward trustworthy vehicular social networks," *IEEE Communications Magazine*, vol. 53, no. 08, pp. 42–47, 2015.

[8] K. M. Alam, M. Saini, and A. EI. Saddik, "Toward social internet of vehicles: concept, architecture, and applications," *IEEE Access*, vol. 03, pp. 343–357, 2015.

[9] L. Atzori, A. Iera, G. Morabito, and et al, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[10] M. M. Nabi and F. Nabi, "Cybersecurity mechanism and user authentication security methods," *I.J. of Electronics and Information Engineerin*, vol. 14, no. 1, pp. 1–9, 2022.

[11] R. J. Pan, G. C. Wang, and H. Y. Huang, "Attribute access control based on dynamic user trust in cloud computing," *Computer Science*, vol. 48, no. 05, pp. 313–319, 2021.

[12] A. Long and H. Liu, "Access control model of p2p based on trust fuzzy evaluation," *Computer Engineering(In Chinese)*, vol. 41, no. 03, pp. 125–129, 2015.

[13] S. Anbazhagan and K. Subramanian, "Ctrac: a combined trust and recommendation based access control approach for cloud computing," *Asian Journal of Research in Social Sciences and Humanities*, vol. 06, no. 06, pp. 1824–1841, 2016.

[14] G. H. de Oliveira, A. de Souza Batista, M. Nogueira, and et al, "An access control for iot based on network community perception and social trust against sybil attacks," *International Journal of Network Management*, vol. 32, 2022.

[15] Y. Zhang, L. Q. Tian, Z. N. Wu, and et al, "Trust oriented dynamic access control game mechanism in cloud services," *Journal of Chinese Computer Systems(In Chinese)*, vol. 42, no. 08, pp. 1774–1779, 2021.

[16] Y. Y. Li, H. R. Guo, W. P. Peng, and et al, "Trust attribute-based access control policies composition," *Application Research of Computers(In Chinese)*, vol. 33, no. 07, pp. 2175–2180, 2016.

[17] B. Zhao and J. He, "Bayes decision theory based risk minimization authorization mapping," *Journal on Communications(In Chinese)*, vol. 36, no. S1, pp. 157–161, 2016.

[18] R. H. Dong, T. T. Xu, and Q. Y. Zhang, "Access control model of industrial control system based on multi-attribute decision making," *International*

Journal of Network Security, vol. 23, no. 6, pp. 1037–1048, 2021.

[19] L. Y. Zhang, C. Song, and Y. Mu, "Secure and accountable data access control against malicious behavior in smart grids," *International Journal of Network Security*, vol. 24, no. 1, pp. 109–122, 2022.

[20] P. S. Xie, H. J. Fan, T. Feng, and et. al, "Adaptive access control model of vehicular network big data based on xacml and security risk," *International Journal of Network Security*, vol. 22, no. 2, pp. 347–357, 2020.

[21] J. Wu, M. Dong, K. Ota, and et al, "A fine-grained cross-domain access control mechanism for social internet of things," in *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, pp. 666–671, 2014.

[22] H. B. Zhang, P. C. Ma, and B. Liu, "Adaptive fine-grained access control method in social internet of things," *International Journal of Network Security*, vol. 23, no. 01, pp. 42–48, 2021.

[23] G. H. Chang, Y. B. Liu, and Q. Ye, "An attribute-based fine-grained access control scheme in vehicular ad-hoc networks," in *2017 International Conference on Wireless Communications, Networking and Applications*, pp. 40–44, 2017.

[24] Y. Qiao, D. Xu, and G. Dai, "A new role-based access control model and it's implement mechanism," *Journal of Computer Research and Development(In Chinese)*, vol. 01, pp. 37–41, 2000.

[25] W. Liu, H. X. Duan, H. Zhang, and et al, "Trbac: trust based access control model," *Journal of Computer Research and Development(In Chinese)*, vol. 48, no. 08, pp. 1414–1420, 2011.

[26] X. F. Li, D. G. Feng, Z., W. Chen, and et al, "Model for attribute based access control," *Journal on Communications(In Chinese)*, vol. 04, pp. 90–98, 2008.

[27] C. Marche, L. Atzori, V. Pilloni, and et al, "How to exploit the social internet of things: query generation model and device profiles' dataset," *Computer Networks*, vol. 174, 2020.

[28] L. J. Zheng, L. Zhang, M. Cui, and et al, "The research of mobile location privacy protection access control method based on game theory," *Wireless Communications and Mobile Computing*, 2018.

Biography

Hongbin Zhang received the Ph.D. degree from the School of Computer Science and Technology, Xidian University, Xi'an, China. His current research interests include network security situation awareness, system internal threat analysis, Social Internet of Things security, etc.

Jian Liu is a postgraduate student of Hebei University of Science and Technology, majoring in Social Internet of Things security and access control during his postgraduate period, and so on.

Dongmei Zhao received the Ph.D. degree from Xidian University, Xi'an, China. Now she is a professor of Hebei Normal University and director of cyberspace security discipline. Her current research interests include networks and information security, artificial neural networks, computer applications, etc.

Bin Liu is a Professor and Master Tutor of Hebei University of Science and Technology, received Ph.D. in Computer Software and Theory, Beijing University of Technology, and post-doctor in Department of Computer Science, Tsinghua University (State Key Laboratory of Intelligent Technology and Systems).

Yanmei Wang received the M.D. degree from School of Finance, Tianjin University of Finance and Economics, Tianjin, China. Her research interests include economic information management, Social Internet of Things security.

Fan Fan is a postgraduate student of Hebei University of Science and Technology, majoring in Social Internet of Things security and trust management during his postgraduate period, and so on.

Analysis Distributed Denial-of-Service Attack Deploy Deep Learning Techniques

Sirajuddin Qureshi, Jingsha He, Saima Tunio, Nafei Zhu, Faheem Ullah, Ahsan Nazir, and Ahsan Wajahat

(Corresponding author: Nafei Zhu)

Faculty of Information Technology, Beijing University of Technology Beijing. 100124, China

Email: znf@bjut.edu.cn

(Received May 12, 2022; Revised and Accepted May 28, 2023; First Online Aug. 28, 2023)

Abstract

Network devices are essential to connect nodes and users on any given network. Network devices perform the additional task of protecting services and users from known and unknown attacks. This feature of network devices to stop or minimize network attacks to secure the nodes and all attached devices needs further research studies and experimentation to confirm their resilience against potential known attacks. Denial-of-Service (DoS) Attack is one of the deadliest attacks that make network services and devices unavailable. One of these attacks, which is growing significantly, is the Distributed Denial of Service (DDoS) attack. DDoS attack has a high impact on crashing the network resources, making the target servers unable to support valid users. The current methods use the standard datasets to deploy the Deep Learning (DL) Model for intrusion detection against DDoS attacks in the network. However, these methods suffer several drawbacks, and the used datasets do not contain the most recent attack patterns – henceforward, lacking in attack variety. In this paper, we proposed an interruption detection model system and, against DDoS attacks is based on DL technique, the combination of the Recurrent Neural Network (RNN) and Deep Neural Network (DNN) algorithms are compared with an autoencoder. We evaluated our DL model system using the newly released dataset CIC_DDoS2020, which contains a comprehensive variety of DDoS attacks and addresses the gaps of the existing current datasets (CIC_DDoS2020). We obtained a significant improvement in attack detection compared to other benchmarking methods. Hence, our model provides excellent confidence in securing these networks.

Keywords: Deep Learning (DL) Technique; Deep Neural Network (DNN Algorithm); Distributed Denial of Service (DDoS Attack); Intrusion Detection Against; Recurrent Neural Network (RNN Algorithm)

1 Introduction

The Distributed Denial of Services attack (DDoS) attack is a popular threat in the services provided online. Such a type of attack is used to target the packets and destroy different network resources such as the bandwidth of the network is compromised, the servers or the equipment can be crashed [20]. One of the useful methods to overcome the DoS attack is packet filtration on the DDoS attack routers that prevent DDoS attacks by identifying as well as blocking the attack before reaching its target. The authors provided another method for the prevention of DDoS attacks. An algorithm based on learning as well as the statistical analysis was proposed for a packet filtration system. [24] researched Distributed Denial of Services attacks (DDoS). The DDoS in the UDP-based network has been abused by troublemakers. Amplification systems can face these vulnerable challenges. The authors divided this problem into four steps. The first step was to monitor and classify the source of amplification that showed the high diversity in OS architecture.

Based on the results, the authors collaborated with the security community in a large-scale campaign for reducing the vulnerable NTP up to 92%. The authors analyzed and found the root cause of amplification attacks these may be the networks that are allowing the spoofing of IP addresses. The authors deployed a method for identifying spoofing-enabled networks. In our recent experiment, no significant differences in tuber yield of water yam strain cv. proposed a defense system for facing the Distributed Denial of Services attacks that is the combination of both software-defined controllers as well as the decision-making system based on fuzzy. The Numerical results in the research of the authors show that their proposed system has a very low computation load as well as the response times are high as much as 38. 04% of the N intake were thought to have been derived from the air and strains of nitrogen-fixing bacteria (NFB) were subsequently isolated from the stem and roots. The publisher regrets to inform the readers that the typesetter misinterpreted the

correction from the author. The text 'In the case of the Andean condor, recent satellite tracking revealed that the home range of immature birds (299, 770 km2) is more than twofold that of adults (> 290, 000 km2) in northern Patagonia an attack prevention system known as Link Scope. According to the authors, it is a novel system that can employ both hope-to-hope as well as end-to-end network measure techniques for capturing the abnormal paths for detecting flooding attacks. The authors tackled several numbers of problems such measurement of long-scale internet paths. The authors deployed link scope in 7174 lines of the Python Programming code and detected with accuracy.

1.1 Distributed Denial of Services Attack (DDoS)

This research Distributed Denial of Services attack (DDoS). The DDoS in the UDP-based network has been abused by troublemakers. Amplification systems can face these vulnerable challenges. The authors divided this problem into four steps. The first step was to monitor and classify the source of amplification that showed the high diversity in OS architecture. Based on the results, the authors collaborated with the security community in a large-scale campaign for reducing the vulnerable NTP up to 92%. The authors analyzed and found the root cause of amplification attacks these may be the networks that are allowing the spoofing of IP addresses. The authors deployed a method for identifying spoofing-enabled networks.

In normal network circumstances, a user and server perform a three-way handshake. However, DDoS attack, a user sends a multiple handshake request by sending packets to the server but does not reply to the server by sending an acknowledgment. In this process, the connection between the user and server remains half-opened for a certain period of time. These half-open connections formulate and resource constraint on the server and the server remains busy for that whole period of time. In the meanwhile, if a legitimate user tries to develop a connection with the server, then the server declines to open any of the connections and hence this scenario can perfectly be called a denial-of-service attack. This TCP and IP connection is the backbone of the DDoS packets which are exchanged between a user and a server. This is also known as TCP/IP three-way handshake where these three steps are performed.

1) A client request to the server for a connection through DDoS: (Synchronizepackets);
2) The server replies with Synchronization-Acknowledgement: DDoS-ACK;
3) A client replies to the server with DDoS-ACK to develop and connection.

Different scientists have proposed different approaches to handle DDoS attacks at various network levels.

1.2 Router Based DDoS Attack

This research work encompasses and router-based DDoS attack. They further evaluated the router-based DDoS attack. The authors proposed a novel approach for preventing the DoS attack in-network, and an approach of DPF was introduced [23]. The authors have shown that the DPF is a packet filtration system that is able to achieve scalability and proactiveness as well. they have shown that there is an intimate relationship between the effectiveness of DPF in mitigating the DoS attack [22]. The silent feature of the authors' research was to filter the spoofed packet and prevent attacks.

To achieve this research study, it is very essential to analyze as well as to compare the quality of routing devices that are used to manage the network. More specifically, the focus may be kept on comparing the performance of Ubuntu router pc router DDoS attacks from both ends (Intranet and Internet). The Quality of Service (QoS) may also need to be analyzed by using a static routing method in accordance with parameter delay, throughputs, and loss of packets. According to collection comparison results, the DDoS attacks router is most stable in moderate conditions with 47 ms delay parameter 54 KBit/s throughput and time of crowded condition delay parameter 51 ms, packet loss 7% while router PC path is only stable at the low time seen from 45 ms delay parameter, throughput 54 KBit/s. In terms of comparison of hardware interface and software interface, router DDoS is better in the software interface, and PC router is better in hardware interfaces.

2 Related Works

[4] argues that most notions of flatness are problematic for deep models and cannot be directly applied to explain generalization. Specifically, when focusing on deep networks with rectifier units, we can exploit the geometry of parameter space induced by the inherent symmetries that these architectures exhibit to build equivalent models corresponding to arbitrarily sharper minima. Furthermore, if we allow to reparametrize of a function, the geometry of its parameters can change drastically without affecting its generalization properties. The key objective of a Distributed Denial of Service (DDoS) attack is to compile multiple systems across the Internet with infected zombies/agents and form botnets of networks [16]. The purpose of this paper is to detect and mitigate known and unknown DDoS attacks in real-time environments. We have chosen an Artificial Neural Network (ANN) algorithm to detect DDoS attacks based on specific characteristic features (patterns) that separate DDoS attack traffic from genuine traffic. Detection of DDoS attacks in the wake of flash crowds is a challenging problem to be addressed [6]. The existing solutions are generally meant for either flash crowds or DDoS attacks and more research is needed to have a comprehensive approach for catering to the needs of detection of spoofed and non-spoofed variants of DDoS

attacks. This paper proposes a methodology that can detect DDoS attacks and differentiate them from flash crowds. NS-2 simulations are carried out on the Ubuntu platform for validating the effectiveness of the proposed methodology. Nowadays, in the field of SDN, various machine learning (ML) techniques are being deployed for detecting malicious traffic. Despite these works, choosing the relevant features and accurate classifiers for attack detection is an open question. For better detection accuracy, in this work, Support Vector Machine (SVM) is assisted by kernel principal component analysis (KPCA) with a genetic algorithm (GA) [15]. In the proposed SVM model, KPCA is used for reducing the dimension of feature vectors, and GA is used for optimizing different SVM parameters. To reduce the noise caused by feature differences, an improved kernel function (N-RBF) is proposed. The experimental results show that compared to single-SVM, the proposed model achieves more accurate classification with better generalization. Moreover, the proposed model can be embedded within the controller to define security rules to prevent possible attacks by attackers. If the attack source is single, then the attack is referred to as denial of service (DoS) and if the attack is sourced from divergent servers, then it is referred to as DDOS. It is imperative from the analysis that there are constraints in the existing models since most of these models are user session-based and/or packet flow patterns [13]. The session-based evolution models are vulnerable to botnets and packet flow pattern-based models are vulnerable if attack sources are equipped with human resources and/or proxy servers. Hence, there is an inherent need for improving the solutions towards addressing the App-DDoS attacks over the system. The crux for such a system is about ensuring fast and early detection with minimal false alarms in streaming network transactions and ensuring that genuine requests are not impacted. To address such a system, the model of Bio-Inspired Anomaly-based App-DDoS detection is aimed, and the proposed model is depicted in detail along with experimental inputs. As for the mitigation approaches, to detect the flooding DDoS attack, the conventional schemes using the bloom filter, machine learning, and pattern analysis have been investigated. However, those schemes are not effective to ensure high accuracy (ACC), a high true positive rate (TPR), and a low false-positive rate (FPR) [3]. In addition, the data size and calculation time are high. Moreover, the performance is not effective from the fluctuant attack packet per second (PPS).

Threats of distributed denial of service (DDoS) attacks have been increasing day by day due to the rapid development of computer networks and associated infrastructure, and millions of software applications, large and small, addressing all varieties of tasks. Botnets pose a major threat to network security as they are widely used for many Internet crimes such as DDoS attacks, identity theft, email spamming, and click fraud [7,19]. Botnet-based DDoS attacks are catastrophic to the victim network as they can exhaust both network bandwidth and resources of the victim machine. An Ad hoc Network is a wireless multi-hop network with various mobile, self-organized, and wireless infrastructure nodes.

The goal of [1, 5] is to implement a simulation model called DDoS Attack Simulation Model (DDoS) in Network Simulator 2(NS-2) and to examine the effect of DDoS attacks on various routing protocol types in MANET namely: Zone Routing Protocol (ZRP), Ad hoc On-Demand Distance Vector (AODV) protocol and Location-Aided Routing (LAR) protocol. The introduced model uses the NS-2 simulator to apply DDoS on the three chosen routing protocols. In terms of throughput and end-to-end latency under the consequences of the attack, the performance of three routing protocols was analyzed. Distributed Denial of Service (DDoS) attack has become one of the most destructive network attacks which can pose a mortal threat to Internet security. Existing detection methods cannot effectively detect early attacks. In this paper, we propose a detection method for DDoS attacks based on generalized multiple kernel learning (GMKL) combined with the constructed parameter R. The superfusion feature value (SFV) and comprehensive degree of feature (CDF) are defined to describe the characteristic of attack flow and normal flow [4].

The network traffic was classified by the detection system in a controlled network environment using different sampling rates. In the experiments, raw network traffic of the CIC_DDoS 2020 [10, 18], datasets and the raw network traffic captured in the customized testbed experiments were employed. DDoS attacks Detection system has reached high accuracy and low false-positive rate. Experiments were conducted using two Virtual network traffic classifieds.

Antidote system [2,19] presents a means of interaction between a vulnerable peripheral service and an indirectly related Autonomous System (AS), which allows the AS to confidently deploy local filtering rules under the control of the remote service.

3 Methodology

DDoS attacks on the Internet in a modern collaborative way. In this approach, the system collects network traffic samples and classifies them. Attack notification messages are shared using a cloud platform for convenient use by traffic control protection systems. Whole process is illustrated in Figure 1. The crucial steps from model build to system operation.

First, normal traffic and DDoS signatures were extracted, labeled, and stored in a database (CIC_DoS2020), which was then created using feature selection techniques. Finally, the most accurate MLA was selected, trained, and loaded into the traffic classification system. (e.g. architecture of the detection system was designed to work with samples of network traffic provided by industrial standard traffic sampling protocols, collected from network devices). The samples are received and grouped in

Table 1: Current related works

References	Dataset	Online	L/H_DoS	DoS_Attacks
[16]	CIC-DoS	True	False	True
[18]	None	False	True	False
[19]	DoS_Customized	True	False	True
[8]	Developed_Authors	False	True	True
[21]	CIC_DoS (2019)	True	False	False
Proposed DL Model	CIC_DoS (2020)	True	True	True

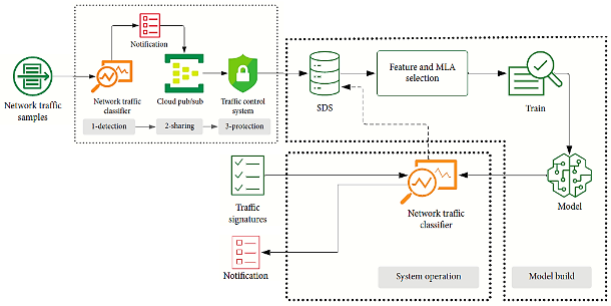


Figure 1: DDoS attacks on the Internet in a modern collaborative

flow tables in the receiver buffer. US, when the table length is greater than or equal to the reference values, they are presented to the classifier responsible for labeling them. If the flow table expires, it may be processed one more time. (e.g. occurrence of small flow tables is higher at lower sampling rates or under some types of DDoS attacks - DDoS flood attacks). Table 2 details the parameters for fine-tuning the system. (e.g. complete algorithm of the detection system is summarized in Figure 1). During each cycle of the detection process, traffic samples are received and stored in a flow for each new flow, and a unique identifier (FlowID) is calculated based on the 5-tuple (src_IP, dst_IP, src_port, dst_port, and transport_protocol) in Steps 1 and 2. If this is a new flow, [13, 14, 19], there is not any other flow table stored with the same FlowID, the flow table is registered in a shared memory buffer. Otherwise, if there is a flow table registered with the same FlowID such as the previously calculated one, the data of the new flow will be merged with the data in the existing flow table in steps 3 and 4. After the merging operation, if the length is greater than or equal to the reference value $Tl \leq Tmax$, the flow table is classified, and if it is found to be an attack, a notification is emitted. Meanwhile, in Step 7, the cleanup task looks for expired flow tables in the shared buffer, i.e., flow tables that exceed the expiration time of the system $E < ET$. For each expired flow table, the system checks the table length. If the flow table length is less than or equal to the minimum reference value $Tl \leq Tmin$, this flow table will be processed by Step 8. A new FlowID is calculated using the 3-tuple (src_IP, dst_IP, and transport_protocol).

3.1 Traffic Sampling

Detection uses a network traffic sampling technique because processing all the packets in the network can be a computationally expensive task, even if only the packet headers are parsed. In many cases, performing a deep inspection and analyzing the data area of the application layer is unfeasible for detection systems. Among the protocols adopted by the industry for sampling network traffic, the sFlow protocol is widely used in current devices, e.g. technique used by sFlow is called N-out-of-N sampling.

In this technique, n samples are selected out of N packets. One way to achieve a simple random sample is to randomly generate n different numbers in the range of 1 to N and then choose all packets with a packet position equal to one of the n values. Besides, the sample size is fixed in this approach [14, 24]. Flow monitoring system consists of an agent (embedded in a switch, a router, or an independent probe) and a collector. Architecture used in the monitoring system is designed to provide continuous network monitoring of high-speed switched and routed devices. Agent uses the sampling technology to capture traffic statistics from the monitored device and forward them to a collector system.

3.2 Feature Extraction

In supervised classification strategies, a set of examples is required for training the classifier model. (Is set is commonly defined as the signature database. Each instance of the database has a set of characteristics or variables associated with a label or a class. In this work, the goal is to identify characteristics in network traffic that can distinguish normal network behavior from DoS attacks. Study is focused on the analysis of the header variables of the network and transport layer packets of the TCP/IP architecture because it allows saving computational resources and simplifies the deployment in the ISP networks [11,12]. To achieve this research study, a proper literature review has been conducted to validate the null hypothesis. Additionally, a real environment, as well as a simulation setting, has been developed for experimentation of this research work. Additionally, PYTHON has been utilized to

develop a proper testing environment by defining a DDoS router and by fine-tuning and optimizing it against DDoS attacks. At the initial stage, the router has been tested against a DDoS attack without fine-tuning the configuration against this kind of attack. Test readings have been taken, and data have been analyzed for comparison at the later stages. Since DDoS attack exploits TCP and IP three-way handshake and lets the half-open connection be used for malicious data transfer or theft of data. A similar attack has been simulated in Deep Learning Techniques. Readings of real vs simulated data have been compared and presented in the following subsequent sections. As the last step, the countermeasures of DDoS attacks have been taken to see if there is any improvement in router performance by comparing the result before and after taking countermeasures. This research study is going to help the research community to effectively take measures against DDoS attacks and make the performance of routers improved in many folds. In the following, a PYTHON simulation environment is presented where a test network is presented.

To effectively detect the flooding DDoS attack, we proposed lightweight detection using a bloom filter against flooding DDoS attacks. To detect the flooding DDoS attack and ensure high accuracy, a high true positive rate, and a low false-positive rate, the dec-all (decrement-all) operation and the checkpoint are flexibly changed from the fluctuant PPS in the bloom filter. Since we only consider the IP address, all kinds of flooding attacks can be detected without the blacklist and whitelist. Moreover, there is no complexity to recognize the attack. By the computer simulation with the datasets, the authors introduce the DDoS attacks and discuss the incapability of network-level detection methods for catching the DDoS attacks. These attacks are growing rapidly, are harder to detect, and cause severe problems in accessing a particular online service (or webserver) as compared to the Net-DDoS attacks. In invulnerability attacks, the attacker browses for unprotected openings in the software implementation and exploits them to bring the system down or to recruit zombies for further attacks. These attacks use the exacted performance of different protocols (such as TCP/IP and HTTP) to ravage the resources of the victim server and prevent it from processing events or requests from authorized users.

PYTHON network simulation environment contains DDoS router software called RouterOSWinBox v5. 20 which is installed on the PC having network connectivity with the network switch. Router_OS can also be installed on Router_BOARD and serve as Router Operating System. DDoSRouterOS relates to a switch. There are three more nodes that relate to the switch which are a test client, a web server, and an attacker machine. These three machines are connected on a separate VLAN of the switch. A test client is connected to VLAN 10, a web server machine is connected to Vlan 20, and an attacker machine is connected to Vlan 30. After achieving these connections, a ping is performed to see whether

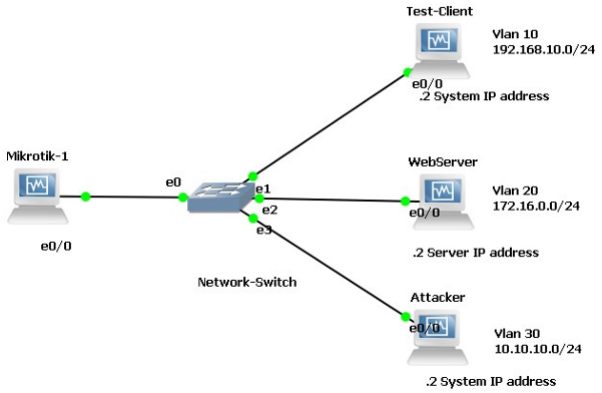


Figure 2: A particular online web server as compared to the DDoS attacks

every node is reachable. As a major research contribution of this research work, it was of utmost importance to find out the attacker on the network. Since DDoS attack prevention is only possible when an attacker is found. We have achieved this task by examining ICMP packets. Since attackers commonly use ICMP protocol to generate a DDoS attack. Through properly configuring DDoS we traced out the origin of the ICMP packet. Specifically, we traced out the TCP and IP address of the attacker by properly configuring the built-in firewall for DDoS attacks. After tracing out the IP address of the attacker, the DDoS grab the IP address of the attacker. After tracing the TCP/IP address of the attacker we made sure to enlist the attacker's IP address into the built-in firewall by setting Level-2 (L-2) and Level-3 (L-3) policies. L2 is the network portion that is specifically associated with the local area network where no routing is required. On the other side, L-3 is the network portion of which is specifically associated with the routing portion means that routing is strictly required. As a summary of the handling of the attacker through DDoS attacks, we essentially perform the following four steps importantly what we need to look at in the above code is line 5. As we know that the ground_truth output(y) is of the form [0, 0. . . . , 1.. . 0] and predicted $\hat{y}$ is of the form [0.34, 0. 03. . . . . , 0.45], we need the loss to be a single value to infer the total loss from it. For this reason, we use the sum function to get the sum of the differences/error for each value in the y and $\hat{y}$ hat vectors for that timestamp. The total_loss is the loss for the entire model inclusive of all time stamps.

To know more about the loss derivatives, please refer to this blog. There are two gradient functions that we will be calculating, one is the multiplication_backward and the other is addition_backward. In the case of multiplication _backward, we return 2 parameters, one is the gradient with respect to the weights (DLoss/DV) and the other is a chain gradient which will be a part of the chain to calculate another weight gradient. In the case of addition

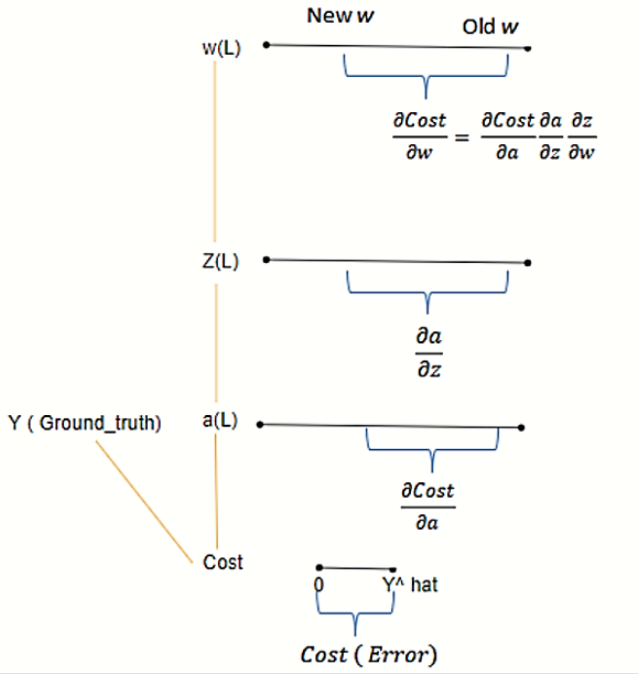


Figure 3: The total_loss is the loss for the entire model

backward while calculating the derivative we find out that the derivative of the individual components in the add function(ht_unactivated) are dh_unactivated/dU_frd= 1 as (h_unactivated = U_frd + W_frd_) and the derivative of dU_frd/dU_frd= 1. Since the cost is a function output of activation a, the change reflected by the activation is represented by dCost/da. Practically, it means the change (error) value seen from the point of view of the activation nodes. Similarly, the change of activation with respect to z is represented by DA/DZ, and z with respect to w is given by DW/DZ. We are concerned with how much the change (error) is with respect to weights. Since there is no direct relation between weights and cost, the intermediate change values from cost all the way to the weights are multiplied (as can be seen in the equation above). After configuring the network on PYTHON, it was important to test the stress level of the DDoS. Importantly, we must check whether processor usage is normal, or it increases while communicating. We have used the ping command to test the stress level of broadcasting a ping of 65000 bits/sec to verify the stress level of the DDoS attacks. The IP Ping command is to be incorporated here furthermore; it was important to verify the stress level SYN attack not only on DDoS but also on the other network nodes which are connected through DDOS Attacks.

3.3 System Architecture DDoS Attacks

The factors are data from the network history and network background of the DDoS attacks. These above-specified variables will be extracted from the given data

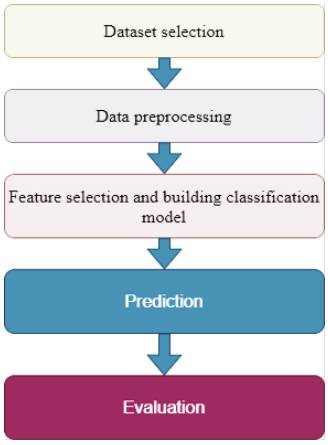


Figure 4: System architecture DDoS attacks

and will be provided to the neural network layers as the input. Then the neural network layers are also given the target output for the mapping of the input variable to the corresponding output variable by adjusting the weight of DDos Networks attacks. The Activation Function is used to ease this task. An activation function of a layer defines the output of that layer given an input or set of inputs.

- Dataset selection;
- Data preprocessing;
- Feature selection and building a classification model;
- Prediction;
- Evaluation.

3.3.1 Dataset Selection

From the provided dataset by the (CIC_DDoS2020). We have modified it by decreasing the number of dimensions in the dataset for the implementation of our DDoS network services. The data collected for the process may contain missing values, noise, or DDoS network attacks. This leads to producing inconsistent information from the process. A data process with high-quality data will produce efficient data results. The dataset after selection and understanding is loaded into Python programming language.

3.3.2 Data Preprocessing

Import the Libraries: There are many libraries we have used for this experiment.

1) **NumPy:** which is the fundamental package for scientific computing with Python.
2) **Pandas:** is for data manipulation and analysis. In particular, we have used operations for manipulating numerical data.

3) **Matplotlib:** is a Python plotting library we have used to plot the figures in an interactive environment across platforms.
4) **Seaborn:** we have used to visualize the static data to data visualization based on matplotlib because it provides informative statistical graphics.
5) **NumPy:** is used for the multidimensional array as we have used in our work to compute with and manipulate these arrays. Fancy impute is used to handle the missing values because fancy impute can be easily used to replace missing values in huge data sets.

Import the Data-set: By using the Pandas library we import our dataset and the file I used here is used firstly we use CSV files because of their lightweight. After importing the dataset, we can see the head function (This function returns the first n rows for the object based on position. It is useful for quickly testing if your object has the right type of data in it.

Missing Values: The concept of missing values is important to understand in order to successfully manage data. If the missing values are not handled properly by the researcher, that's the way we have removed the missing values and have chosen the Imputation method to handle the missing values. This method can only be used with numeric data and we are using numerical data that's the way we have chosen the Imputation method, to replace the missing values within each column separately and independently from the others.

3.3.3 Feature Selection and Building Classification Model

Splitting the dataset into Training and Test In our model we have organized as training dataset contains 99% and the testing dataset 1% from the original data and the model learns on this data to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset and got a better result.

Feature selection: It is a process of selecting the most significant features from a given dataset. In many cases, Feature Selection can enhance the performance of a deep learning model as well. There are two types of feature selections Unsupervised, and Supervised, we have chosen the Supervised type which uses the target variable (e. g. remove irrelevant variables). For the usage of the Unsupervised type of feature selection, we have chosen the "Attacks" and for the Supervised type of feature selection, we have chosen the "Normal" variable for further processing.

Feature Scaling: It is a step of Data Pre-Processing that is applied to make independent variables or features of data. It basically helps to normalize the data within a particular range. And it also helps in speeding up the calculations in an algorithm and another befit is that is generally performed during the data preprocessing steps.

3.3.4 Prediction

It is important to use the correct model from various models present because the model chosen plays a crucial role in determining the efficiency, and accuracy of the prediction system. predicting models use this data to predict whether the particular case may be a loan default case or not. However, from various models, there is no specific model which can be said as the most optimal model. DNN has better adaptability than other predicting models and this model is able to construct a non-linear model and can better predict.

3.3.5 Evaluation

In the final stage, the designed system is tested with the test set, and the performance is assured. Evolution analysis refers to the description and model of regularities or trends for objects whose behavior changes over time. Common metrics calculated from the confusion matrix are Precision; Accuracy. The calculations for the same are listed below.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True -nonPositive} + \text{False -non Negative}}$$

$$\text{F} = \frac{\text{2 x Precision x Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Accuracy} = \frac{\text{True Positives} \times \text{False -non Negative}}{\text{Total Sum of True Positive False Naative}}$$

In the reprocessing, classification, and evaluation of the traffic during Steps 4 and 5, the raw data traffic was replayed by TCP and IP Replay software in a specific DDoS port and sampled by the Flow agent for DDoS. The probability model is a conditional model over a dependent class variable with a limited number of outcomes means classes, and conditions on the feature variables F_1 to F_n.

$$P(c = F_1, \cdots, F_n)$$

If the value of n is large, basing a model is infeasible. Then we are reformulating the model then it is feasible or tractable.

$$\begin{aligned} & P(c = F_1, \cdots, F_n) \\ \rightarrow \quad & [(P(c)P(F_1, \cdots, F_n) \rightarrow (F_1, \cdots, F_n)] \end{aligned}$$

The above equation can be written in plain as follows

$$\textbf{posterior} = \textbf{(prior*like\_li\_hood)/evidence)}$$

We are only concentrating on the numerator because the denominator not depending on the class and values of

Algorithm 1 RNN usisg in DL Model

Input: Database Descriptors
Output: Selected Variables
1: Begin
2: Create empty optimized model set;
3: For i ⇐ 1 **to** Number of rounds **do**
4: Define all the Descriptor database variables as current variables
5: while True **do**
6: Split database in training and test partition;
7: Create and train the model using training data partitions;
8: Select the most important variables from the trained model;
9: Calculated the cumulative importance of variables from the trained model;
10: if max (cumulative importance of variables)¡ variables of importance threshold **then**
11: Exit loop;
12: end
13: Train the model using only the most important variables;
14: Test the trained model and calculate the accuracy;

15: if Calculated accuracy < threshold **then**
16: Exit loop;
17: end
18: Add current model to optimized model set;
19: Define the most important variables from the trained model as the current variables;
20: end

Algorithm 2 DNN using in DL Model

Input: Network attacks Characterized by type
Output: Specialized attacks based on attack intentions
Lets a set attacks originating from different source $s\_1, s\_2, s\_3 \ldots, s\_n$
Define AR_k whereby K, ≤ 1
Set AD as the conditions for attack dependencies $ad_1, ad_2, ad_3 \ldots, ad_n$
Define AD for AR_k whereby AD , $AR_K \geq 1$
Designate I as the set all probable attack intentions $i\_1, i\_2, i\_3 \ldots, i\_n$
Do Define I for each AR_k
While AR, AD is associated with AR_k
End While
End Do
End

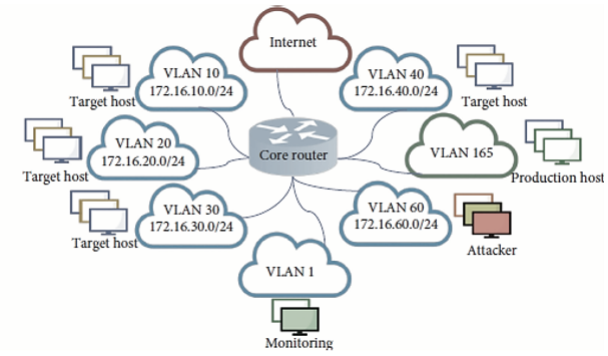


Figure 5: Customized topology, network traffic file is reprocessed on SVM-01 [16]

features F.

$$\begin{aligned} P &= TP + FP \\ R &= TP + FN \\ A &= TP + FN + FP + TN. \end{aligned}$$

Evaluation Metric From the confusion matrix table Accuracy (A) and F-measure are the metrics that are used for the evaluation of the classifier performance. F- Measure is defined in terms of Recall (R) and Precision (P). If evaluation metrics have a higher value, then the classifier is best suitable for the data set. The evaluation metrics are described effectively by a confusion matrix.

System Setup: The DDoS attacks detection system has three main parameters that directly influence its performance. The parameters shown in Table 1 allow the user to calibrate the detection system according to the operating environment. In scenarios where the DDoS attacks are too large, for example, traffic samples are discarded before processing by the classifier. On the other hand, is too small, the True Positive (TP) increases because the classifier has little data to analyze. In the case of slow False Negative (FN) DDoS, low True Negative (TN) and also False Positive (FP) reduce the attack detection rate due to in-memory flow table-2 expiration time (E_T = 2).

Evaluation Metrics: System performance was evaluated using the Precision (PREC), Recall (REC), and F-Measure (F) metrics present in the literature [8,17]. PREC measures the ability to avoid false positives, while REC-Measures system sensitivity. F is a harmonic average between PREC and REC. In this context, (i) True Positive (TP) is the attack traffic predicted correctly, (ii) True Negative (TN) is normal traffic also predicted correctly, (iii) False Positive (FP) is the normal traffic predicted incorrectly, and (iv) False Negative (FN) is the attack traffic predicted incorrectly.

The DDoS attack Detection system and the classification result were compared with the attack plan. Figure 9 summarizes the procedures carried out by the proposed validation methodology.

Figure 5: Customized topology, network traffic file is reprocessed on SVM-01 [16] The customized topology network traffic file is reprocessed on SVM-01 [16], and the Flow agent collects traffic samples and sends them to DDoS attack Detection.

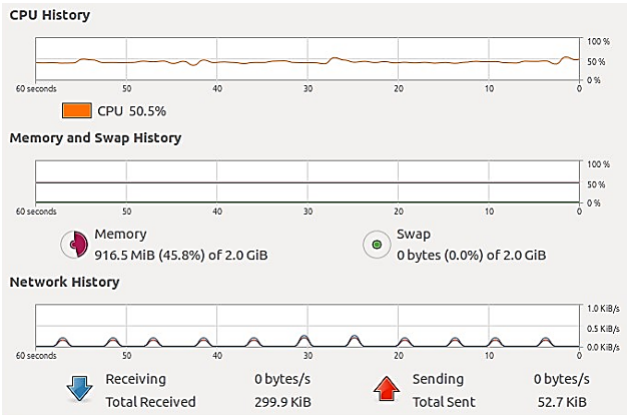


Figure 6: Through properly configuring out the origin of ICMP packets

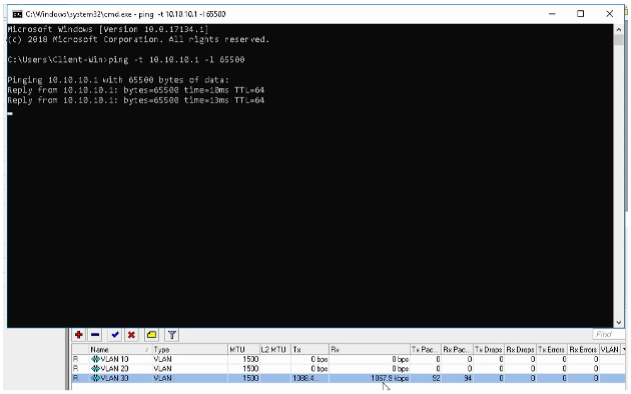


Figure 7: Single DoS attack having data burst stress level

4 Results

As a major research contribution of this thesis work, it was of utmost importance to find out the attacker on the network. Since DDoS attack prevention is only possible when an attacker is found. We have achieved this task by examining ICMP packets. Since attackers commonly use ICMP protocol to generate a DDoS attack. Therefore, we are encompassing the direct stress level of DDoS and the stress level of nodes connected through DDoS. We utilize the ping command to test the stress level of network nodes. We have verified the stress level of the Test-PC and Web server through the ping command.

The major finding of stress level on the single attack is presented in the following. It was recorded that when a single hacker is engaged in the process of a DDoS attack on DDoS, we observed 100% of CPU usage on the DDoS router see Figure 5. Additionally, it was observed that when the single hacker DDoS attack was performed on Webserver, a severe bandwidth constraint was observed as can be seen in Figure 6.

Figure 6: Through properly configuring out the origin of ICMP packets Through properly configuring DDoS we traced out the origin of the ICMP packet. Specifically, we traced out the IP address of the attacker by properly configuring the built-in firewall of DDoS. After tracing out the IP address of the attacker, the DDoS grab the IP address of the attacker. After tracing the IP address of the attacker, we made sure to enlist the attacker's IP address into the built-in firewall of DDoS by setting Level-2 (L-2) and Level-3 (L-3) policies. L2 is the network portion that is specifically associated with the local area network where no routing is required. On the other side, L-3 is the network portion which is specifically associated with the routing portion means that routing is strictly required. As a summary of handling the attacker through DDoS attacks, we essentially perform the following four steps.

1) To identify the IP address of the attacker through ICMP packets.

2) To add the IP address of the attacker into the firewall list to set the policy.

3) To perform the above two steps on L-2 (which can be referred to as a chain input).

4) To perform the above two steps on L-3 (which can be referred to as chain forward).

This contains the outcome of the efforts made in this research study. Since the chief concern of this research study was to assess the resilience of DDoS routers against D0S Attacks. DoS attack causes the router to be overloaded and makes it reach its CPU usage to 100% and this attack causes the router to be unreachable by the clients or services. Additionally, in this research study, the DDoS_FLOOD attack which is one of the famous DoS attacks has been formulated on the DDoS router. The effectiveness of the DDoS_FLOOD attack on the DDoS router has been tested by properly observing the stress level of its processor in the PYTHON simulation environment. Furthermore, results have been taken precisely and presented in the following section for analysis.

4.1 Single DoS Attack

The impact of a single DoS attack (SYN attack) was measured on CPU stress and network bandwidth. In Figure 5. it can be seen that a single DoS attack put 2% stress on RouterOS, This stress level can be increased in many folds if a single DoS attack is converted into multiple DoS attacks and or if Data burst packet through broadcasting is increased. Additionally, the load on the network has also been verified which is 500 kbps on a normal single DDoS attack. It is pertinent to mention that this network load can be increased in many folds when it comes to multiple DDoS attacks.

After performing a single DDoS attack and a Double DDoS attack and it was required to check the resilience of the DDoS router on script attack. To perform the script attack we developed a script having a constant loop for increasing data bursts of 65000 each. As can be seen in Figure 5, a script can be seen to perform a double DDoS attack.

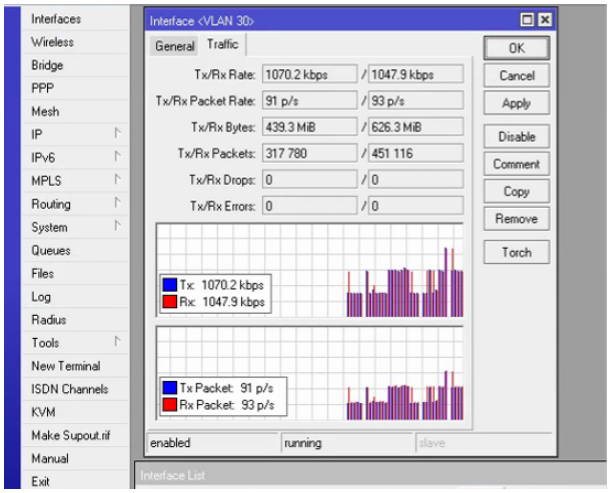


Figure 8: Distributed denial-of-service attack deliver connectivity to nodes and users on the given network

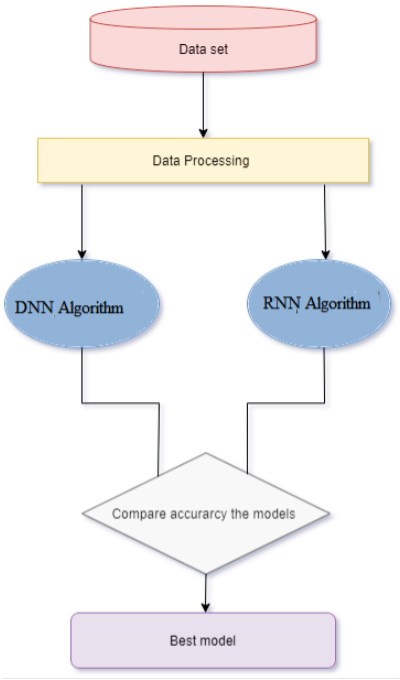


Figure 9: Proposed model for DDoS attacks dataset

Network devices are using a vital role to deliver connectivity to nodes and users on any given network. Network policies perform the additional task of shielding facilities and users from known and unknown attacks. This feature of network policies to stop or curtail network attacks in order to make the nodes and all attached devices secure needs further research studies and experimentation to confirm their resilience against potential known attacks. The Distributed Denial-of-Service Attack (DDoS Attack) is known as one of the deadliest attacks which make network services and/or devices completely unavailable. DDoS routers are very well known for their performance and functionalities among their peers.

In this research study, the resilience of the DDoS router is going to be tested against DDoS attacks. The DDoS attack causes the router to be overloaded and makes it reach its CPU usage to 100% and this attack causes the router to be unreachable by the clients or services.

The four important data pre-processing techniques are data cleaning, data integration, data reduction, and data transformation. Here feature selection has a major role in preprocessing to select suitable features that affect the accuracy of the algorithm and it comes under data reduction. The data sets are implemented using the Deep Neural Network (DNN) model and the Recurrent Neural Network RNN model. The comparison of accuracies obtained from the two models was made and arrived at the conclusion was that the loan credibility behavior of DDoS network users can be predicted more accurately using the proposed model.

Table 3 showed that precision, recall, precision, f score, and accuracy are show attack and normal scores.

The proposed DL system model can reduce the data dimensionality by automatically extracting the features from input data. we also use various metrics to evaluate our proposed model, such as precision, recall, precision, F-score, and accuracy, to have a systematic benchmarking analysis with other related.

Figure 10 is compared to show learners of RNN and DNN algorithms, and techniques achieved the best performance in terms of precision, recall, F-score, and accuracy of the RNN 99% and DNN 89. 78%. The ability of the ANN proposed model to deal with a high ratio of complex nonlinear relationships makes them promising techniques for detecting network intrusion. It can be used to tackle the limitation of the traditional classification methods, which are implemented to identify the anomalies in traffic based on the domain of the services. this paper is to represent the potential of the ANN proposed model for anomaly detection systems. We achieved a new technique based on RNN-Autoencoder classified the input traffic into normal or malicious types of URLs

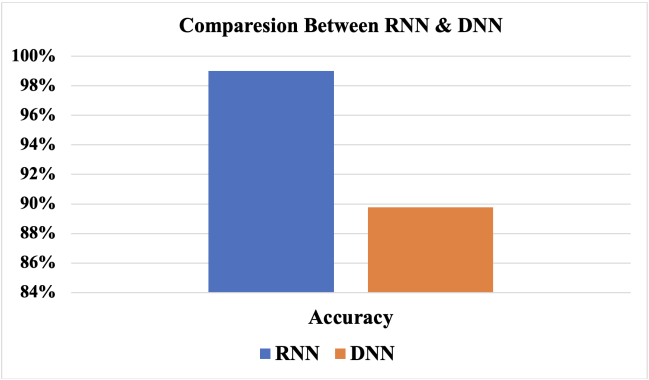


Figure 10: Compared between RNN and DNN for the DDoS Attacks

Table 2: Attack and normal values

	Attack	Normal
Attack	0.99	0.99
Normal	0.01	0.99

Table 3: Precision, Recall, Precision, F score, and Accuracy are showing attack and normal scores

Techniques	Precision		Recall		precision		Accuracy
	Attack	Normal	Attack	Normal	Attack	Normal	
DNN	**0.63**	0.74	**0.54**	0.67	0.56	0.70	**89. 78%**
RNN	0.99	**1.00**	0.99	**0.99**	0.99	0.99	**99%**

F-score, and the accuracy of the RNN is 99%.

5 Conclusion

This article is presented the DDoS attack Detection (ANN) proposed model system, an online approach to a DDoS attacks detection system. The internet network users and show learners compared RNN and DNN algorithms, and techniques achieved the best performance in terms of precision, recall, F-score, and accuracy of the RNN is 99%, and then DNN is 89. 78%, is classified network traffic based on samples taken by the flow protocol directly from network devices. In this research study, the resilience of the router is going to be tested against DDoS Attacks. The DDoS attacks cause the router to be overloaded and make it reach its CPU usage to 100% and this attack causes the router to be unreachable by the clients or services. Not only that but it causes all operations on packets performed by the router CPU such as packet filtering, TCP/IP ping, and logging, queuing may also cause overloading of the router. Particularly in this research study, DDoS attacks that are focused on DDoS attacks have been formulated on DDoS routers. The DDoS attacks have been tested in the PYTHON simulation environment and a physical environment and results have been taken for analysis and further processing which are implemented and identified the anomalies traffic based on the domain of the services. This research paper has represented the potential of DL techniques for anomaly detection (DL) proposed model system and we are achieved a new DL technique based on RNN-Autoencoder are classified the input traffic into normal or malicious types of URLs. This research study has fulfilled the gap in testing identified security DDoS network services against DDoS attacks and will help the research community to develop new mechanisms to make the routers more powerful against identified security, and DDoS attack holes systems. We achieved a new DL technique based on RNN-Autoencoder classified the input traffic into normal or malicious types of URLs.

6 Future Work

Furthermore, the major achievement of this research work is to secure the webservers through a firewall by hiding TCP/IP ping on the web services. If the webserver cannot be pinged then it is more secure. The most important thing to achieve is that with the current configuration simulation, we can track down the IP address of the hacker and can set policies.

References

[1] M. Abdelhaq, R. Alsaqour, M. Alaskar, F. Alotaibi, R. Almutlaq, B. Alghamdi, B. Alhammad, M. Sehaibani, D. Moyna, "THE resistance of routing protocols against DDOS attack in MANET," *International Journal of Electrical & Computer Engineering*, vol. 10, no. 5, 2020.

[2] J. Cheng, M. Li, X. Tang, V. S. Sheng, Y. Liu, and W. Guo, "Flow correlation degree optimization driven random forest for detecting DDoS attacks in cloud computing," *Security and Communication Networks*, pp. 1-14, 2018.

[3] S. Choi, Y. An, and I. Sasase, "A lightweight detection using bloom filter against flooding DDoS attack," *IEICE Transactions on Information and Systems*, vol. 103, no. 12, pp. 2600-2610, 2020.

[4] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," in *International Conference on Machine Learning*, pp. 1019-1028, 2017.

[5] M. S. Elsayed, N. A Le-Khac, S. Dev, and A. D Jurcut, "Ddosnet: A deep-learning model for detecting network attacks," in *IEEE 21st International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM'20)*, pp. 391-396, 2020.

[6] J. Gera, B. P Battula, "Detection of spoofed and non-spoofed DDoS attacks and discriminating them from flash crowds," *EURASIP Journal on Information Security*, pp. 1-12, 2018.

[7] N. Hoque, D. K. Bhattacharyya, and, J. K. Kalita, "MIFS-ND: A mutual information-based feature selection method," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6371-6385. 2014.

[8] F. S. D. Lima Filho, F. A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, L. F. Silveira, "Smart detection: An online approach for DoS/DDoS attack detection using machine learning," *Security and Communication Networks*, vol. 2019, Article ID 1574749, pp. 1-15, 2019. (`https://doi.org/10.1155/2019/15747492019`)

[9] M. Pokrinchak, and, M. M. Chowdhury, "Distributed denial of service: Problems and solutions," in *IEEE International Conference on Electro Information Technology (EIT'2)*, pp. 032-037, 2021.

[10] R. Panigrahi, and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," *International Journal of Engineering & Technology*, vol. 7, no. 3, pp. 479-482.2018.

[11] S. H. Park, J. M. Goo, and C. H. Jo, "Receiver operating characteristic (ROC) curve: practical review for radiologists," *Korean Journal of Radiology*, vol. 5, no. 1, pp. 11-18. 2004.

[12] D. M Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint*, arXiv:2010.16061, 2020.

[13] K. M. Prasad, A. R. M. Reddy, and K. V. Rao, "BARTD: Bio-inspired anomaly based real time detection of under-rated App-DDoS attack on web," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 1, pp. 73-87, 2020.

[14] J. D. Rennie, "Tackling the poor assumptions of naive bayes text classification," in *International Conference on Machine Learning (ICML'03)*, 2003.

[15] K. S. Sahoo, B. K. Tripathy, K. Naik, S. Ramasubbareddy, B. Balusamy, M. Khari, and D. Burgos, "An evolutionary SVM model for DDOS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 132502-132513, 2020.

[16] A. Saied, R. E Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," *Neurocomputing*, vol. 172, pp. 385-393, 2016.

[17] G. F. Scaranti, L. F. Carvalho, S. B. Junior, J. Lloret, M. L. Proença Jr, "Unsupervised online anomaly detection in Software Defined Network environments," *Expert Systems with Applications*, vol. 191, p.116225, 2022.

[18] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, pp. 108-116, 2018.

[19] S. Simpson, S. N. Shirazi, A. Marnerides, S. Jouet, D. Pezaros, and D. Hutchison, "An inter-domain collaboration scheme to remedy DDoS attacks in computer networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 879-893, 2018.

[20] J. R. Sun, M. S. Hwang, "A new investigation approach for tracing source IP in DDoS attack from proxy server", in *Intelligent Systems and Applications*, pp. 850-857, 2015.

[21] S. Ullah, M. A. Khan, J. Ahmad, S. S. Jamal, Z. Huma, M. T. Hassan, N. Pitropakis, W. J. Buchanan, "HDL-IDS: A hybrid deep learning architecture for intrusion detection in the internet of vehicles," *Sensors*, vol. 22, no. 4, p. 1340, 2022.

[22] A. Verma, R. Saha, N. Kumar, G. Kumar, "A detailed survey of denial of service for IoT and multimedia systems: Past, present and futuristic development," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19879-19944, 2022.

[23] C. D. Xuan, M. H. Dao, "A novel approach for APT attack detection based on a combined deep learning model," *Neural Computing and Applications*, vol. 33, pp. 13251-13264, 2021.

[24] B. Zhang, C. Shen, B. Bealmear, S. Ragheb, W. C. Xiong, R. A. Lewis, R. P. Lisak, and L. Mei, "Autoantibodies to agrin in myasthenia gravis patients," *PloS One*, vol. 9, no. 3, p. e91816, 2014.

Biography

Sirajuddin Qureshi received his bachelor's degree in Computer Sciences from Quaid-e-Awam University of Engineering, Science & Technology, Pakistan. Afterwards, he pursued his Master's in Information Technology from Sindh Agricultural University Tandojam, Pakistan. Currently he is pursuing PhD in Information Technology at Beijing University of Technology, China. He has nine research publications to his credit as main author and co-author, which featured national and international journals and conferences. Sirajuddin's research areas includes but not limited to Network Forensics Analysis, Digital Forensics, Cyber security, Computer Networks and Network Security.

Jingsha He received the bachelor's degree in computer science from Xi'an Jiaotong University, China, and the master's and Ph. D. degrees in computer engineering from the University of Maryland, College Park, MD, USA. He worked for several multinational companies in USA, including IBM Corp. , MCI Communications Corp. , and Fujitsu Laboratories. He is currently a Professor with the Faculty of Information Technology, Beijing University of Technology(BJUT), Beijing. He has published more than ten articles. He holds 12 U. S. patents. Since August 2003, he has been published over 300 papers in scholarly journals and international conferences. He also holds over 84 patents and 57 software copyrights in China and authored nine books. He was a principal investigator of more than 40 research and development projects. His research interests include information security, wireless networks, and digital forensics.

Saima Tunio received the BSIT(Hons) with gold medal from Sindh Agricture University Tandojam, Pakistan.

Afterwards, she pursued her MSIT from Isra University Hyderabad, Pakistan. Currently she is pursuing PhD in Information Technology at Beijing University of Technology, China. She has more than five research publications to her credit as main author and co-author, which featured national and international journals and conferences. Saima's research areas includes but not limited to Information security. IoT security, Digital Forensics, Cyber security, Computer Networks.

Nafei Zhe received the B. S. and M. S. degrees from Central South University, China, in 2003 and 2006, respectively, and the Ph. D. degree in com- puter science and technology from the Beijing University of Technology, Beijing, China, in 2012. She was a Postdoctoral Research Fellow with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, from 2015 to 2017. She is currently an Associate Professor with the Faculty of Information Technology, Beijing University of Technology. She has published over 20 research papers in scholarly journals and international conferences. Her research interests include information security and privacy, wireless communications, and network measurement.

Ahsan Nazir has received his M. Sc degree from University of Engineering and Technology Lahore in 2016. From September 2015 to August 2018 he worked as software Engineer at Dunya Media group Lahore since September 2018 he is doing PhD in Software Engineering from Beijing University of Technology , Beijing China . He has published more than 10 journals and conference papers . His area of research include eGovernment, IoT, Software Engineering and Machine learning applications.

Ahsan Wajahat received the B. S. and M. S degrees in information technology from the Sindh agriculture University, Pakistan, in 2012 and 2016, respectively. He is currently pursuing the Ph. D. degree with the Beijing University of Technology, Beijing, China. His research interests include machine learning, information security, forensic network and data mining. He has received awards and honors from the China Scholarship Council (CSC), China.

An Improved Software Defined Network Detection Algorithm for Real-Time Detection and Anomaly Identification of Network Traffic

Ke Zhang

(Corresponding author: Ke Zhang)

Shaanxi Institute of Mechatronic Technology, Baoji 721001, Shaanxi, China

Email: kz562122887@yeah.net

(Received July 10, 2022; Revised and Accepted June 13, 2023; First Online Aug. 25, 2023)

Abstract

The Internet is used in all aspects of people's lives, and the scale of the network is expanding, but every network that people access may have abnormal network traffic, which brings hidden dangers to people's lives. This paper combined K nearest neighbors (KNN) with a support vector machine (SVM) to form the KNN-SVM algorithm. Then, the information gain method was used to obtain the feature vectors of network traffic in the dataset, and the KNN-SVM algorithm was used to detect and identify the network traffic in the dataset. The experimental results showed that the accuracy, precision, and recall rate of the KNN-SVM algorithm were over 90% in identifying different categories of abnormal traffic; in identifying network traffic with different features, the combination of all features had the best identification result; compared with other algorithms, such as SVM and decision trees, the KNN-SVM algorithm not only had higher detection accuracy but also took less time to detect. The results show that the KNN-SVM algorithm has excellent detection and recognition performance and high detection efficiency, which is very suitable for real-time detection and anomaly recognition of network traffic.

Keywords: Anomaly Traffic Identification; K Nearest Neighbor; Real-Time Detection; Software Defined Network; Support Vector Machine

1 Introduction

As Internet technology continues to evolve, researchers have proposed software defined network (SDN) to solve problems such as configuration difficulties. However, the emergence of SDN also brings new network security problems, so it is necessary to protect the network by detecting anomalies and preventing network intrusion [1]. Traditional detection methods cannot guarantee the traffic detection accuracy while considering the detection rate, so how to perform real-time detection and anomaly identification for network traffic in SDN environment has become another new research topic for researchers. The related literature on network traffic detection in SDN is reviewed. Zhang *et al.* [16] put forward a network-wide forwarding anomaly detection and localization method FOCES in SDN. The experimental results show that the proposed method achieved more than 90% accuracy when the packet loss rate was less than or equal to 10%, and the localization accuracy reached about 80% when the packet loss rate was less than or equal to 5%.

Tang *et al.* [14] designed and implemented an online attack detection and mitigation system (ADMS) framework. The final assessment of the prototype ADMS implementation showed that the framework was able to accurately identify and effectively mitigate low-rate denial-of-service attacks in real time. Jafarian *et al.* [7] developed an organizational structure for detecting anomalies in SDN environments through information gain ratio and integrated learning scheme (stacking). The experimental results showed that the suggested technique outperformed other methods in terms of improving accuracy and detection rate, lowering classification error, and decreasing false alarm rate.

DeepGuard, a framework for effective anomaly detection presented by Phan *et al.* [12], improved the effectiveness of network attack detection in SDN-based networks by using a fine-grained traffic monitoring technique. Through a lot of experiments, it was shown that DeepGuard significantly outperformed previous traffic matching techniques in terms of performance at the traffic granularity level. For edge computing-based architecture in Internet of Things networks, Qureshi *et al.* [13] put forward a SDN-based anomaly detection system. Simulation experiments showed that the system had better performance in case of different performance parameters. In order to enhance the accuracy of network traffic anomaly detection in SDN, this paper combined K-nearest neigh-

bor (KNN) algorithm and support vector machine (SVM) algorithm to obtain a KNN-SVM algorithm for network traffic detection and classification in datasets. It was compared with other detection algorithms to verify its effectiveness and superiority. This paper aims to provide an approach for the subsequent real-time detection of abnormal SDN traffic.

2 SDN Detection Algorithm

2.1 SDN Architecture

SDN is a new innovative architecture for networking [4], which includes three layers, namely data layer, control layer and application layer. The SDN is introduced layer by layer. The data layer is mainly composed of multiple network devices, such as switches and routers. The control layer is the most critical part of the whole network, and its core is the SDN controller. Through the controller, all switch devices can be seen, and it is the brain of managing the network in the SDN. SDN can use its core technology, Open Flow protocol, to separate the control layer and data layer of network devices [6]. The application layer consists of various types of network applications, which are mainly designed to provide services to users. Users implement their needs through programming. Meanwhile, the SDN architecture allows administrators to program and control the control part of the network directly to achieve flexible management and control of all network devices.

2.2 Network Traffic Detection Algorithm

By reviewing the related literature on network traffic detection, it is found that SVMs [8], random forests [10], decision trees [3], and neural networks [17] are commonly used in machine learning-based network traffic anomaly detection algorithms. Since the results of the KNN algorithm are concise and easy to understand, it is well suited for solving classification problems, so it is used in this paper. The principle of this algorithm is to classify samples by calculating the distance between sample data after data statistics. The distance calculation method used in this paper is the Euclidean distance, and its expression is:

$$d(x,y)=\sqrt{\sum_{i=1}^{N}(x_i-y_i)^2}$$

However, often individual algorithms have their own limitations in detection effect. The KNN algorithm needs to calculate the distance between all data and then classify them according to the distance size, which makes the operation time become longer. In order to better detect network traffic in real time, this paper finally considers the combination of KNN algorithm and SVM algorithm and proposed to use the KNN-SVM algorithm to perform real-time network traffic detection and anomaly identification. The difference between the two algorithms is that the KNN algorithm determines the value of the target point by selecting the proximity of the target point, while the SVM algorithm determines the category to which the target point belongs by dividing the region. The combination of the two algorithms can simultaneously solve the problem of inaccurate classification of the SVM algorithm when it is close to the hyperplane, and the problem of long computation time of the KNN algorithm. The main core idea and steps of this combined algorithm in this research paper are as follows.

1) The traffic data to be identified are input, and the corresponding support vector sets and SVM classifiers are calculated by the SVM algorithm.

2) Distance threshold ϵ between the data to be classified and the hyperplane in the SVM algorithm is set as 1.

3) The support vector set obtained by training using the KNN algorithm to obtain a KNN model;

4) For traffic data X_i to be identified, the distance from the SVM hyperplane is calculated, i.e., $P(X_i) = WX_i + b$, where W and b represent the coefficient and hyperplane intercept of the SVM algorithm detection model, respectively.

5) If the calculation result is $P(X_i) < \epsilon$, then the detection result of the SVM algorithm is output directly. If the calculation result is $P(X_i) > \epsilon$, the algorithm is replaced, and the KNN model is used to detect and identify the traffic data.

2.3 Feature Vector Selection

Before performing anomaly detection and identification work on network traffic, in order to detect anomalous traffic more accurately, network traffic data processing is first required, which involves extracting a certain class of features that appear multiple times or individually in network traffic, so that the extracted traffic feature vector can be used by the detection algorithm. In this study, the information gain method [5] is used to select the features in the dataset. The so-called information gain is the entropy. The larger the information gain value, the more information the feature contains, the smaller the uncertainty, and the more important the feature. Suppose dataset A contains N categories of features in total, and the proportion of samples of the i-th category in A is C_i. The formula of information entropy is:

$$I(A)=-\sum_{i=1}^{N}C_i\log_2(C_i).$$

The empirical conditional entropy obtained by further dividing dataset A according to feature T is:

$$I(A|T)=\sum_{i=1}^{N}\frac{|A_i|}{|A|}I(A).$$

The final formula for the information gain is:

$$G(T) = I(A|T) - I(A).$$

3 Example Analysis

3.1 Experimental Platform and Data

In order to verify the detection effect of the algorithm proposed in this paper, the study built up a traffic detection platform for SDN using Ryu controller, Openflow switch and Mininet topology simulation tool [2]. The data for the experiments were mainly from the KDDCUP-99 dataset [15], which is the most widely used in network traffic detection, and includes both normal network traffic and attack network traffic. The abnormal network traffic in the KDDCUP-99 dataset was divided into four major categories and 39 attack types.

Table 1 shows some of the attack types. In this paper, a total of 10,000 network traffic were selected from the KDDCUP-99 dataset, including 4,000 normal traffic, 2,000 denial of service (DoS) abnormal traffic, 2,000 probe abnormal traffic, 1,500 remote to local (R2L) abnormal traffic, and 500 user to root (U2R) abnormal traffic. The data were divided into training set and test set according to the ratio of 7:3 for experiments.

Table 1: KDDCUP-99 dataset abnormal traffic types

Category	Type of attack
DoS	Ping-of-death, neptune, mailbomb, pod
Probe	Port-scan, Satan, ping-sweep, Saint
R2L	Guessing password, Imap, Ftp write, Spy
U2R	Buffer overflow attacks, perl, SQL attack

There were 41 features in the KDDCUP-99 dataset, which were coded from 0 to 40. The information gain value of each feature was calculated, and the selection was based on the magnitude of the calculated value. The ten features with the largest information gain values were selected and input to the algorithm for training, as shown in Table 2.

3.2 Evaluation Indicators

In order to better verify the detection results of the KNN-SVM algorithm, decision tree and SVM algorithms were also selected to simultaneously detect and identify the network traffic, and the detection and identification results were compared. In view of the detection effectiveness of abnormal network traffic mainly includes two aspects: time efficiency and correct rate; therefore, the algorithm evaluation indicators were determined as accuracy, precision, recall rate, and detection time [9] for result comparison and analysis.

1) Accuracy, representing the proportion of correctly classified network traffic to total traffic:

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

2) Precision, representing the proportion of network traffic correctly classified as positive classes to the traffic predicted as positive classes:

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

3) Recall rate, representing the proportion of network traffic correctly classified as positive classes to the total traffic in all positive classes:

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

3.3 Analysis of Results

The choice of the K value in the KNN algorithm determines the final result of the algorithm in detecting and identifying network traffic. To obtain the best traffic detection results, the best K value should be chosen. In the experiments, the final K value was first determined by finding the highest accuracy through training the algorithm under different K values. From the data in Figure 1, it can be seen that the KNN-SVM algorithm had the highest accuracy when the K value was 30. Then, as the K value increased, the accuracy tended to decrease. Therefore, the K value for the KNN-SVM algorithm was set to 30 in the experiments.

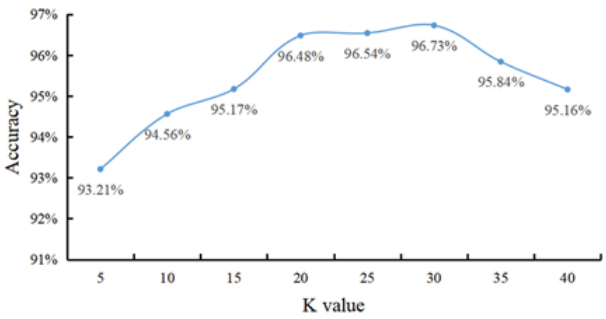


Figure 1: Variation of detection accuracy of the KNN-SVM algorithm under different K values

According to the introduction of the dataset above, it is known that the dataset contained five types of data, namely, normal traffic, DoS, probe, R2L, and U2R. The purpose of this study is to better detect the abnormal traffic, so the results of the KNN-SVM algorithm in detecting and identifying the four types of abnormal traffic in the dataset were analyzed separately, and Figure 2 is obtained. It can be seen from Figure 2 that the accuracy of the algorithm in detecting DoS and probe types of abnormal traffic was over 98%, and the precision and recall

Table 2: Results of the selection of flow characteristics

Code	Feature name	Description	Information gain
1	duration	The length of time this network has been connected	0.57
2	protocal_type	The type of the network connection protocol	0.63
4	flag	The status of the network connection protocol	0.49
6	dst_bytes	The data traffic from the destination host to the source host	0.51
11	num_failed_logins	The number of failed login attempts for this network	0.58
24	serror_rate	The percentage of SYN errors in connections to the same destination host	0.64
26	rerroe_rate	The percentage of REJ errors in connections to the same destination host	0.65
32	dst_host_srv_count	The number of connections to the same destination host and service	0.52
34	dst_host_diff_srv_rate	The percentage of connections to different services and the same destination host	0.55
35	dst_host_same_src_port_rate	The percentage of connections to the same destination host and the source port	0.48

rate were close to 100%. However, for R2L and U2R attacks, the accuracy was below 95%, and the precision and recall rate were much lower than those of DoS and probe. The reason for this is that the sample data of R2L and U2R were relatively small, which made the detection of R2L and U2R difficult to some extent. However, overall, the KNN-SVM algorithm had good results in identifying anomalous network traffic, and the accuracy, precision, and recall rate of the algorithm for four different categories of anomalous network traffic were all above 90

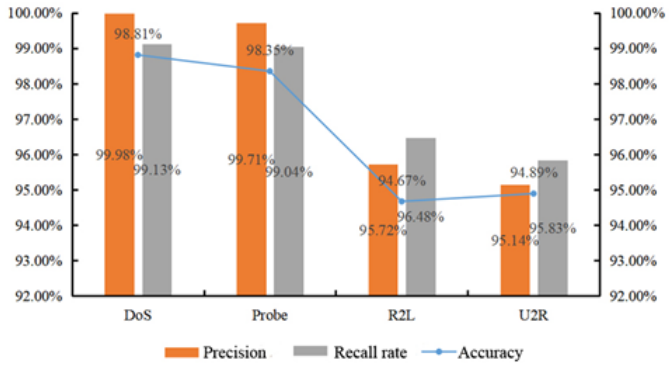


Figure 2: Detection and identification results of different abnormal traffic categories

To ensure the validity of the algorithm, a tenfold cross-validation [11] was performed on the dataset, and the experimental results obtained for the different features are shown in Table 3. From the values in Table 3, it can be seen that the values of the three evaluation indicators were the highest when all the features were combined for the detection and identification of the algorithm, which were 96.73%, 97.69%, and 97.86%, respectively, and this meant that the detection results obtained were the best. It was found that whether the detection of network traffic was performed by a single feature or a combination of ten features, the recognition accuracy was over 90%. It is verified that the KNN-SVM algorithm has high accuracy for detecting and identifying anomaly network traffic, and the use of this algorithm is effective and feasible.

In addition, the accuracy, precision, and detection time of the KNN-SVM algorithm were also compared with other algorithms, including SVM and decision tree algorithms, and Table 4 was obtained. From the accuracy and precision data in Table 4, it can be seen that the KNN-SVM algorithm had the highest accuracy and precision values, 96.73% and 97.69%, respectively, followed by the decision tree algorithm, which also reached 90% or more, while the SVM algorithm had the lowest accuracy and precision, 89.74% and 91.35%, respectively. In terms of the detection time of the algorithms, the KNN-SVM algorithm took the shortest time, only 78.22 s, i.e., it had the highest detection efficiency. This result showed that the KNN-SVM algorithm had excellent detection and recognition performance and high detection efficiency, which is very suitable for real-time detection and anomaly recognition of network traffic.

4 Conclusion

This paper briefly introduced the SDN and the KNN-SVM algorithm. The study designed and implemented an anomalous network traffic detection model based on the SDN using the KNN-SVM algorithm to identify the anomalous network traffic within the dataset. Finally, the detection results were compared with other detection algorithms to demonstrate the reliability of the KNN-SVM

Table 3: Tenfold cross-validation results for different features

Feature name	Average accuracy	Average precision	Average recall rate
duration	94.17%	95.24%	96.08%
protocal_type	94.56%	95.13%	95.94%
flag	96.21%	96.88%	97.12%
dst_bytes	95.83%	96.25%	96.74%
num_failed_logins	96.54%	97.19%	97.57%
serror_rate	95.68%	96.27%	96.95%
rerroe_rae	96.24%	96.75%	97.18%
dst_host_srv_count	95.15%	96.31%	96.96%
dst_host_diff_srv_rate	96.39%	97.24%	97.67%
dst_host_same_src_port_rate	96.22%	97.10%	97.48%
The combination of all features	96.73%	97.69%	97.86%

Table 4: Detection results of network traffic by different detection algorithms

	Overall accuracy	Overall precision	Detection time/s
The KNN-SVM algorithm	96.73%	97.69%	78.22
The SVM algorithm	89.74%	91.35%	117.14
The decision tree algorithm	93.61%	94.92%	91.46

algorithm. The experimental results showed that the accuracy, precision, and recall rate of the KNN-SVM algorithm were over 90% in identifying different categories of anomalous traffic; when identifying network traffic using different features, the best detection result was obtained by combining all the features; compared with algorithms such as SVM and decision tree algorithms, the KNN-SVM algorithm not only had higher detection accuracy, but also took less time to detect. These results show that the KNN-SVM algorithm has excellent detection and recognition performance and high detection efficiency, which is very suitable for real-time detection and anomaly recognition of network traffic.

References

[1] W. A. Ali, K. N. Manasa, M. F. Aljunid, M. Bendechache, P. Sandhya, "A review of current machine learning approaches for anomaly detection in network traffic," *Journal of Telecommunications and the Digital Economy*, vol. 8, no. 4, pp. 64-95, 2020.

[2] O. Alssaheli, Z. Z. Abidin, N. A. Zakaria, Z. A. Abas, "Implementation of network traffic monitoring using software defined networking ryu controller," *WSEAS Transactions on Systems and Control*, vol. 16, pp. 270-277, 2021.

[3] N. A. Awad, "Enhancing network intrusion detection model using machine learning algorithms," *Computers, Materials and Continua*, vol. 67, no. 1, pp. 979-990, 2021.

[4] A. B. Dehkordi, M. R. Soltanaghaei, F. Z. Boroujeni, "A new DDoS detection method in software defined network," *IEEE Transactions on Industry Applications*, vol. PP, no. 99, pp. 1-11, 2020.

[5] R. H. Dong, H. H. Yan, Q. Y. Zhang, "An intrusion detection model for wireless sensor network based on information gain ratio and bagging algorithm," *International Journal of Network Security*, vol. 22, no. 2, pp. 218-230, 2020.

[6] S. Gajanand, S. Himanshu, P. Rajneesh, G. Nidhi, S. R. Shanker, K. Ashutosh, "Self-healing topology for DDoS attack identification & discovery protocol in software-defined networks," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 8, pp. 2221-2232, 2021.

[7] T. Jafarian, M. Masdari, A. Ghaffari, K. Majidzadeh, "Security anomaly detection in software-defined networking based on a prediction technique," *International Journal of Communication Systems*, vol. 33, no. 14, pp. e4524.1-e4524.23, 2020.

[8] M. Jain, V. Saxena, "An ECOSVS-based support vector machine for network anomaly detection," *International Journal of Data Analysis Techniques and Strategies*, vol. 14, no. 1, pp. 32-54, 2022.

[9] R. Kumar, S. Singla, "Multiclass severity classification for software bugs using support vector machine, k-nearest neighbor, decision tree and nave bayes," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 11, pp. 5109-5112, 2020.

[10] Z. Noshad, N. Javaid, T. Saba, Z. Wadud, M. Q. Saleem, M. Alzahrani, O. E. Sheta, "Fault detection

in wireless sensor networks through the random forest classifier," *Sensors*, vol. 19, no. 7, pp. 1-21, 2019.

[11] D. R. Patel, V. Vakharia, M. B. Kiran, "Texture classification of machined surfaces using image processing and machine learning techniques," *FME Transactions*, vol. 47, no. 4, pp. 865-872, 2019.

[12] T. V. Phan, T. G. Nguyen, N. N. Dao, T. T. Huong, N. H. Thanh, T. Bauschert, "DeepGuard: Efficient anomaly detection in SDN with fine-grained traffic flow monitoring," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1349-1362, 2020.

[13] K. N. Qureshi, G. Jeon, F. Piccialli, "Anomaly detection and trust authority in artificial intelligence and cloud computing," *Computer Networks*, vol. 184, no. Jan.15, pp. 1-14, 2020.

[14] D. Tang, X. Wang, Y. Yan, D. Zhang, H. Zhao, "ADMS: An online attack detection and mitigation system for LDoS attacks via SDN," *Computer Communications*, vol. 181, no. Jan., pp. 454-471, 2022.

[15] X. Yu, Z. Tian, J. Qiu, S. Su, X. Yan, "An intrusion detection algorithm based on feature graph," *Computers, Materials and Continua*, vol. 61, no. 1, pp. 255-273, 2019.

[16] P. Zhang, F. Zhang, S. Xu, Z. Yang, H. Li, Q. Li, H. Wang, C. Shen, C. Hu, "Network-wide forwarding anomaly detection and localization in software defined networks," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 332-345, 2020.

[17] Y. Zhang, X. Chen, D. Guo, M. Song, Y. Teng, X. Wang, "PCCN: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows," *IEEE Access*, vol. 7, pp. 119904-119916, 2019.

Biography

Zhang Ke was born in Fuping, Shaanxi Province, China. Associate professor of Shaanxi Institute of Mechatronic Technology. In 1997 graduated from Xi'an University of Architecture and Technology with a bachelor's degree .Obtained a master's degree in Engineering from 2010 to 2013 at Xi'an University of Technology. He has presided over one second prize of the "Teaching Achievement Award" of the Shaanxi Provincial People's Government and has served as the chief editor or co-editor of more than ten computer-related textbooks. He has published multiple academic papers. His main teaching and research areas include computer networks, software technology, and digital education.

Cryptanalysis and Improvement of a Multi-factor Authenticated Key Exchange Protocol

Zhiqiang Ma[1,2] and Jun He[2]
(Corresponding author: Jun He)

College of Computer and Cyber Security & Fujian Normal University[1]
School of Computer Science and Engineering & Chongqing University of Technology[2]
69 Hongguang Avenue, Banan District, Chongqing 400054, China
Email: hejun@cqut.edu.cn

(Received Nov. 3, 2022; Revised and Accepted June 20, 2023; First Online Aug. 25, 2023)

Abstract

Authenticated key exchange (AKE) protocol is one of the most fundamental cryptographic primitives for secure communication systems. It allows two parties to securely establish a common session key over an insecure public network. Recently, Zhang *et al.* proposed a multi-factor authenticated key exchange (MFAKE) protocol for mobile communications. This paper presents the cryptanalysis of Zhang's MFAKE protocol. We find out Zhang's MFAKE protocol has a security flaw that renders it insecure against Man-in-the-Middle (MITM) attacks and outsider Key Compromise Impersonation (KCI) attacks. We present a simple case of MITM attacks and illustrate how an adversary impersonates the client to the server if just compromising the key of the server. And an improved MFAKE protocol is proposed to overcome the weakness of Zhang's MFAKE protocol with minimum changes. Then we give the formal security proof of our improved MFAKE protocol in the random oracle model. The security features and performance of our improved protocol are compared with related protocols. The results show that our improved MFAKE protocol is more secure and efficient.

Keywords: Authenticated Key Exchange; Key Compromise Impersonation Attack; Multi-factor

1 Introduction

With the rapid development of communication technologies, mobile devices have become popular in daily life. Advances in mobile telecommunication technology lay the foundation for accessing critical infrastructure. (e.g., industrial manufacturing, energy, healthcare, transportation). People interact with these systems to obtain personal services. However, adversaries could intercept, modify or replay messages, as well as impersonate a legal user to access the protected resource. Communication security has become one of the most crucial issues when accessing critical infrastructure for services [13, 25]. To prevent unauthorized access, authentication is a dominant form of access control in various services.

Authenticated key exchange (AKE) protocol allows two parties to share a common session key for secure communication over insecure public channels and verify the legitimacy of each other. Legitimate access to any information system requires authentication of the user accessing the protected information. Thus, password-based authenticated key exchange (PAKE) protocols [4,5,8,15,17] have received significant attention in user authentication systems. PAKE protocols assume a realistic scenario in which secret keys are not uniformly distributed over a large keyspace, but chosen from a small and low-entropy keyspace [19]. It is a realistic scenario in which users tend to choose short, easily-rememberable passwords since they may require to remember many passwords and change the password frequently [16, 20]. Thus, passwords are vulnerable to many brute-force and dictionary-based attack tools [12]. Although a solution [26] that the server stores a one-way transform of password is introduced to strengthen the security in client/server setting, Jarecki *et al.* [15] pointed out that this solution allows for precomputation attacks that lead to the instantaneous compromise of user passwords upon server compromise. Simple password-based authentication has proven to be more and more inadequate [18] since the existing solutions cannot sufficiently prevent password-cracking, data-stealing, and data-phishing practices. Various schemes [8, 31, 34] have been proposed in succession to reduce the affection of password-cracking and compromised password database.

With the growing number of innovative ways to authenticate users, there are three main approaches [24] for authentication: something you *know* (e.g., passwords), something you *have* (e.g., smartphones and smart cards), and something you *are* (e.g., biometric characteristics). In certain circumstances, however, the above factors may be insecure. When the honest user types in the correct password, the malicious user could peep the input. The

smart card might be lost, stolen, or cloned. Once an adversary obtains the smart card, all the information stored could be lost. The biometric characteristics are irrevocable. Once copied by the adversary, this will cause permanent damage. Various multi-factor authenticated key exchange (MFAKE) schemes [7, 22, 27, 36] were proposed successively by combining three factors in an authentication process to reduce the damage caused by compromising an authentication factor.

1.1 Motivations

User authentication is becoming more widely used to protect sensitive information from the illegitimate user. However, research over the past decade has shown that designing a secure authenticated key exchange scheme is very difficult. MFAKE schemes aim to achieve higher security by combining three factors within the same authentication process. Intuitively, an adversary would have to break all three factors to break the MFAKE scheme. However, an adversary could compromise less than three factors to break the scheme if the scheme is not well designed.

An AKE protocol is provable security if and only if the security proof is correct. Several results [10, 21, 33, 35] show that even several of the proposed AKE protocols that have provided security proof cannot achieve their security aims since the security proof might be flawed. Constructing a multi-factor authentication protocol remains hard work. Analysis of defects in existing protocols can make us avoid these shortcomings when designing a new scheme.

1.2 Contributions

In this paper, we revisit Zhang's MFAKE protocol [36] and analyze its security. We hope our analysis would help avoid such mistakes when designing a new MFAKE protocol in the future. This paper is an extension work of Ma *et al.* [23]. The first contribution was presented in [23] at the CIMSS of ACNS workshop in 2022. The second contribution has some minor changes for entity authentication, which is different from [23]. The last two contributions are our new results. All contributions of this paper are listed as follows:

1) First, we show this protocol has a vital security flaw, which may lead the protocol insecure against Man-in-the-Middle (MITM) attacks and outsider Key Compromise Impersonation (KCI) attacks. The main problem of Zhang's MFAKE is the protocol message transcript is not bound to the session key. We give the details of a simple MITM attack and an outsider KCI attack in Section 5.

2) We then propose an improved MFAKE protocol to fix the problem of Zhang's MFAKE protocol with minimum changes. A hash algorithm takes protocol messages as inputs and outputs the session key. And one party only computes the session key after it authenticates another party.

3) In addition to the key indistinguishability security experiment, we also define an entity authentication security experiment. We provide the formal security proof of entity authentication and key indistinguishability in the random oracle model.

4) Finally, we evaluate security features and performance of our improved MFAKE protocol.

1.3 Organization of the Rest Article

The rest of the paper is organized as follows. In Section 2, we review the related works. In Section 3, we introduce the basic definitions for Zhang's MFAKE protocol. In Section 4, we give the security model. In Section 5, we review Zhang's MFAKE protocol, analyze the drawback of Zhang's MFAKE protocol, propose an improved MFAKE protocol and provide the formal security proof in the random oracle model. In Section 6, we show security features and performance of some related protocols and our improved MFAKE protocol. We conclude the paper in Section 7.

2 Related Work

Bellovin and Merritt [4] proposed the first password-based authenticated key exchange protocol, Encrypted Key Exchange (EKE), which allows the client and server to share the plaintext password and exchange key material to derive a common session key. Then the augmented EKE protocol proposed by Bellovin and Merritt [5], replaced the requirement that the server stores the plaintext password with a one-way transformed value of the password. Augment EKE protocol prevents the adversary from impersonating the honest user. They presented two ways to accomplish this goal, digital signatures and a family of commutative one-way functions. However, the EKE and augment EKE are not given formal security analysis since the lack of a proper security model. The first formal security model of AKE protocols between two parties was introduced by Bellare and Rogaway [3]. Bellare *et al.* [2] proposed the security model of PAKE protocols by extending the definition of Bellare and Rogaway [3]. And this PAKE security model has been followed extensively in papers [1, 23, 27].

The protocols referred to above build on the single authentication factor. Recently, MFAKE, a valuable and challenging goal, has wildly caught researchers' attention [7, 11, 22, 27]. Many papers claim security by combining all three factors in a protocol. Pointcheval and Zimmer [27] defined a new security model for MFAKE protocols and proposed a multi-factor AKE protocol that was proved to be secure in their security model. They claim their MFAKE protocol remains semantically secure

if there are at most two corrupt queries. Namely, an adversary must break all three factors to win the game. Liu *et al.* [22] proposed a three-party MFAKE protocol by extending Pointcheval's protocol [27]. They provided the formal security proof of their three-party MFAKE protocol in the random oracle model. However, Hao and Clarke [10] found out Pointcheval's protocol and Liu's protocol are insecure. If an adversary has compromised the client's password, it could impersonate the server to compromise the other two factors, thus breaking the entire system. Fleischhacker *et al.* [7] introduced and modeled a general framework for (α, β, γ)-MFAKE by extending the three-factor AKE model from [27]. And they defined a generalized notion of *tag-based* multi-factor authentication, extending the preliminary concepts from [14] that considered the use of tags (auxiliary strings) in public key-based challenge-response scenarios. In this way, they avoided the problems identified in [10] for the protocol in [27]. Wang *et al.* [29] introduced a multi-factor authentication protocol using elliptic curve cryptography. But Wu *et al.* [32] demonstrated that Wang *et al.* [29] protocol was insecure against impersonation attacks. Therefore, they proposed an improved authentication scheme and fixed the problems in [29]. Wu *et al.* [30] proposed a lightweight scheme for wireless sensor networks with multi-factor authentication. Hossein *et al.* [6] proposed a hash-chain-based provably secure MFAKE scheme and analyzed the security of their scheme in the Real-or-Random (ROR) model [1].

Most recently, Zhang *et al.* [36] proposed a multi-factor authenticated key exchange (MFAKE) scheme based on the security model from [27]. It claims to reduce the security of protocol to the Decisional Diffie-Hellman (DDH) hard problem. In this work, however, we found two weaknesses that led to Zhang's MFAKE insecurity. One problem is that an adversary could easily modify the exchanged message to lead two non-partnered sessions to compute the same session key. Another is that once an adversary compromises the server, it could impersonate the client to the server.

3 Preliminaries

Let $\lambda \in \mathbb{N}$ be the security parameter and 1^λ be a string that consists of λ bits. $\emptyset$ denotes an empty string. $\|$ is the string concatenation operation. $\bigoplus$ is the XOR operation. For $n \in \mathbb{N}$, $[n] := \{1, 2, \ldots, n\}$ denotes the set of integers between 1 and n. If X is a set, $x \xleftarrow{\$} X$ denotes the operation of sampling a uniform random element x from X. If A is a probabilistic algorithm, $a \xleftarrow{\$} A$ means that a is the output of running A with fresh random coins. The hash function $\mathsf{h}(\cdot) : \{0,1\}^* \to \{0,1\}^\lambda$ is modeled as a random oracle.

3.1 Metric Space

A *metric space* is a set $\mathcal{M}$ with a distance function $\mathsf{Dist} : \mathcal{M} \times \mathcal{M} \to [0, \infty)$. Commonly, *Hamming distance* is used to measure the distance from one value to another value. $\mathsf{Dist}(w, w')$ is the number of positions in which the strings $w \in \mathcal{M}$ and $w' \in \mathcal{M}$ differ. For an element $w \in \mathcal{M}$, let $\mathsf{Dist}(w) := \mathsf{Dist}(w, 0)$.

3.2 Min-Entropy and Statistical Distance

Definition 1 (Min-Entropy)**.** *The min-entropy of X is $H_\infty(X) = -\log_2(\max_x \Pr[X = x])$.*

Definition 2 (Statistical Distance)**.** *The statistical distance between two random variables A and B with the same domain $\mathcal{M}$ is*

$$\mathbf{SD}(A, B) = \frac{1}{2} \sum_{w \in \mathcal{M}} |\Pr[A = w] - \Pr[B = w]|.$$

If $\mathbf{SD}(A, B) \leq \epsilon$, A and B are called ϵ-statistically indistinguishable.

3.3 Public Key Encryption Scheme

Generally, we consider a public key encryption scheme PKE that consists of three probabilistic polynomial time (PPT) algorithms $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$. The PKE scheme is associated with public keyspace $\mathcal{PK}_{\mathsf{PKE}}$, private keyspace $\mathcal{SK}_{\mathsf{PKE}}$, message space $\mathcal{M}_{\mathsf{PKE}}$ and ciphertext space $\mathcal{C}_{\mathsf{PKE}}$. The algorithms of PKE are defined as follows:

- $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.KeyGen}(1^\lambda)$: This algorithm takes as input the security parameter 1^λ and outputs a pair of public/private keys $(\mathsf{pk}, \mathsf{sk})$, where the public key $\mathsf{pk} \in \mathcal{PK}_{\mathsf{PKE}}$ and the private key $\mathsf{sk} \in \mathcal{SK}_{\mathsf{PKE}}$.
- $c \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk}, m)$: This is the encryption algorithm that generates a ciphertext $c \in \mathcal{C}_{\mathsf{PKE}}$ for a message $m \in \mathcal{M}_{\mathsf{PKE}}$ with the public key pk.
- $m \xleftarrow{\$} \mathsf{PKE.Dec}(\mathsf{sk}, c)$: This is the decryption algorithm which takes as input a private key sk, a ciphertext c, and outputs a message m. The correctness requirement is for all pairs $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.KeyGen}(1^\lambda)$, we have $m \equiv \mathsf{PKE.Dec}(\mathsf{sk}, \mathsf{PKE.Enc}(\mathsf{pk}, m))$.

Definition 3 (Public Key Encryption Scheme)**.** *We say that a public key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ is $(q, t, \epsilon_{\mathsf{PKE}})$-secure (indistinguishable) against adaptive chosen-ciphertext attacks, if $|\Pr[\mathsf{EXP}^{ind\text{-}cca}_{\mathsf{PKE},\mathcal{A}}(\lambda) = 1] - 1/2| \leq \epsilon_{\mathsf{PKE}}$ holds for all adversaries $\mathcal{A}$ running in time at most t in*

the following experiment:

$\mathsf{EXP}^{ind\text{-}cca}_{\mathsf{PKE},\mathcal{A}}(\lambda)$:
$(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{PKE.KeyGen}(1^\lambda)$;
$(m_0,m_1) \xleftarrow{\$} \mathcal{A}(pk)$;
$b \xleftarrow{\$} \{0,1\}$;
$c^* \xleftarrow{\$} \mathsf{PKE.Enc}(\mathsf{pk},m_b)$;
$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{PKE.Dec}}(\mathsf{sk},\cdot)}(\mathsf{pk},c^*)$;
if $b=b'$ *then return 1, otherwise return 0* ;

$\mathcal{O}_{\mathsf{PKE.Dec}}(\mathsf{sk},c)$:
if $c=c^*$, *return a failure* $\perp$;
otherwise $m \xleftarrow{\$} \mathsf{PKE.Dec}(\mathsf{sk},c)$
return m;

where $\epsilon_{\mathsf{PKE}} = \epsilon_{\mathsf{PKE}}(\lambda)$ *is a negligible function in the security parameter* λ *and the number of queries* q *is bound by time* t.

3.4 Message Authentication Code Scheme

We consider a message authentication code scheme MAC that consists of three probabilistic polynomial time (PPT) algorithms $\mathsf{MAC} = (\mathsf{MAC.KeyGen}, \mathsf{MAC.Tag}, \mathsf{MAC.Vfy})$. The MAC scheme is associated with tag space $\mathcal{T}_{\mathsf{MAC}}$, message space $\mathcal{M}_{\mathsf{MAC}}$ and private keyspace $\mathcal{SK}_{\mathsf{MAC}}$. The algorithms of MAC are defined as follows:

- $\mathsf{sk}_{\mathsf{MAC}} \xleftarrow{\$} \mathsf{MAC.KeyGen}(1^\lambda)$: This is the key generation algorithm which takes as input 1^λ and outputs a secret key $\mathsf{sk}_{\mathsf{MAC}} \in \mathcal{SK}_{\mathsf{MAC}}$.
- $\tau \xleftarrow{\$} \mathsf{MAC.Tag}(\mathsf{sk}_{\mathsf{MAC}}, m)$: The generation algorithm is run by a party. It generates a tag $\tau \in \mathcal{T}_{\mathsf{MAC}}$ for a message $m \in \mathcal{M}_{\mathsf{MAC}}$ with the generation key $\mathsf{sk}_{\mathsf{MAC}}$.
- $\{0,1\} \xleftarrow{\$} \mathsf{MAC.Vfy}(\mathsf{sk}_{\mathsf{MAC}}, \tau, m)$: The verification algorithm is run by the verifier. It takes as input a private key $\mathsf{sk}_{\mathsf{MAC}}$, a tag τ, and a message m. Then it outputs 1 if τ is a valid tag for m under $\mathsf{sk}_{\mathsf{MAC}}$, and 0 otherwise.

Definition 4 (Message Authentication Code Scheme)**.** *We say that a message authentication code scheme* $\mathsf{MAC} = (\mathsf{MAC.KeyGen}, \mathsf{MAC.Tag}, \mathsf{MAC.Vfy})$ *is* $(q,t,\epsilon_{\mathsf{MAC}})$*-secure against strongly existential forgeries under chosen message attacks, if* $\Pr[\mathsf{EXP}^{seuf-cma}_{\mathsf{MAC},\mathcal{A}}(\lambda) = 1] \leq \epsilon_{\mathsf{MAC}}$ *holds for all adversaries* $\mathcal{A}$ *running in time at most* t *in the following experiment:*

$\mathsf{EXP}^{seuf\text{-}cma}_{\mathsf{MAC},\mathcal{A}}(\lambda)$:
$\mathsf{sk}_{\mathsf{MAC}} \xleftarrow{\$} \mathsf{MAC.KeyGen}(1^\lambda)$;
$(m^*,\tau^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{MAC.Tag}}(\mathsf{sk}_{\mathsf{MAC}},\cdot)}$;
return 1 *if the following conditions are held:*

1) $\mathsf{MAC.Vfy}(\mathsf{sk}_{\mathsf{MAC}}, \tau^*, m^*) = 1$ *and*
2) $\mathcal{A}$ *didn't submit* m^* *to* $\mathsf{MAC.Tag}(\mathsf{sk}_{\mathsf{MAC}},\cdot)$,

and 0 *otherwise*;

where $\epsilon_{\mathsf{MAC}} = \epsilon_{\mathsf{MAC}}(\lambda)$ *is a negligible function in the security parameter* λ*, on input message* m *the oracle* $\mathcal{O}_{\mathsf{MAC.Tag}}(\mathsf{sk}_{\mathsf{MAC}},\cdot)$ *returns* $\tau \xleftarrow{\$} \mathsf{MAC.Tag}(\mathsf{sk}_{\mathsf{MAC}}, m)$ *and the number of queries* q *is bound by time* t.

If $\mathsf{sk}_{\mathsf{MAC}}$ *is a one-time authentication key of* MAC *scheme, then* MAC *scheme is known as a one-time message authentication code (OTMAC) scheme which is* $(1,t,\epsilon_{\mathsf{MAC}})$*-secure.*

3.5 Fuzzy Extractor

We consider a fuzzy extractor FE that consists of a pair of probabilistic polynomial time (PPT) algorithms $\mathsf{FE} = (\mathsf{FE.Gen}, \mathsf{FE.Rep})$. The FE is associated with metric space $\mathcal{M}_{\mathsf{FE}}$, randomness space $\mathcal{RS}_{\mathsf{FE}}$, extracted string space $\mathcal{ES}_{\mathsf{FE}}$ and helper string space $\mathcal{HS}_{\mathsf{FE}}$. The algorithms of FE are defined as follows:

- $(R,P) \xleftarrow{\$} \mathsf{FE.Gen}(crs, w)$: This is the generation algorithm that takes as input $crs \in \mathcal{RS}_{\mathsf{FE}}$ and $w \in \mathcal{M}_{\mathsf{FE}}$ and outputs an extracted string $R \in \mathcal{ES}_{\mathsf{FE}}$ and a helper string $P \in \mathcal{HS}_{\mathsf{FE}}$. Note that $\mathbf{SD}((R,P),(U_\lambda,P)) \leq \epsilon_{\mathsf{FE}}$, where U_λ is uniform distribution on $\{0,1\}^\lambda$.
- $R \xleftarrow{\$} \mathsf{FE.Rep}(w',P)$: This is the reproduce algorithm that takes as input a string $w' \in \mathcal{M}_{\mathsf{FE}}$ and a helper string $P \in \mathcal{HS}_{\mathsf{FE}}$. If $\mathsf{Dist}(w,w')$ is no more than a predetermined threshold ts and $(R,P) \xleftarrow{\$} \mathsf{FE.Gen}(crs,w)$, this algorithm outputs $\mathsf{FE.Rep}(w',P) = R$. Otherwise, no guarantee is provided about the output of FE.Rep.

Definition 5 (Fuzzy Extractor)**.** *Let* $\mathcal{W}$ *be a family of distributions over metric space* $\mathcal{M}_{\mathsf{FE}}$ *with* $H_\infty(\mathcal{W}) \geq min$*, where min is min-entropy of* $\mathcal{M}_{\mathsf{FE}}$*. We say that a fuzzy extractor* $\mathsf{FE} = (\mathsf{FE.Gen}, \mathsf{FE.Rep})$ *is* $(min, ts, q, t, \epsilon_{\mathsf{FE}})$*-secure (indistinguishable), if* $|\Pr[\mathsf{EXP}^{ind}_{\mathsf{FE},\mathcal{A}}(\lambda) = 1] - 1/2| \leq \epsilon_{\mathsf{FE}}$ *holds for all adversaries* $\mathcal{A}$ *running in time at most* t *in the following experiment:*

$\mathsf{EXP}^{ind}_{\mathsf{FE},\mathcal{A}}(\lambda)$:
$crs \xleftarrow{\$} \mathcal{RS}_{\mathsf{FE}}$;
$w \xleftarrow{\$} \mathcal{W}, U_\lambda \xleftarrow{\$} \{0,1\}^\lambda$;
$b \xleftarrow{\$} \{0,1\}$;
$(R^*,P^*) \xleftarrow{\$} \mathsf{FE.Gen}(crs,w)$;
$R_0 = U_\lambda, R_1 = R^*$;
$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_{\mathsf{FE.Gen}}(\cdot,\cdot)}(crs, R_b, P^*)$;
if $b=b'$ *then return 1, and 0 otherwise*;

$\mathcal{O}_{\mathsf{FE.Gen}}(crs', w')$:
if $\mathcal{A}$ *submits* $crs' \neq crs$ *and* $0 < \mathsf{Dist}(w,w') \leq ts$,
$(R_i,P_i) \xleftarrow{\$} \mathsf{FE.Gen}(crs', w')$,
return (R_i,P_i);
else, return a failure $\perp$.

where $\epsilon_{\mathsf{FE}} = \epsilon_{\mathsf{FE}}(\lambda)$ *is a negligible function in the security parameter* λ *and the number of queries* q *is always bound by time* t.

Definition 6 (DDH Assumption)**.** *We say the DDH assumption holds, given parameters* $(\mathbb{G},p,g)$ *where* $\mathbb{G}$ *is a cyclic group of prime order* p *and* g *as a generator of* $\mathbb{G}$*, if it is hard to distinguish triples of the form* (g^x,g^y,g^{xy}) *from triples of the form* (g^x,g^y,g^z)*, where* x*,* y*, and* z *are random chosen from* $\mathbb{Z}^*_p$*. Namely, the DDH problem is* $(t,\epsilon_{\mathsf{DDH}})$*-hard, if* $|\Pr[\mathsf{EXP}^{\mathsf{DDH}}_{\mathbb{G},p,g,\mathcal{A}}(\lambda) = 1] - 1/2| \leq \epsilon_{\mathsf{DDH}}$ *holds for all adversaries* $\mathcal{A}$ *running in time at most* t *in the following experiment:*

$\mathsf{EXP}^{\mathsf{DDH}}_{\mathbb{G},p,g,\mathcal{A}}(\lambda)$:
$g \xleftarrow{\$} \mathbb{G}, (x,y,z) \xleftarrow{\$} \mathbb{Z}_p^*$;
$b \xleftarrow{\$} \{0,1\}$;
if $b=0$ *then* $X \xleftarrow{\$} g^{xy}$, *otherwise* $X \xleftarrow{\$} g^z$;
$b' \xleftarrow{\$} \mathcal{A}(\mathbb{G}, p, g, g^x, g^y, X)$;
return 1, *if* $b=b'$, *and* 0 *otherwise*;

where $\epsilon_{\mathsf{DDH}} = \epsilon_{\mathsf{DDH}}(\lambda)$ *is a negligible probability in the security parameter* λ.

4 Security Model

4.1 Execution Environment

In the execution environment, we fix a set of honest parties $\mathcal{IDS} = \{\mathsf{id}_1, \ldots, \mathsf{id}_l\}$ for $l \in \mathbb{N}$, where id_i $(i \in [l])$ is the identity of client or server. Each identity id_i is associated with a pair of long-term keys $(\mathsf{pk}_i, \mathsf{sk}_i) \in (\mathcal{PK}_{\mathsf{PKE}}, \mathcal{SK}_{\mathsf{PKE}})$. Each honest party id_i can sequentially and concurrently execute the protocol multiple times with different intended partners. We may realize a collection of oracles $\{\Pi^s_{\mathsf{id}_i} : i \in [l], s \in [d]\}$ for $(l,d) \in \mathbb{N}$ that represent the protocol executions of a set of honest parties. Each oracle $\Pi^s_{\mathsf{id}_i}$ works as the s-th protocol instance performed by party id_i. Moreover, we assume each oracle $\Pi^s_{\mathsf{id}_i}$ maintains a list of independent internal state variables with semantics listed in Table 1.

Table 1: Internal states of oracles

Variable	Description
pid^s_i	Identity of id_i's intended partner.
sid^s_i	Session identity of $\Pi^s_{\mathsf{id}_i}$, $\mathsf{sid}^s_i \xleftarrow{\$} \{0,1\}^\lambda$.
Φ^s_i	Internal state of $\Pi^s_{\mathsf{id}_i}$, $\Phi^s_i \in \{\mathsf{accept}, \mathsf{reject}\}$.
K^s_i	Session Key of $\Pi^s_{\mathsf{id}_i}$, $K^s_i \in \mathcal{K}$.
sT^s_i	Transcript of messages sent by $\Pi^s_{\mathsf{id}_i}$.
rT^s_i	Transcript of messages received by $\Pi^s_{\mathsf{id}_i}$.

All those variables of each oracle are initialized with the empty string $\emptyset$. At some point, each oracle $\Pi^s_{\mathsf{id}_i}$ may complete the execution and decide the internal state $\Phi^s_i \in \{\mathsf{accept}, \mathsf{reject}\}$. Additionally, we assume that the real session key is assigned to the variable K^s_i iff oracle $\Pi^s_{\mathsf{id}_i}$ has reached an internal state $\Phi^s_i = \mathsf{accept}$.

4.2 Adversarial Model

The adversary $\mathcal{A}$ considers being a probabilistic polynomial time (PPT) Turing Machine, having complete control of the communication network. The adversary $\mathcal{A}$ could interact with the challenger $\mathcal{C}$ by issuing the following queries:

- $\mathsf{Execute}(\mathsf{id}_i, s, \mathsf{id}_j, t)$: If the client oracle $\Pi^s_{\mathsf{id}_i}$ and server oracle $\Pi^t_{\mathsf{id}_j}$ have not been used, this query will carry out an honest execution of the protocol between two oracles, and return the transcripts sT^s_i and sT^t_j to $\mathcal{A}$. This query models the capability of $\mathcal{A}$ passively eavesdrops on plenty of honest executions.
- $\mathsf{Send}(\mathsf{id}_i, s, m)$: This query allows $\mathcal{A}$ to send a message m of his own choice to the oracle $\Pi^s_{\mathsf{id}_i}$. The oracle $\Pi^s_{\mathsf{id}_i}$ will send back the response message m' (if any) according to the protocol specification and its internal states. After answering a Send query, the variables of $\Pi^s_{\mathsf{id}_i}$ will be updated depending on the specific protocol. This query models active attacks in the real world.
- $\mathsf{Reveal}(\mathsf{id}_i, s)$: If the oracle $\Pi^s_{\mathsf{id}_i}$ has reached an internal state $\Phi^s_i = \mathsf{accept}$ (holding a session key) and a Test query has not been made to $\Pi^s_{\mathsf{id}_i}$ or its partner oracle (if it exists), it responds with the contents of the variable K^s_i. Otherwise, a failure symbol $\perp$ is returned. This query models the leakage of the session key agreed by the two parties.
- $\mathsf{Corrupt}(\mathsf{client}, a)$: This query will respond with the password pwd for $a = 0$, biometric data W for $a = 1$, and private key sk for $a = 2$. By issuing this query, $\mathcal{A}$ could obtain a-th authenticated factor $\{\mathsf{pwd}, \mathsf{W}, \mathsf{sk}\}$ of client. This query models corrupt capabilities of $\mathcal{A}$.
- $\mathsf{Corrupt}(\mathsf{server})$: This query will return server's private key to $\mathcal{A}$.
- $\mathsf{Test}(\mathsf{id}_i, s)$: $\mathcal{C}$ first flips a coin $b \in \{0,1\}$ uesd for all Test queries. If the oracle $\Pi^s_{\mathsf{id}_i}$ has state $\Phi^s_i = \mathsf{reject}$ or $K^s_i = \emptyset$, then this query returns a failure symbol $\perp$. Otherwise, $\mathcal{C}$ samples a random element K_r from session key space $\mathcal{K}$, and sets $K_0 = K_r$ and $K_1 = K^s_i$. Finally, this query responds with K_b. The oracle $\Pi^s_{\mathsf{id}_i}$ selected by the adversary in this query is called as *test oracle*. This query does not model any actual capabilities of $\mathcal{A}$. It is used to measure the semantic security of session keys.

4.3 Secure AKE Protocols

We first review the notion regarding the partnership of two oracles, i.e. *matching sessions* [2].

Definition 7 (Matching Sessions)**.** *In an MFAKE protocol, we say that the oracle* $\Pi^s_{\mathsf{id}_i}$ *and oracle* $\Pi^t_{\mathsf{id}_j}$ *are* matching sessions, *if both of them have been* accept, *hold* $(K^s_i, \mathsf{sid}^s_i, \mathsf{pid}^s_i)$ *and* $(K^t_j, \mathsf{sid}^t_j, \mathsf{pid}^t_j)$, *respectively, and all of the following conditions hold:*

1) $\mathsf{sid}^s_i = \mathsf{sid}^t_j$ *and* $K^s_i = K^t_j$.
2) $\mathsf{id}_i \in \mathsf{client}$, $\mathsf{id}_j \in \mathsf{server}$, *and vice versa.*
3) $\Pi^s_{\mathsf{id}_i}$ *has* $\mathsf{pid}^s_i = \mathsf{id}_j$ *and* $\Pi^t_{\mathsf{id}_j}$ *has* $\mathsf{pid}^t_j = \mathsf{id}_i$.
4) $sT^s_i = rT^t_j$ *and* $rT^s_i = sT^t_j$.

Correctness. We say an AKE protocol Π is correct, if an oracle $\Pi^s_{\mathsf{id}_i}$ has a *matching session* to an oracle $\Pi^t_{\mathsf{id}_j}$ and they both accept with the same session key, i.e. $K^s_i = K^t_j$.

To define the security of the session key, we need the notion of *freshness* of an oracle.

Definition 8 (Freshness). *We assume that a* client *instance* $\Pi^s_{\mathsf{id}_i}$ *has been* accept *with its intended* server id_j. *And a* server *instance* $\Pi^t_{\mathsf{id}_j}$ *(if it exists) is an oracle with intended* client id_i, *such that* $\Pi^s_{\mathsf{id}_i}$ *has a* matching session *to* $\Pi^t_{\mathsf{id}_j}$. *Then the oracle* $\Pi^s_{\mathsf{id}_i}$ *is said to be fresh if none of the following conditions holds:*

1) $\mathcal{A}$ *queried* $\mathsf{Reveal}(\mathsf{id}_i, s)$.

2) If $\Pi^t_{\mathsf{id}_j}$ *exists,* $\mathcal{A}$ *queried* $\mathsf{Reveal}(\mathsf{id}_j, t)$.

3) $\mathcal{A}$ *queried* $\mathsf{Corrupt}(\mathsf{client}, a)$ *for all three factors.*

4) If $\Pi^t_{\mathsf{id}_j}$ *exists,* $\mathcal{A}$ *queried* $\mathsf{Corrupt}(\mathsf{server})$.

4.4 Entity Authentication Security Experiment $\mathsf{EXP}^{\mathsf{Ent\text{-}Auth}}_{\Pi,\mathcal{A}}(\lambda)$

The entity authentication security experiment is processed as a game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$ based on MFAKE protocol Π, where the following steps are performed:

1) With the security parameter λ, the challenger $\mathcal{C}$ first implements the collection of oracles $\{\Pi^s_{\mathsf{id}_i} : i \in [l], s \in [d]\}$, and generates l long-term key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ for all honest parties id_i where identity $\mathsf{id}_i \in \mathcal{IDS}$ of each party is chosen uniquely.

2) $\mathcal{A}$ could issue queries to oracles Execute, Send, Reveal and Corrupt as defined above.

3) Finally, the experiment outputs 1 if and only if there exists Φ^s_i is accept and the following two conditions hold: both id_i and its intended partner id_j were not corrupted before query Test; there is no unique $\Pi^t_{\mathsf{id}_j}$, such that $\Pi^s_{\mathsf{id}_i}$ has a matching session to $\Pi^t_{\mathsf{id}_j}$.

Definition 9 (Entity Authentication). *A correct MFAKE protocol* Π *is called* $(t, \epsilon_{\mathsf{Ent\text{-}Auth}})$*-entity-authentication-secure, if for all adversaries* $\mathcal{A}$ *running within time* t *in the above MFAKE security experiment* $\mathsf{EXP}^{\mathsf{Ent\text{-}Auth}}_{\Pi,\mathcal{A}}(\lambda)$, *it holds that:*

$$\Pr[\mathsf{EXP}^{\mathsf{Ent\text{-}Auth}}_{\Pi,\mathcal{A}}(\lambda) = 1] \leq \epsilon_{\mathsf{Ent\text{-}Auth}},$$

where $\epsilon_{\mathsf{Ent\text{-}Auth}} = \epsilon_{\mathsf{Ent\text{-}Auth}}(\lambda)$ *is a negligible probability in the security parameter* λ.

4.5 Key Indistinguishability Security Experiment $\mathsf{EXP}^{\mathsf{Key\text{-}Ind}}_{\Pi,\mathcal{A}}(\lambda)$

This security experiment is also processed as a game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$ based on MFAKE protocol Π, where the following steps are performed:

1) With the security parameter λ, the challenger $\mathcal{C}$ first implements the collection of oracles $\{\Pi^s_{\mathsf{id}_i} : i \in [l], s \in [d]\}$, and generates l long-term key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$ for all honest parties id_i where identity $\mathsf{id}_i \in \mathcal{IDS}$ of each party is chosen uniquely. $\mathcal{C}$ flips a coin $b \in \{0, 1\}$ uesd for all Test queries. $\mathcal{C}$ will give all public parameters to $\mathcal{A}$ and keep track of all variables of the execution environment.

2) $\mathcal{A}$ may interact by issuing the polynomial number of queries as aforementioned, namely, $\mathcal{A}$ makes queries: Execute, Send, Reveal and Corrupt.

3) At some point of time during the game, $\mathcal{A}$ may issue a $\mathsf{Test}(\mathsf{id}_i, s)$ query.

4) $\mathcal{A}$ may continue to make the above queries. The binding constraints on this experiment are that: $\mathcal{A}$ cannot make a Reveal query on either the test session or its partnered session; $\mathcal{A}$ can make Corrupt query no more than twice if id_i is a client.

5) Finally, $\mathcal{A}$ terminates and outputs its guess b'. The experiment returns 1 if $b = b'$, and 0 otherwise.

Definition 10 (Key Indistinguishability). *A correct MFAKE protocol* Π *is called* $(t', \epsilon_{\mathsf{Key\text{-}Ind}})$*-session-key-indistinguishability, if for all adversaries* $\mathcal{A}$ *running within time* t' *in the above MFAKE security experiment* $\mathsf{EXP}^{\mathsf{Key\text{-}Ind}}_{\Pi,\mathcal{A}}(\lambda)$, *it holds that:*

$$|\Pr[\mathsf{EXP}^{\mathsf{Key\text{-}Ind}}_{\Pi,\mathcal{A}}(\lambda) = 1] - 1/2| \leq \epsilon_{\mathsf{Key\text{-}Ind}},$$

where $\epsilon_{\mathsf{Key\text{-}Ind}} = \epsilon_{\mathsf{Key\text{-}Ind}}(\lambda)$ *is a negligible probability in the security parameter* λ.

5 Security Analysis and Improvement of Zhang's MFAKE Protocol

In this section, we first review Zhang's MFAKE protocol in Figure 1. Then we analyze the drawbacks of Zhang's MFAKE protocol. Finally, an improved scheme is proposed with slight modification on the generation of the session key. The formal security proof of our scheme is provided in the random oracle model.

5.1 Zhang's MFAKE Protocol

This MFAKE scheme [36] is specified by the following algorithms in the sense of definitions in Section 3:

- Public key encryption scheme PKE = (PKE.KeyGen, PKE.Enc, PKE.Dec).
- Message authentication code scheme MAC = (MAC.Tag, MAC.Vfy).
- Fuzzy extractor FE = (FE.Gen, FE.Rep).

Initialization. Assuming that parameters are $(\mathbb{G}, p, g)$, where $\mathbb{G}$ is a cyclic group of prime order p and g is a generator of $\mathbb{G}$. Each party id_i runs PKE.KeyGen to generate

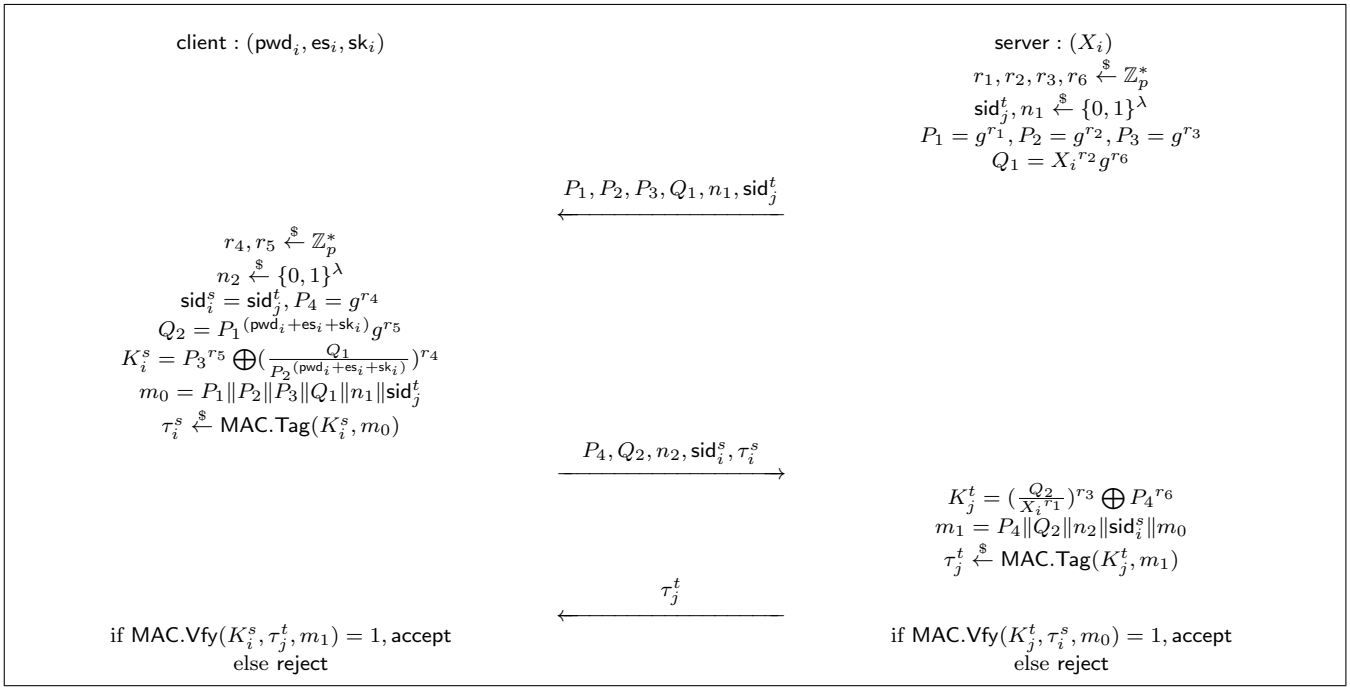


Figure 1: MFAKE protocol

key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$. We denote the public parameters are $((\mathbb{G}, p, g), \mathsf{pk}_i)$, and sk_i is a private key.

Registration. We assume the registration phase accomplishes in a secure channel. A client id_i interacts with the server id_j as following steps:

- The client randomly chooses a password pwd_i from password dictionary $\mathcal{PW}$ and creates a biometric template $\mathsf{W}_i \in \mathcal{M}_{\mathsf{FE}}$. Its private key sk_i is regarded as device data. The client sents $(\mathsf{id}_i, \mathsf{pwd}_i, \mathsf{W}_i, \mathsf{sk}_i)$ to the server.
- The server runs $\mathsf{FE.Gen}$ with W_i to obtain an extracted string $\mathsf{es}_i \in \mathcal{ES}$ and a helper string $\mathsf{hs}_i \in \mathcal{HS}$, computes $X_i = g^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)}$, runs $\mathsf{PKE.Enc}$ to obtain the ciphertext Y_i of X_i. Then it deletes the template W_i and extracted string es_i and returns hs_i to the client.
- Finally, the client stores hs_i, and the server stores identity id_i of client and Y_i.

Login-Authentication. An honest client id_i first inputs $\mathsf{pwd}_i, \mathsf{W}_i'$, runs $\mathsf{FE.Rep}$ and sends an *authentication request* to a server id_j. If there exists a Y_i corresponding to id_i, the server computes $X_i \xleftarrow{\$} \mathsf{PKE.Dec}(\mathsf{sk}_j, Y_i)$. After that, the client id_i and server id_j hold $(\mathsf{pwd}_i, \mathsf{es}_i, \mathsf{sk}_i)$ and X_i, respectively, where $X_i = g^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)}$. The MFAKE protocol performs as the following steps (as shown in Figure 1):

- The server samples four ephemeral keys r_1, r_2, r_3, r_6 from $\mathbb{Z}_p^*$, a current session identity sid_j^t and a random nonce n_1. Then it computes $P_1 = g^{r_1}$, $P_2 = g^{r_2}$, $P_3 = g^{r_3}$ and $Q_1 = {X_i}^{r_2} g^{r_6}$. The *authentication challenge* $(P_1, P_2, P_3, Q_1, n_1, \mathsf{sid}_j^t)$ sends to the client.
- After receiving the *authentication challenge*, the client samples two ephemeral keys r_4, r_5 from $\mathbb{Z}_p^*$ and a random element n_2. It sets $\mathsf{sid}_i^s = \mathsf{sid}_j^t$, computes $P_4 = g^{r_4}$, $Q_2 = {P_1}^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)} g^{r_5}$ and $K_i^s = {P_3}^{r_5} \bigoplus (\frac{Q_1}{{P_2}^{(\mathsf{pwd}_i + \mathsf{es}_i + \mathsf{sk}_i)}})^{r_4}$. The client runs $\mathsf{MAC.Tag}$ to generate a tag τ_i^s of $m_0 = P_1 \| P_2 \| P_3 \| Q_1 \| n_1 \| \mathsf{sid}_j^t$, and sends $(P_4, Q_2, n_2, \mathsf{sid}_i^s, \tau_i^s)$ as *authentication response* to server.
- After receiving the *authentication response*, the server can compute $K_j^t = (\frac{Q_2}{{X_i}^{r_1}})^{r_3} \bigoplus {P_4}^{r_6}$. It runs $\mathsf{MAC.Tag}$ to generate a tag τ_j^t of $m_1 = P_4 \| Q_2 \| n_2 \| \mathsf{sid}_i^s \| m_0$, and sends τ_j^t to the client.
- Finally, the client and server run $\mathsf{MAC.Vfy}(K_i^s, \tau_j^t, m_1)$ and $\mathsf{MAC.Vfy}(K_j^t, \tau_i^s, m_0)$, respectively. Φ_i^s or Φ_j^t sets to be accept if the output is 1, and reject otherwise.

5.2 The Insecurity of Zhang's MFAKE Scheme

Man-in-the-Middle Attack. In the following, we present a Man-in-the-Middle (MITM) attack on Zhang's MFAKE scheme. We assume that an adversary $\mathcal{A}$ intervenes in communication between the client and server. $\mathcal{A}$ could receive, forward, and modify the message exchanged between them.

The concrete MITM attack steps are performed as below:

1) $\mathcal{A}$ arbitrarily chooses client oracle $\Pi_{\mathsf{id}_i}^s$ and server oracle $\Pi_{\mathsf{id}_j}^t$ as target oracles.

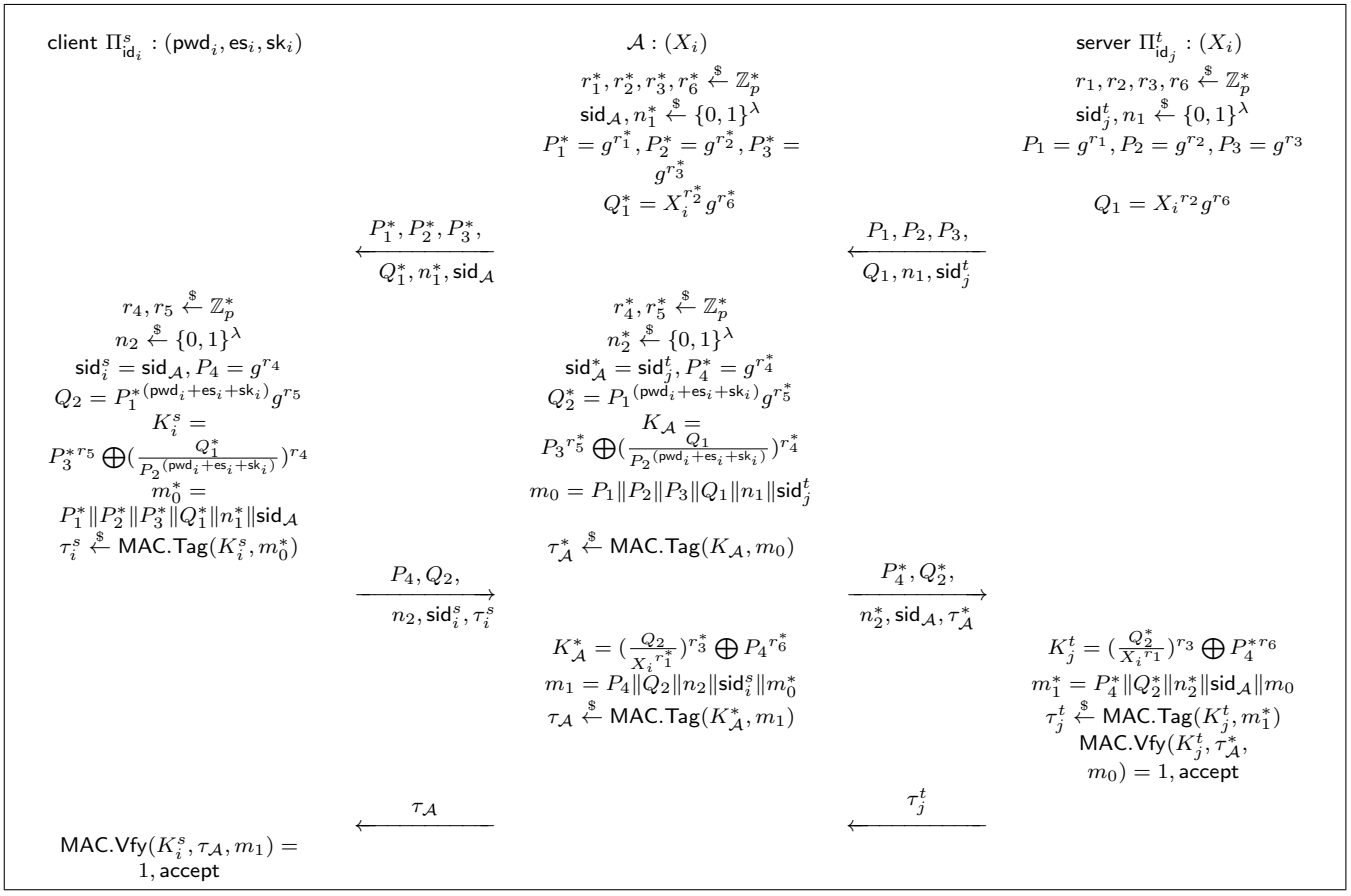


Figure 2: Outsider KCI attack

2) $\mathcal{A}$ asks $\Pi^s_{\mathsf{id}_i}$ to execute the protocol instance.

3) $\mathcal{A}$ intercepts $(P_4, Q_2, n_2, \mathsf{sid}_i^s, \tau_i^s)$ and changes n_2 to n_3, where $n_3 \in \{0,1\}^\lambda$ is randomly chosen by $\mathcal{A}$.

4) $\mathcal{A}$ does not forge the keying materials of session key. Thus $\Pi^t_{\mathsf{id}_j}$ could compute a session key $K_j^t = K_i^s$ and accept for $\mathsf{MAC.Vfy}(K_j^t, \tau_i^s, m_0) = 1$.

5) At this moment, however, $sT_i^s \neq rT_j^t$, the oracle $\Pi^s_{\mathsf{id}_i}$ doesn't have a *matching session* to an oracle $\Pi^t_{\mathsf{id}_j}$.

6) $\mathcal{A}$ could queries $\mathsf{Reveal}(\mathsf{id}_j, t)$ to get the session key K_j^t. Then $\mathcal{A}$ generates a tag $\tau_j^{t^*}$ of $m_1 = P_4 \| Q_2 \| n_2 \| \mathsf{sid}_i^s \| m_0$ to make $\Pi^s_{\mathsf{id}_i}$ be accept. $K_j^t = K_i^s$ means that $\mathcal{A}$ has the session key K_i^s of oracle $\Pi^s_{\mathsf{id}_i}$ while $\Pi^s_{\mathsf{id}_i}$ is fresh.

7) Finally, $\mathcal{A}$ can query $\mathsf{Test}(\mathsf{id}_i, s)$ and wins the game by comparing K_b with K_i^s.

Outsider KCI Attack. In the following, we show that if $\mathcal{A}$ corrupts the server id_j, it could impersonate an uncorrupted client id_i to the server id_j. $\mathcal{A}$ corrupts id_j to get X_i (this is allowed due to the modeling of KCI attacks) and behaves as if the server interacts with the client. We use the superscript * of a value to be an element chosen by $\mathcal{A}$. Then $\mathcal{A}$ could get the session key $K_{\mathcal{A}}^* = g^{r_3^* r_5} \bigoplus P_4^{r_6^*} = K_i^s$ just like the server. $\mathcal{A}$ then computes g^{r_5} since it has r_3^*, r_6^*, P_4. The keying material $P_1^{*(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}$ is easily computed from $Q_2 = P_1^{*(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)} g^{r_5}$ and it leads the protocol insecure. $\mathcal{A}$ could violate the security of the MFAKE protocol via the following steps:

1) $\mathcal{A}$ first chooses a client oracle $\Pi^s_{\mathsf{id}_i}$ and a server oracle $\Pi^t_{\mathsf{id}_j}$ and executes the MFAKE protocol instances between them.

2) $\mathcal{A}$ corrupts the oracle $\Pi^t_{\mathsf{id}_j}$ to get X_i, and intercepts $P_1, P_2, P_3, Q_1, n_1, \mathsf{sid}_j^t$.

3) Meanwhile, $\mathcal{A}$ executes protocol instance with the client id_i. If $\mathcal{A}$ replaces P_1^* with P_1, it can get $P_1^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}$. If $\mathcal{A}$ replaces P_1^* with P_2, it can get $P_2^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}$.

4) $\mathcal{A}$ computes $Q_2^* = P_1^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)} g^{r_5^*}$ and $K_{\mathcal{A}} = P_3^{r_5^*} \bigoplus (\frac{Q_1}{P_2^{(\mathsf{pwd}_i+\mathsf{es}_i+\mathsf{sk}_i)}})^{r_4^*}$. $\mathcal{A}$ generates a tag $\tau_{\mathcal{A}}^*$ of message m_0, and sends $P_4^*, Q_2^*, n_2^*, \mathsf{sid}_{\mathcal{A}}, \tau_{\mathcal{A}}^*$ to $\Pi^t_{\mathsf{id}_j}$. The oracle $\Pi^t_{\mathsf{id}_j}$ would compute $K_j^t = K_{\mathcal{A}}$ and accept the session but it does not have a *matching session* to $\Pi^s_{\mathsf{id}_i}$.

5) $\mathcal{A}$ selects the oracle $\Pi^t_{\mathsf{id}_j}$ as the test oracle which should generate the session key K_j^t. Then $\mathcal{A}$ could win the game by impersonating a client and computing the session key $K_{\mathcal{A}} = K_j^t$.

The details of this attack are shown in Figure 2. $\mathcal{A}$ succeeds in impersonating the honest client id_i to server id_j's oracle $\Pi^t_{\mathsf{id}_j}$ and id_i has no *matching session* to $\Pi^t_{\mathsf{id}_j}$.

5.3 An Improvement Solution of Zhang's MFAKE Scheme

We have shown that Zhang's MFAKE scheme is vulnerable to MITM and outsider KCI attacks since the protocol message transcript is not fully bound to the keying material. We are trying to circumvent the above attacks by modifying the key derivation function. A hash function takes as input $K_i(K_j), n_1, n_2, \mathsf{sid}^s_i(\mathsf{sid}^t_j)$ and outputs the session key $K^s_i(K^t_j)$. More specifically, our improved scheme is shown in Figure 3.

Theorem 1. *Suppose that the message authentication code scheme* MAC *is* $(1, t, \epsilon_{\mathsf{MAC}})$*-secure against strongly existential forgeries under chosen message attacks. Then our improved MFAKE scheme is* $(t, \epsilon_{\mathsf{Ent\text{-}Auth}})$*-entity-authentication-secure provided that*

$$\epsilon_{\mathsf{Ent\text{-}Auth}} \leq \frac{(9dl)^2}{2^\lambda} + dl \cdot \epsilon_{\mathsf{MAC}}.$$

Proof. We consider the proof following a sequence of games. Generally speaking, the values processed in $\Pi^{s^*}_{\mathsf{id}_i}$ are highlighted with *. Let S_ξ be the event that the adversary wins the security experiment in Game ξ, and $\mathsf{Adv}_\xi = \Pr[\mathsf{S}_\xi]$ denotes the advantage of $\mathcal{A}$ in Game ξ.

Game 0. This is the original entity authentication security game between $\mathcal{A}$ and $\mathcal{C}$. So we can write the following:

$$\mathsf{Adv}_0 = \Pr[\mathsf{S}_0].$$

Game 1. The challenger proceeds exactly like the previous game but aborts if event E_1 happens, where E_1 denotes two oracles generate the nonce, $((r^*_1, r^*_2, r^*_3, r^*_6, n^*_1, \mathsf{sid}^{t^*}_j), (r^*_4, r^*_5, n^*_2))$, which has been sampled before. The probability of the collision of those values is negligible since the nonces are chosen uniformly at random. There are l parties and at most d oracles for each party, the birthday paradox results provide that the event E_1 happens with the probability $\Pr[\mathsf{E}_1] \leq \frac{(9dl)^2}{2^\lambda}$. Thus we have that

$$\mathsf{Adv}_0 \leq \mathsf{Adv}_1 + \frac{(9dl)^2}{2^\lambda}.$$

Game 2. In this game, $\mathcal{C}$ aborts when event E_2 happens. We define the event E_2 which happens if $\Pi^s_{\mathsf{id}_i}$ receives messages with a valid tag τ^t_j which is not send by its intended partner oracle $\Pi^t_{\mathsf{id}_j}$. We have $\mathsf{Adv}_1 \leq \mathsf{Adv}_2 + \Pr[\mathsf{E}_2]$.

If the event E_2 happens with overwhelming probability, then we could construct a tag forger $\mathcal{F}_2$ against the security of the message authentication code scheme as follows. The forger $\mathcal{F}_2$ simulates the challenger for $\mathcal{A}$. It first guesses an oracle that the adversary can forge, i.e. $\Pi^{t^*}_{\mathsf{id}_j}$. Next $\mathcal{F}_2$ generates all other secret keys honestly as the challenger in the previous game. If $\mathcal{A}$ outputs a message with a valid tag not generated by $\mathcal{F}_2$, then $\mathcal{F}_2$ could use the tag to break security. Since there are at most dl oracles for all parties, the event E_2 happens with the probability $\Pr[\mathsf{E}_2] \leq dl \cdot \epsilon_{\mathsf{MAC}}$. Thus it holds that

$$\mathsf{Adv}_1 \leq \mathsf{Adv}_2 + dl \cdot \epsilon_{\mathsf{MAC}}.$$

Summing up all the probabilities from Game 0 to Game 2, we hold the result of Theorem 1.

□

Theorem 2. *Suppose that the public key scheme* PKE *is* $(d, t, \epsilon_{\mathsf{PKE}})$*-secure against adaptive chosen-ciphertext attacks, the fuzzy extractor* FE *is* $(min, ts, d, t, \epsilon_{\mathsf{FE}})$*-secure, the hash function* h *is collision-resistant and the DDH problem is* $(t, \epsilon_{\mathsf{DDH}})$*-hard. Assume that the bit-length of* pwd *is* μ_1*, the bit-length of* W *is* μ_2*, and the bit-length of* sk *is* μ_3*. Then the improved MFAKE scheme is* (t', ϵ)*-session-key-secure with* $t \approx t'$ *and*

$$\epsilon_{\mathsf{Key\text{-}Ind}} \leq \epsilon_{\mathsf{Ent\text{-}Auth}} + dl \cdot (max\{\frac{1}{2^{\mu_1}}, \epsilon_{\mathsf{FE}}, \frac{1}{2^{\mu_3}}\} + 2\epsilon_{\mathsf{DDH}}).$$

Proof. We consider the proof following a sequence of games. $\mathcal{A}$ chooses the test oracle $\Pi^{s^*}_{\mathsf{id}_i}$ executed between its owner id_i and its intended partner id_j. Generally speaking, the values processed in $\Pi^{s^*}_{\mathsf{id}_i}$ are highlighted with *. Let S_ξ be the event that the adversary wins the security experiment in Game ξ, and $\mathsf{Adv}_\xi = \Pr[\mathsf{S}_\xi] - \frac{1}{2}$ denotes the advantage of $\mathcal{A}$ in Game ξ.

Game 0. This is the original security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The bit b is chosen at the beginning of Game 0. $\mathcal{C}$ will answer the queries of $\mathcal{A}$ on behalf of the instances. By definition, it holds that

$$\Pr[\mathsf{S}_0] = \frac{1}{2} + \epsilon = \frac{1}{2} + \mathsf{Adv}_0.$$

Game 1. The challenger proceeds exactly like the previous game but aborts if event E_1 happens, where E_1 denotes an oracle accepts maliciously. From Theorem 1, we have that

$$\mathsf{Adv}_0 \leq \mathsf{Adv}_1 + \epsilon_{\mathsf{Ent\text{-}Auth}}.$$

Game 2. In this game, $\mathcal{C}$ aborts if $\mathcal{A}$ asks the Send query with client's keys $(\mathsf{pwd}^*_i, \mathsf{es}^*_i, \mathsf{sk}^*_i)$ or server's key X^*_i. We let $pes^* = \mathsf{pwd}^*_i + \mathsf{es}^*_i + \mathsf{sk}^*_i$. Due to definition of three-factors security, $\mathcal{A}$ can only compromise two factors. Since there are l parties and at most d oracles for each party, $\mathcal{A}$ can ask dl Send queries. The three possible cases might occur as follows:

1) If W^*_i and sk^*_i are leaked, $\mathcal{A}$ could try to guess low-entropy passwords using the password dictionary attacks. $\mathcal{A}$ could guess correctly in this case with probability $\frac{dl}{2^{\mu_1}}$.

2) If pwd^*_i and sk^*_i are leaked, $\mathcal{A}$ could guess the extracted string es^*_i from helper string hs^*_i with the $\mathsf{FE.Rep}(\cdot)$ function. Due to the use of the fuzzy extractor, $\mathcal{A}$ has an additional advantage ϵ_{FE}. Namely, $\mathcal{A}$ could guess correctly in this case with probability $dl \cdot \epsilon_{\mathsf{FE}}$.

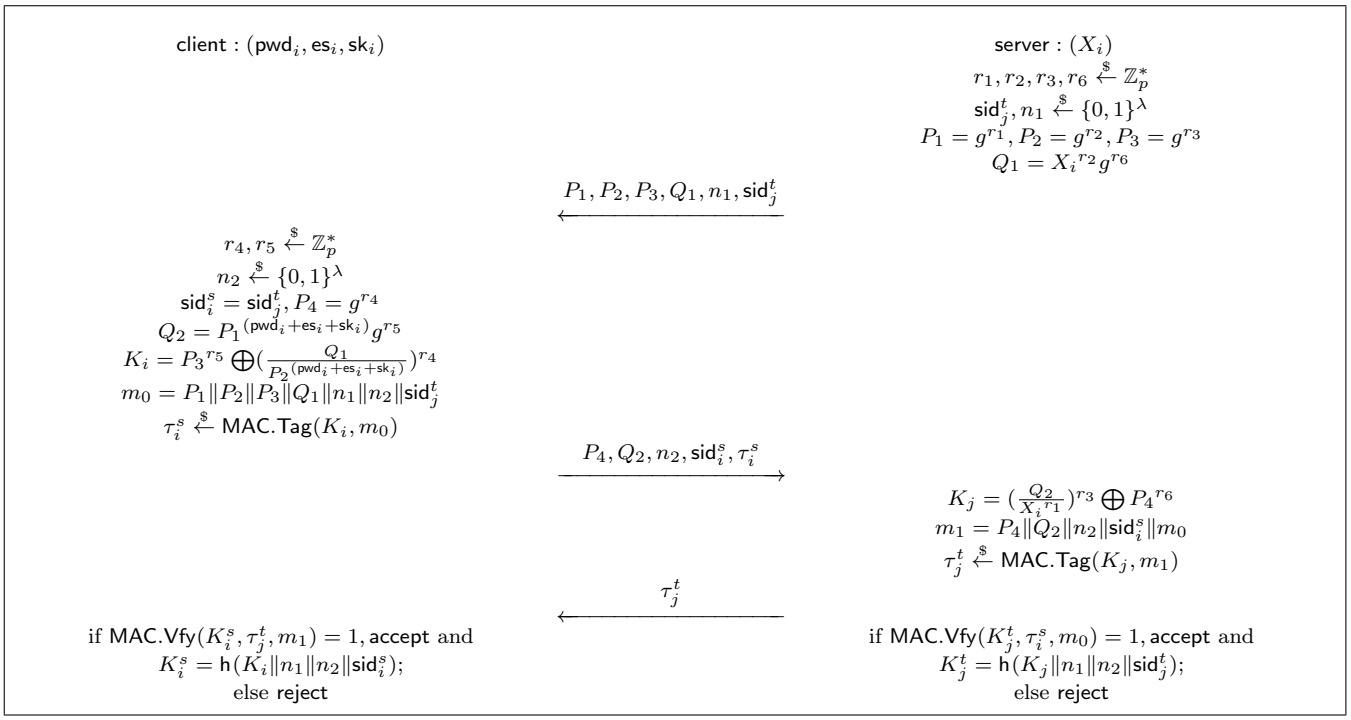


Figure 3: Improved MFAKE protocol

3) If pwd_i^* and W_i^* are leaked, $\mathcal{A}$ still has no information about sk_i^* which means pes^* is still random for $\mathcal{A}$. $\mathcal{A}$ could guess correctly in this case with probability $\frac{dl}{2^{\mu_3}}$.

Then, we have that

$$\mathsf{Adv}_1 \leq \mathsf{Adv}_2 + dl \cdot max\{\frac{1}{2^{\mu_1}}, \epsilon_{\mathsf{FE}}, \frac{1}{2^{\mu_3}}\}.$$

Game 3. In this game, $\mathcal{C}$ change the computations of Q_1^* and Q_2^* by $Q_1^* = g^{r_6^*}$ and $Q_2^* = g^{r_5^*}$. Similarly, the computations of K_i^* and K_j^* change to $K_i^* = {P_3^*}^{r_5^*} \bigoplus {Q_1^*}^{r_4^*}$ and $K_j^* = {Q_2^*}^{r_3^*} \bigoplus {P_4^*}^{r_6^*}$. We change this game that $\mathcal{C}$ will answer the Test oracle with a random key and abort if event E_3 happens. We define the event E_3 which happens if $\mathcal{A}$ asks hash oracle with valid K_i^*. If E_3 happens with non-negligible probability, we can build an algorithm $\mathcal{A}_3$ against the DDH challenge. The $\mathcal{A}_3$ receives values (g^x, g^y, g^z) such that either $z = xy$ or $z \xleftarrow{\$} \mathbb{Z}_p^*$ and runs the adversary $\mathcal{A}$ as a subroutine. If $\mathcal{A}_3$ receives a Diffie-Hellman triple, this game proceeds exactly as Game 2, otherwise it is identical to Game 3. If $\mathcal{A}$ can distinguish with non-negligible probability whether $g^z = g^{xy}$ or not, then $\mathcal{A}_3$ can use $\mathcal{A}$ to break the DDH assumption. There are at most dl oracles for all parties. Due to the security of DDH assumption, it holds that

$$\mathsf{Adv}_2 \leq \mathsf{Adv}_3 + 2dl \cdot \epsilon_{\mathsf{DDH}}.$$

In this game, the answer of each Test query is a random key that is independent of the bit b. Thus, the advantage that $\mathcal{A}$ wins is $\mathsf{Adv}_3 = 0$.

Summing up all the probabilities from Game 0 to Game 3, we hold the result of Theorem 2.

□

Table 2: Security features

	[27]	[11]	[7]	[36]	[28]	[9]	Ours
Session key security	×	×	×	×	√	√	√
Mutual authentication	√	√	√	√	√	√	√
Impersonation attack resilience	×	√	×	×	×	×	√
Man-in-the-Middle attack resilience	×	√	×	×	×	×	√
Replay attack resilience	√	√	√	√	√	√	√
Forward secrecy	√	×	√	√	√	√	√
Password guessing attack resilience	√	√	√	√	√	√	√
Biometrics template privacy	×	√	×	√	√	√	√
Stolen smartcard attack resilience	×	×	√	√	√	√	√
Known session key security	√	×	√	√	√	√	√

6 Comparison

In this section, we compare our improved protocol with some proposed MFAKE schemes in terms of security features and performance, i.e. Pointcheval-Zimmer [27], Huang *et al.* [11], Fleischhacker *et al.* [7], Zhang *et al.* [36], Wan *et al.* [28] and Guo *et al.* [9].

6.1 Security Features

The major security properties of the listed schemes are shown in Table 2. The terms √/× represent that a security property is satisfied/unsatisfied by a protocol.

In Table 2, the protocols [7, 11, 27, 36] fail to provide session key security. The protocols [7, 9, 27, 28, 36] cannot resist impersonation attack and Man-in-the-Middle attack. Huang *et al.* [11] cannot provide forward secrecy and known session key security. Biometrics template privacy leaks in Pointcheval-Zimmer [27] and Fleischhacker *et al.* [7] protocols. Pointcheval-Zimmer [27] and Huang *et al.* [11] are vulnerable to stolen smartcard attack. In contrast to these protocols, it is easy to see that our scheme can provide all of those security properties as listed in

Table 3: Performance comparison

	Computation Cost		Communication Cost (Bytes)		Storage Cost (Bytes)		Rounds
	Client	Server	Client	Server	Client	Server	
[27]	$(2N+4)T_{\mathsf{pm}} + NT_{\mathsf{h}}$	$(2N+4)T_{\mathsf{pm}} + NT_{\mathsf{h}}$	60 + 20N	60 + 60N	20 + N/8	80 + 80N	4
[11]	$T_{\mathsf{pm}} + T_{\mathsf{FE}} + T_{\mathsf{SIG}} + T_{\mathsf{PKE}} + T_{\mathsf{MAC}} + 2T_{\mathsf{h}}$	$T_{\mathsf{pm}} + T_{\mathsf{SIG}} + T_{\mathsf{PKE}} + T_{\mathsf{MAC}} + T_{\mathsf{h}}$	232	120	192	80	3
[7]	$9T_{\mathsf{pm}} + T_{\mathsf{SIG}} + 2T_{\mathsf{PKE}} + (N+7)T_{\mathsf{h}}$	$9T_{\mathsf{pm}} + T_{\mathsf{SIG}} + 2T_{\mathsf{PKE}} + (N+7)T_{\mathsf{h}}$	204 + 32N	172	40	80 + N/8	6
[36]	$6T_{\mathsf{pm}} + T_{\mathsf{FE}} + 2T_{\mathsf{MAC}}$	$8T_{\mathsf{pm}} + T_{\mathsf{PKE}} + 2T_{\mathsf{MAC}}$	140	220	40	60	3
[28]	$2T_{\mathsf{pm}} + T_{\mathsf{FE}} + T_{\mathsf{PKE}} + 9T_{\mathsf{h}}$	$2T_{\mathsf{pm}} + T_{\mathsf{FE}} + T_{\mathsf{PKE}} + 5T_{\mathsf{h}}$	100	40	120	40	3
[9]	$3T_{\mathsf{pm}} + T_{\mathsf{FE}} + T_{\mathsf{SIG}} + 7T_{\mathsf{h}}$	$3T_{\mathsf{pm}} + T_{\mathsf{SIG}} + 4T_{\mathsf{h}}$	132	112	176	120	2
Ours	$6T_{\mathsf{pm}} + T_{\mathsf{FE}} + 2T_{\mathsf{MAC}} + T_{\mathsf{h}}$	$8T_{\mathsf{pm}} + T_{\mathsf{PKE}} + 2T_{\mathsf{MAC}} + T_{\mathsf{h}}$	140	220	40	60	3

Table 2.

6.2 Performance Evaluation

In Table 3, we will compare the performance of our scheme with schemes [7, 9, 11, 27, 28, 36] in terms of computation cost, communication cost, storage cost, and rounds, respectively. We consider the computation cost and communication cost of the login and authentication phase. The experiment uses 160 bits standard elliptic curve as in [36]. To compare the computation cost, we define the time of the primary functions required in each protocol. Let T_{pm} denote the time of executing an elliptic curve point multiplication operation. Let T_{FE} denote the time of executing a fuzzy extractor/reproduce operation. Let T_{SIG} denote the time of executing a signature/verify operation. Let T_{PKE} denote the time of executing a encryption/decryption operation. Let T_{MAC} denote the time of executing a tag/verify operation. Let T_{h} denote the time of executing a one-way hash function operation.

For computing the communication and storage overhead, the length of elliptic curve group value, random nonce, and session identity are 160 bits, respectively. Message authentication code is instantiated with HMAC-SHA1, which outputs 160 bits hash value. We simulate the hash function with SHA256 hashing algorithm, which outputs 256 bits hash value. N is bit length of biometrics.

In terms of computation cost, our protocol is more efficient than Pointcheval-Zimmer [27] and Fleischhacker *et al.* [7] protocols, and almost equal to Zhang *et al.* [36] protocol. Our protocol takes more computation cost than Huang *et al.* [11], Wan *et al.* [28] and Guo *et al.* [9]. In contrast to our protocol, however, the protocol in [11] includes a multi-factor authentication without providing session key agreement and is insecure against stolen smartcard attack. Furthermore, our protocol resists impersonation attack and Man-in-the-Middle attack, which are not satisfied in Wan *et al.* [28] and Guo *et al.* [9] protocol. Although our scheme is less efficient than Guo *et al.* [9] in terms of communication cost and rounds, we can provide more security properties. Our scheme takes more communication cost than Wu *et al.* [28]. However, storage cost of our scheme is more efficient. And our scheme resists impersonation attack and Man-in-the-Middle attack, which is not provided by Wu *et al.* [28]. The storage cost of our protocol is optimal.

6.3 Summary

Our protocol satisfies all security features, while each of the others has some weaknesses. We argue that our protocol is more in accordance with the actual application requirements while ensuring security and efficiency.

7 Conclusion

In this paper, we have studied the MFAKE protocol proposed by Zhang *et al.* [36]. As described above, we prove that the security of the MFAKE protocol has some flaws. A simple Man-in-the-Middle attack and an outsider Key Compromise Impersonation attack have been shown in detail. To remedy these weaknesses, an improvement MFAKE scheme has been proposed, which is secure against the attacks mentioned above. The security of the improved protocol was verified in the random oracle model. The results of the formal security proof, security features, and performance evaluation show our improved protocol is more suitable for practical application.

Acknowledgments

Portions of this work were presented at the CIMSS of ACNS workshop in 2022. We would like to thank Zengpeng Li for insightful comments and discussions. This work was supported by the Natural Science Foundation of China (Grant No.61872051) and the Action Plan for high-quality Development of Graduate Education of Chongqing University of Technology (No.gzlcx20223226).

References

[1] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *8th International Workshop on Theory and Practice in Public Key Cryptography (PKC 2005)*, pp. 65–84, Les Diablerets, Switzerland, January 2005.

[2] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *International Conference on the Theory*

and Application of Cryptographic Techniques (EUROCRYPT 2000), pp. 139–155, Bruges, Belgium, May 2000.

[3] M. Bellare and Phillip Rogaway, "Entity authentication and key distribution," in *13th Annual International Cryptology Conference (CRYPTO 1993)*, pp. 232–249, Santa Barbara, California, USA, August 1993.

[4] S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *1992 IEEE Computer Society Symposium on Research in Security and Privacy (IEEE S&P 1992)*, pp. 72–84, Oakland, CA, USA, May 1992.

[5] S. M. Bellovin and M. Merritt, "Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise," in *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS 1993)*, pp. 244–250, Fairfax, Virginia, USA, November 1993.

[6] H. A. N. Far, M. Bayat, A. K. Das, M. Fotouhi, S. M. Pournaghi, and M. A. Doostari, "LAPTAS: lightweight anonymous privacy-preserving three-factor authentication scheme for wsn-based iiot," *Wireless Networks*, vol. 27, no. 2, pp. 1389–1412, 2021.

[7] N. Fleischhacker, M. Manulis, and A. Azodi, "A modular framework for multi-factor authentication and key exchange," in *Security Standardisation Research - First International Conference (SSR 2014)*, pp. 190–214, London, UK, December 2014.

[8] Y. Gu, S. Jarecki, and H. Krawczyk, "KHAPE: asymmetric PAKE from key-hiding key exchange," in *41st Annual International Cryptology Conference (CRYPTO 2021)*, pp. 701–730, Virtual Event, August 2021.

[9] J. Guo, S. Lu, C. Gu, X. Chen, and F. Wei, "Security analysis and design of authentication key agreement protocol in medical internet of things," in *International Conference on Networking and Network Applications (NaNA 2020)*, pp. 233–240, Haikou City, China, December 2020.

[10] F. Hao and D. Clarke, "Security analysis of a multi-factor authenticated key exchange protocol," in *Applied Cryptography and Network Security - 10th International Conference (ACNS 2012)*, pp. 1–11, Singapore, June 2012.

[11] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2014.

[12] M. S. Hwang, J. W. Lo, S. C. Lin, "An efficient user identification scheme based on ID-based cryptosystem", *Computer Standards & Interfaces*, vol. 26, no. 6, pp. 565-569, 2004.

[13] M. S. Hwang and W. P. Yang, "Conference key distribution protocols for digital mobile communication systems", *IEEE Journal on Selected Areas in Communications*, vo1. 13, no. 2, pp. 416-420, Feb. 1995.

[14] T. Jager, F. Kohlar, S. Schäge, and J. Schwenk, "Generic compilers for authenticated key exchange," in *16th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2010)*, pp. 232–249, Singapore, December 2010.

[15] S. Jarecki, H. Krawczyk, and J. Xu, "Opaque: An asymmetric pake protocol secure against pre-computation attacks," in *37th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2018)*, pp. 456–486, Tel Aviv, Israel, 2018.

[16] C. C. Lee, M. S. Hwang, L. H. Li, "A new key authentication scheme based on discrete logarithms", *Applied Mathematics and Computation*, vol. 139, no. 2, pp. 343-349, July 2003.

[17] C. C. Lee, C. H. Liu, M. S. Hwang, "Guessing attacks on strong-password authentication protocol", *International Journal of Network Security*, vol. 15, no. 1, pp. 64-67, 2013.

[18] Z. Li, Z. Yang, P. Szalachowski, and J. Zhou, "Building low-interactivity multifactor authenticated key exchange for industrial internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 844–859, 2021.

[19] I. C. Lin, C. C. Chang, M. S. Hwang, "Security enhancement for the simple authentication key agreement algorithm", in *Proceedings 24th Annual International Computer Software and Applications Conference (COMPSAC'00)*, 2000.

[20] C. H. Ling, C. C. Lee, C. C. Yang, and M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN", *International Journal of Network Security*, vol. 19, no. 2, pp. 177-181, Mar. 2017.

[21] W. Liu, X. He, and Z. Ji, "An improved authentication protocol for telecare medical information system," *International Journal of Electronics and Information Engineering*, vol. 12, no. 4, pp. 170–181, 2020.

[22] Y. Liu, F. Wei, and C. Ma, "Multi-factor authenticated key exchange protocol in the three-party setting," in *Information Security and Cryptology - 6th International Conference (Inscrypt 2010)*, pp. 255–267, Shanghai, China, October 2011.

[23] Z. Ma and J. He, "Outsider key compromise impersonation attack on a multi-factor authenticated key exchange protocol," in *Applied Cryptography and Network Security Workshops (ACNS Workshops 2022)*, pp. 320–337, Rome, Italy, June 2022.

[24] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication: A survey," *Cryptography*, vol. 2, no. 1, pp. 1–31, 2018.

[25] H. H. Ou, M. S. Hwang and J. K. Jan, "A cocktail protocol with the authentication and key agreement

on the UMTS", *Journal of Systems and Software*, vol. 83, no. 2, pp. 316-325, Feb. 2010.

[26] D. Pointcheval and G. Wang, "VTBPEKE: verifier-based two-basis password exponential key exchange," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (AsiaCCS 2017)*, pp. 301–312, Abu Dhabi, United Arab Emirates, April 2017.

[27] D. Pointcheval and S. Zimmer, "Multi-factor authenticated key exchange," in *Applied Cryptography and Network Security, 6th International Conference (ACNS 2008)*, pp. 277–180, New York, USA, June 2008.

[28] T. Wan, X. Liu, W. Liao, and N. Jiang, "Cryptanalysis and improvement of a biometric-based authentication scheme for multi-server architecture," *International Journal of Network Security*, vol. 22, no. 3, pp. 490–501, 2020.

[29] F. Wang, G. Xu, C. Wang, and J. Peng, "A provably secure biometrics-based authentication scheme for multiserver environment," *Security and Communication Networks*, vol. 2019, no. 4, pp. 1–15, 2019.

[30] F. Wu, X. Li, L. Xu, P. Vijayakumar, and N. Kumar, "A novel three-factor authentication protocol for wireless sensor networks with iot notion," *IEEE System Journal*, vol. 15, no. 1, pp. 1120–1129, 2021.

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

[32] T. Y. Wu, L. Yang, Z. Lee, C. M. Chen, J. S. Pan, and S. H. Islam, "Improved ecc-based three-factor multiserver authentication scheme," *Security and Communication Networks*, vol. 2021, no. 1, pp. 1–14, 2021.

[33] Q. Xie, Z. Tang, and K. Chen, "Cryptanalysis and improvement on anonymous three-factor authentication scheme for mobile networks," *Computers & Electrical Engineering*, vol. 59, pp. 218–230, 2017.

[34] Z. Yang, C. Jin, J. Ning, Z. Li, A. Dinh, and J. Zhou, "Group time-based one-time passwords and its application to efficient privacy-preserving proof of location," in *Annual Computer Security Applications Conference (ACSAC 2021)*, pp. 497–512, Virtual Event, USA, December 2021.

[35] Z. Yang and S. Li, "On security analysis of an after-the-fact leakage resilient key exchange protocol," *Information Processing Letters*, vol. 116, no. 1, pp. 33–40, 2016.

[36] R. Zhang, Y. Xiao, S. Sun, and H. Ma, "Efficient multi-factor authenticated key exchange scheme for mobile communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 625–634, 2019.

Biography

Zhiqiang Ma is a master student at School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China. His main research interests include cryptography and security protocol.

Jun He received the Ph.D. degree from the School of Information Science Technology, East China Normal University, Shanghai, China, in 2010. She is currently a Teacher with the School of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China. Her main research interests include information security and cryptography.

A Lightweight Authentication Protocol for Network Security Based on PUF

Zhao-Hui Xu

(Corresponding author: Zhao-Hui Xu)

College of Agricultural and Animal Husbandry Engineering and Intelligent Chemistry, Shandong Vocational Animal Science and Veterinary College, Weifang 261061, China

Email:xuzhhu80@163.com

(Received Nov. 9, 2022; Revised and Accepted July 22, 2023; First Online Aug. 25, 2023)

Abstract

This paper analyses the protocol Wang Li *et al.* designed and points out the protocol security problem; based on the protocol framework, we propose an improved lightweight authentication protocol for resisting exhaustive attacks. The proposed protocol uses ultra-lightweight byte synthesis operation and PUF function instead of Hash and PUF function to encrypt information, which can control the computation amount while ensuring security. The byte synthesis operation fully uses the Hamming weight parameter of the encrypted information and combines the different information bits to disrupt the attacker's exhaustive attack. From the aspect of security and computation, it is shown that the proposed protocol can resist impersonation attacks and exhaustive attacks, and the calculation amount can be accepted in strictly restricted low-cost tags.

Keywords: Exhaustive Attack; Light Weight; Network Security; Physical Unclonable Function (PUF); Radio Frequency Identification (RFID); Word Synthesis Operation (Syn)

1 Introduction

Radio frequency identification technology is a technology that can use radio signals of specific frequencies to complete automatic bidirectional data transmission without contact. It can identify a single target or multiple targets in motion without physical contact [2, 16]. Radio frequency identification system (RFID) is a classic application of radio frequency identification technology. The RFID system contains tags, readers and servers. In more complex application scenarios, other devices can be added based on required [10, 15].

Since the wireless channel is used in data exchange between tags and readers, the openness of the wireless channel makes it easy to be monitored, thus leading to the disclosure of users' private information [12, 14]. To ensure the security of user data, different authentication protocols have been proposed.

The rest of this paper is organized as follows: The first section is introduction, which gives the research background, existing problems and research significance. The second section introduces the current research status at home and abroad, and analyzes the shortcomings of the existing protocol. The third section analyses the security of Wang Li *et al.*'s protocol, points out that the protocol has some potential security problems. The fourth section introduces the byte synthesis operation, explains the realization principle and steps of byte synthesis operation, and introduces it with examples. The fifth section elaborates the protocol design process and gives the protocol symbol in detail. The sixth section, this paper analyses the proposed protocol and other protocols from the perspective of performance, and shows the advantages of the proposed protocol. The seventh section analyses in detail the common attack types that the proposed protocol can resist from the perspective of specific attack types. In eighth section, based on GNY logic formalization, the proposed protocol is proved from the formalization perspective. The ninth section conludes this whole paper.

2 Related Research Works

The classical hash lock protocol is proposed in literature [8], the classical hash chain protocol is proposed in literature [6], and the classical LCAP protocol is proposed in literature [7]. All these protocols are implemented based on hash functions. these protocols' security was not fully considered when they were designed, such as forward security risks.

With the development of science and technology, not only the requirements for tags have increased, but also the requirements for protocol design have increased. The protocol should not only ensure the security, but also reduce the calculation amount as much as possible. Literature [3] designed an authentication protocol based on elliptic ECC. Elliptic ECC encryption and decryption algorithm is currently recognized as a highly secure and

unbreakable algorithm. Therefore, this protocol is impeccable in security, but it can't be used in labels with low cost and limited computing power.

The appearance of physical unclonable technology brings a new direction to people. In short, the physical unclonable function is to use the unavoidable microscopic feature deviation generated by any two physical objects in production to achieve the goal of uniquely identifying a physical object. That is to say, even if a batch of products with the same manufacturing process produced by the same manufacturer and the same production line is identical, their microscopic characteristics cannot be completely consistent.

Based on the unique advantages of physical unclonable technology, many experts and scholars have presented the authentication protocol based on PUF. For example, in literature [5], a protocol based on PUF is designed, which can ensure that the tag can't be cloned, but the tag encryption information lacks random numbers, which can't guarantee the freshness of the message and can be subjected to location attack Literature [9] also proposes a protocol based on PUF, in the process of protocol design, which not only uses PUF to encrypt data, but also adds modulo arithmetic to encrypt data, so that the protocol has good security requirements, but the calculation amount far exceeds the computing power of the tag.

In literature [13], a protocol is proposed by using PUF and pseudo-random function. The main problem of this protocol is that some messages are sent in plain text, which is easy to be intercepted by attackers. Coupled with other messages, attackers can exhaust some important private information.

In literature [11], a lightweight protocol is designed by combining PUF and HASH function, but the lack of consideration in the protocol design process leads to many problems. For example, the tag identifier should always remain the same, but the protocol updates this parameter after each round of authentication. The PUF on the tag side and the PUF on the server side use the same input parameters and compare the results to verify the authenticity of each other. Obviously, the protocol design is failed. Further analysis of the protocol in literature [11] can be found in subsequent sections.

To address the problems of the above classical protocols, such as large calculation amount or defects in protocol design, a lightweight authentication protocol based on PUF is proposed. A new encryption operation is defined, that is, byte synthesis operation. According to its implementation principle, the calculation amount can reach the ultra-lightweight level. The proposed protocol uses the byte synthesis operation and combines the PUF to realize the encrypted transmission of data, which can not only ensure security, but also effectively reduce the calculation amount . In the design process of byte synthesis operations, In order to increase the attack difficulty without adding parameters, The proposed protocol skillfully uses the Hamming weight carried by the parameters involved in the operation as a new parameter. Each parameter has different Hamming weight values and has no correlation, which makes the difficulty of cracking the word synthesis operation increase instantaneously.

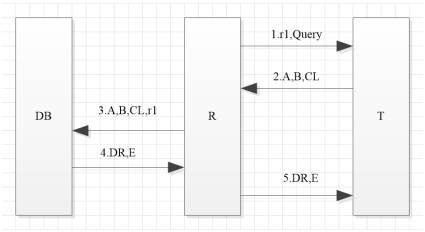


Figure 1: Wang Li *et al.* Protocol

3 Analysis of Wang Li *et al.* Protocol

According to reference [11], the protocol steps designed by Wang Li *et al.* can be summarized as follows. The schematic diagram of the protocol is shown in Figure 1.

Step 1: The reader generates a random number $r1$, and $r1$ will be sent to the tag.

Step 2: The tag generates a random number $r2$, and calculates $A = H(IDn) \oplus r1 \oplus Kn, B = PUF(r1) \oplus r2 \oplus Kn$, and $C = H(PUF(r1 \oplus r2) \oplus Kn)$. At the same time, the left half C_L of C is taken and sent to the reader together with A and B.

Step 3: The reader receives the information, plus $r1$, and sends it to the server.

Step 4: Server through C_L verify the authenticity of the tag, for true, then generate a random number $r3$, while calculating $D = H(PUF(r1 \oplus r2) \oplus Kn), E = PUF(r1 \oplus r2) \oplus Kn \oplus r3$, take the right half D_R of D, together with E sent to the reader, and finally update IDn, Kn.

Step 5: The reader sends D_R and E to the tag.

Step 6: Server through D_R verify the authenticity of the server, for true, IDn, Kn are updated.

According to the analysis of the above protocol steps, it can be found that the protocol has at least the following problems or security defects:

First: According to Step 4, after successful protocol authentication, the server and tag will update IDn. IDn is the tag identifier, and this parameter should remain unchanged. However, in this protocol, it is changed after each round of authentication. The correctness of this practice remains to be discussed.

Second: According to Step 1, it can be seen that the random numbers transmitted by the protocol belong

to plaintext transmission, and third-party can easily eavesdrop and obtain them to provide help for subsequent cracking of other information. The rationality of this method about transmitting information remains to be discussed.

Third: According to Step 2, after receiving the message from the sender, the tag doesn't identify the authenticity of the message source, which may lead to impersonation attacks.

Fourth: The protocol uses PUF to encrypt information. According to the characteristics of PUF, the output values of PUF possessed by any two different entities should be mutually distinct for the same input parameter. However, in Step 4 of this protocol, the server takes out its PUF to calculate $PUF(r1)$, and compares it with the result of tag operation, so as to verify the authenticity of the tag. Obviously, this operation is impossible to achieve, but it passed the verification in this case, which is contrary to reality, and its correctness remains to be discussed.

Fifth: The protocol fails to provide basic security, that is, the protocol can't resist exhaustive attacks. Specific analysis process:

Let $PUF(r1 \oplus r2) \oplus Kn = P$, bring p into the message C, and get $C = H(P)$, where the message C has been obtained by eavesdropping, and the specific implementation method of the hash function $H()$ is open to the public. At this time, only P is unknown to the attacker. And the attacker has only one parameter that is not known. The attacker can launch an exhaustive attack, use the exhaustive method to exhaust each value of P one by one. Finally, the correct value of P can be analyzed.

After the attacker obtains the correct value of P, the attacker can combine message E to crack the random number $r3$ as follows: $r3 = (PUF(r1 \oplus r2) \oplus Kn) \oplus E = P \oplus E$. After cracking $r3$, the attacker can launch a impersonation attack to further obtain more private information.

This protocol has many problems to be discussed and can't resist exhaustive attacks. Based on the framework of this protocol, this paper designs an improved authentication protocol that can resist exhaustive attacks.

4 Byte Synthesis Operation

In this paper, $Syn()$ is used to represent the byte synthesis operation, and the operation is defined as follows.

X, Y is the two parameters to be encrypted, and their Hamming weights are respectively $h(X), h(Y)$. The following implementation steps will be decided based on the value between $h(X)$ and $h(Y)$, which can be divided into the following two cases for discussion.

In the first case, when $h(x) \geqslant h(y)$, the left h(Y) bit of Y is placed in the back, while the right $l - h(Y)$ bit of X is placed in the front, thus forming a new parameter information. For example: $X = 01101101, Y = 00100101$, then $h(X) = 5, h(Y) = 3, l = 8$, satisfies the condition $h(X) \geqslant h(Y)$, so $Syn(X, Y) = 01101001$.

In the second case, when $h(X) < h(Y)$, the left $h(X)$ bit of X is placed in the back, while the right $l - h(X)$ bit of Y is placed in the front, thus forming a new parameter information. For example: $X = 10000100, Y = 11011011$, then $h(X) = 2, h(Y) = 6, l = 8$, satisfies the condition $h(X) < h(Y)$, so $Syn(X, Y) = 01101110$.

5 Protocol Design

This section will give the detailed design and implementation steps of the improved protocol. Different symbols will appear in the protocol design, meaning as follows:

DB, R is a server and reader composed of the whole;

T is tag;

IDi is the tag identifier;

$PUF()$ is physical unclonable function;

$Syn()$ is word synthesis operation;

$\oplus$ is XOR operation;

Ki is the current shared key between DB, R and T;

$Ki - 1$ is the shared key of the previous round between DB, R and T;

x is the random number generated by DB, R;

y is the random number generated by T;

& is and operation.

In the protocol of Wang Li *et al.*, both PUF and HASH function are used to encrypt information, which not only increases the amount of calculation, but also increases the number of gate circuits at tag end , and enlarges the cost of the tag. In view of the above shortcomings, we abandon the HASH function encryption mechanism and adopt the newly designed byte synthesis operation combined with PUF to realize information encryption.

Before the protocol starts, there is an initialization process, which completes the initial allocation of DB, R and tag information. After the initialization process is complete, the DB, R end stores the IDi of each tag and the corresponding $PUF(IDi)$.

A schematic of the protocol can be seen in Figure 2.

Here will be describes the protocol implementation of each step.

The first step: DB, R generates a random number x, calculates A, B in turn, and finally sends A, B, Ask together to the tag.

$$A = x \oplus IDi, B = Syn(x, IDi).$$

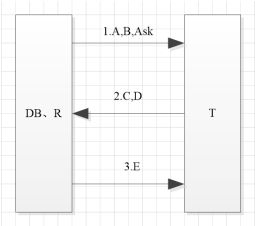


Figure 2: Authentication Protocol Based on PUF

The second step: The tag deforms A to obtain $x = A \oplus IDi$, and combines its own IDi to calculate $B1$ according to the same rule. Determine the relationship between B and $B1$.

If the relationship is not equal, DB, R can't pass the authentication and the protocol stops.

If the relationship is equal, perform subsequent operations. The tag will generate a random number y, then start calculating C, D, and finally send C, D to DB, R.

$$C = (y \oplus IDi)\&x, D = Syn(PUF(IDi)\&y, Ki\&x).$$

The third step: DB, R will search the information group $< IDi, PUF(IDi), Ki >$ stored by itself successively to verify whether D is true or false. If this step can't be verified successfully, DB, R will search for information group $< IDi, PUF(IDi), Ki-1 >$ again to verify the authenticity. If no match is found, the tag is forged by a third party and the protocol stops.

After DB, R successfully verifies the tag with information group $< IDi, PUF(IDi), Ki >, DB, R$ calculates $E = Syn(PUF(IDi)\&y, x)$, then updates the key $Ki - 1 = Ki$ and $Ki = Syn(PUF(IDi)\&x, y\&Ki)$, and finally sends E to the tag.

After DB, R successfully verifies the tag with information group $< IDi, PUF(IDi), Ki-1 >, DB, R$ calculates $E = Syn(PUF(IDi)\&y, x)$, then updates the key $Ki - 1 = Ki - 1$ and $Ki = Syn(PUF(IDi)\&x, y\&Ki - 1)$, and finally sends E to the tag.

The forth step: The tag receives the message and uses E to verify that DB, R is true or false. If true, the protocol stops. Otherwise, the tag starts updating the key: $Ki = Syn(PUF(IDi)\&x, y\&Ki)$.

6 Performance Analysis

In different session entities of RFID systems, The most performance-related entity is the lag l tags. Since readers or servers have enough storage space and powerful computing power, tags are not so generous in terms of computing power and storage space due to the particularity of their construction and the limited cost. The proposed protocol and other protocols are analyzed in different aspects, and the results are shown in Table 1.

Table 1: Performance Comparison Between the Proposed Protocol and other Protocols

Reference	Calculations	Traffic	Storage
Ref [5]	$8P + 7X$	$7l + 1$	$2l$
Ref [9]	$4P + 6M$ $+5X$	$10l + 1$	$4l$
Ref [13]	$2P + 7L$ $4X$	$8l + 1$	$3l$
Ref [11]	$2P + 2H$ $+9X$	$12l + 1$	$3l$
This Protocol	$1P + 4S$ $+3X$	$5l + 1$	$2l$

In Table 1, the meanings of these symbols are explained as follows: P represents the operand of the physical unclonable function; M represents the operand of modular arithmetic; L represents the operand of pseudo-random function; H represents the operand of hash function; S represents the operand of word synthesis operation; X represents the operand of digitwise operation (digitwise operation here mainly refers to nonequivalence operation, and operation); l indicates the length of the session message.

Analysis from the perspective of storage capacity: the proposed protocol is similar to other protocols in terms of storage parameters, among which the proposed protocol mainly stores parameters IDi and Ki, so the storage capacity of one end of the label is $2l$.

Analysis from the perspective of communication traffic: the main reason why the communication traffic of literature [9] and literature [11] is much larger is that the server and reader are separated, they are not regarded as a whole, which leads to a large amount of communication traffic. The proposed protocol and other protocols treat the the server and reader as a whole, so the communication traffic will be relatively much reduced.

Analysis from the perspective of calculation amount: the overall calculation amount at the tag end of the proposed protocol is $1P + 4S + 3X$, the origin of the calculation amount is analyzed in detail as follows.

When calculating $B1$, use S for the first time; When calculating D, use S for the second time; When calculating E, use S for the third time; When calculating Ki, use S for the forth time, so $4S$.

The proposed protocol only uses P when performing physical unclonable function operations on IDi, so $1P$.

When deforming A, use X for the first time; When calculating B, use X for the second and third time respectively, so $3X$.

In conclusion, the total calculation amount of at the tag end in the proposed protocol is $1P + 4S + 3X$, Com-

pared with literature [5], the proposed protocol has no advantage in terms of the calculation amount , but the proposed protocol can make up for the security risks of the protocol in literature [5]. Compared with other literatures, the proposed protocol reduces the calculation amount on the tag side, meanwhile, it can make up for the shortcomings of other protocols, such as the inability to resist asynchronous attacks and exhaustive attacks.

7 Security Analysis

This section will focus on whether the proposed protocol can resist common attacks.

Impersonation Attack

From the previous section, we know that the protocol of Wang Li *et al.* can't provide security requirements resist impersonation attacks, while the proposed protocol can.

According to the description of the proposed protocol, the attacker may impersonate tag or impersonate DB, R.

Firstly, select impersonate tag communication. The attacker needs to analyze the received messages from A, B. Only after analyzing the private information, the attacker can impersonate tags and calculate subsequent messages to pass the authentication of DB, R. However, the protocol attacker in this paper can't succeed, because the attacker lacks IDi, which makes the attacker unable to analyze the random number generated by DB, R. Without this random number, the subsequent calculation of C, D cannot be correct, and DB, R can't be verified, and the impersonation fails.

Secondly, select impersonate for DB, R communication. An attacker can randomly select a random number to participate in the calculation of message A, B. There is nothing wrong with this operation, but the attacker doesn't know the value of IDi, so the attacker can only choose a random number to impersonate it. Obviously the attacker's operations can't be verified by the tag.

Exhaustive Attack

According to the analysis of the protocol proposed by Wang Li *et al.*, the main problem of their protocol is that attackers can crack private information by exhaustive attack, so the proposed protocol must be able to resist exhaustive attack.

Here, select message A, B as an example for detailed analysis. When the attacker obtains the A, B message by some means, first process A to get $x = A \oplus IDi$, and then bring the processing result into B to get $B = Syn(A \oplus IDi, IDi)$. In this formula, the attacker seems to have only one value of IDi unknown, so the attacker attempts to exhaust all possible values of IDi by exhaustive method, but still can't succeed. The main factor is that the byte synthesis operation cleverly uses the Hamming weight value of the encryption parameter itself in the implementation process, that is, the attacker doesn't know the Hamming weight value of IDi and the Hamming weight value of $A \oplus IDi$ except the IDi value. Under the premise that the above three values are unknown, it is impossible for an attacker to exhaust private information in an exhaustive way, so the proposed protocol can provide strong security requirements.

Two-way Authentication

According to the analysis of the protocol proposed by Wang Li *et al.*, at the beginning step of the protocol, the tag doesn't verify the message source , which leads to subsequent security problems.

The proposed protocol uses the mechanism of authentication first and then subsequent operations in each step to ensure the security of privacy information. In Step 2, the tag verifies the authenticity of the source through A, B. In Step 3, DB, R verifies the authenticity of the source through C, D. In Step 4, the tag verifies the authenticity of the sender through E. Through the first authentication method described in the above steps, private information security can be guaranteed, and communication entities can verify each other to achieve two-way authentication.

Asynchronous Attack

In Step 3 of the proposed protocol, DB, R will verify the tag with $< IDi, PUF(IDi), Ki >$ information group first. When all information group $< IDi, PUF(IDi), Ki >$ fails to verify, DB, R will not immediately identify the sender as a forgery, but continue to use information group $< IDi, PUF(IDi), Ki - 1 >$ to verify the tag. And only the current two authentications fail, DB, R will determine that the sender is false. The reason why the DB, R side performs two rounds of authentication is to resist the attacker's asynchronous attack. With two rounds of authentication, the consistency of information between session entities can be guaranteed to some extent.

Replay Attack

In order to ensure that the attacker cannot pass the authentication of any session entity when sending messages, it is necessary to require that there is a difference in the message value of each round . The protocol adds a random number to the message encryption to ensure the difference of message value in each round. Some messages are mixed with a random number parameter, and some messages are mixed with two random number parameters, which disturbs the attacker 's attack idea.

Location Attack

When attackers continuously eavesdrop on messages

sent by tags and analyze these messages, it is likely to accurately locate the location of the tag. In order to make the above attack impossible, it is necessary to ensure that the message values sent by the tag are different each time, and there is no simple correlation, so that the attacker can't locate the location of the tag. The protocol is as follows: All messages sent by the tag are encrypted ciphertext, even if the attacker can obtain, the attacker can't simply crack. At the same time, the random numbers used in each message encryption process are different, and the random numbers are mutually different and unpredictable. In this way, the message value learned by the attacker is different each time, giving the attacker the impression that the tag is in a constantly moving state.

Backward-secure

Attackers use continuous eavesdropping to obtain a large number of communication messages, and perform various analyses on these messages to obtain some useful private information, prepare for subsequent attacks or reversely derive useful information. But in the proposed protocol, attackers can't succeed. In order to make the attacker unable to reversely analyze the previous useful information from the current message value, it is necessary to ensure that there is no correlation between the current message value and the previous message value. The proposed protocol solves this problem by adding random numbers in the message encryption process. Random numbers are randomly generated, uncorrelated, unpredictable, and impossible to reverse the derivation, so that there is no simple or regular relationship between the message values, which leads to the failure of backward derivation by attackers.

Through the above aspects, the security of the proposed protocol can be compared with the security of other protocols, and the results are summarized as shown in Table 2.

Table 2: Security comparison results of protocols

Attack Type	Ref [5]	Ref [9]	Ref [13]	Ref [11]	Our Protocol
Impersonation Attack	√	√	√	×	√
Exhaustive Attack	√	√	×	×	√
Two-way Authentication	√	√	√	×	√
Asynchronous Attack	√	×	√	√	√
Replay Attack	√	√	√	√	√
Location Attack	×	√	√	√	√
Backward-secure	√	√	√	√	√

8 Formal Analysis of the Proposed Protocol

This section will further analyze and reason the proposed protocols from the perspective of logic formalization. There are many options, such as BAN logic [1] and GNY logic [4]. After comprehensive comparison, GNY logic is selected to analyze the protocols in the text.

1) Protocol Formal Description

$msg1 : DB, R \rightarrow T : A, B, Ask$

$msg2 : T \rightarrow DB, R : C, D$

$msg3 : DB, R \rightarrow T : E$

The above protocol is specified in GNY formal logic language and can be described as follows:

$msg1 : T < *\{A, B\}$

$msg2 : DB, R < *\{C, D\}$

$msg3 : T < *\{E\}$

2) The Protocol Initializes Assumptions

$sup1 : (IDi, Ki) \in T$

$sup2 : (IDi, Ki) \in DB, R$

$sup3 : T| \equiv \#(x, y)$

$sup4 : DB, R| \equiv \#(x, y)$

$\sup 5 : T| \equiv DB, R \overset{Ki}{\longleftrightarrow} T$

$\sup 6 : T| \equiv DB, R \overset{IDi}{\longleftrightarrow} T$

$\sup 7 : DB, R| \equiv T \overset{Ki}{\longleftrightarrow} DB, R$

$\sup 8 : DB, R| \equiv T \overset{IDi}{\longleftrightarrow} DB, R$

3) Protocol Proof Objective

$goal1 : T| \equiv DB, R| \sim \#\{A, B\}$

$goal2 : DB, R| \equiv T| \sim \#\{C, D\}$

$goal3 : T| \equiv DB, R| \sim \#\{E\}$

4) Protocol Proof Process

Although there are three proving objectives, the proving process is similar, so only objective 1 is selected for proving.

$\because msg1 : DB, R \rightarrow T : A, B, Ask$ and rule $P1 : \frac{P<X}{X \in P}$

$\therefore \{A, B\} \in T$

$\because \sup 4 : DB, R| \equiv \#(x, y)$ and rule $F1 : \frac{P|\equiv(X)}{P|\equiv(x,y), P|\equiv \#F(X)}$

$\therefore T = \#\{A, B\}$

$\because$ Rules $P2: \frac{X\in P, Y\in P}{(X,Y)\in P, F(X,Y)\in P}, \sup 1: (IDi, Ki) \in T, \sup 2: (IDi, Ki) \in DB, R$

$\therefore \{A, B\}\# \in T$

$\because$ Rule $F10: \frac{P|\equiv(X), X\in P}{P|\equiv\#(H(X))}$ and derived $T = \#\{A, B\}$ and $\{A, B\}\# \in T$

$\therefore T| = \#\{A, B\}$

$\because$ Rule $I3: \frac{P<H(X,<S>)>, (X,S)\in P, P|\equiv P\leftrightarrow Q, P|\equiv\#(X,S)}{P|\equiv Q|\sim(X,S), P|\equiv Q\sim H(X,<S>)}$

Again $\because \sup 5: T| \equiv DB, R \overset{Ki}{\longleftrightarrow} T, \sup 6: T| \equiv DB, R \overset{IDi}{\longleftrightarrow} T, \sup 7: DB, R| \equiv T \overset{Ki}{\longleftrightarrow} DB, R, \sup 8: DB, R| \equiv T \overset{IDi}{\longleftrightarrow} DB, R$ and $msg1: DB, R \rightarrow T: A, B, Ask$

$\therefore T| = \{A, B\}$

$\because$ The definition of freshness and the derived $T = \#\{A, B\}$ and $T| = DB, R \sim \{A, B\}$

$\therefore goal1: T| \equiv DB, R| \sim \#\{A, B\}$ to be proved

And that's it.

9 Conclusion

This paper proposed an improved lightweight authentication protocol to resist exhaustive attacks. The proposed protocol uses an ultra-lightweight byte synthesis operation and a lightweight PUF to encrypt private data, which reduces the overall calculation amount at entity side. The detailed implementation process of the byte synthesis operation is given, The protocol cleverly combines the Hamming weight variable of the encryption parameter itself, which can reduce the storage burden and increase the attack difficulty without introducing new parameters. From multiple attack analysis, calculation amount at tag side analysis, logical formalization analysis, it shows that the proposed protocol can not only provide strong security requirements, but also satisfy the low-cost tag calculating.

References

[1] M. Burrowa, M. Abadi, R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[2] Y. C. Chen, W. L. Wang, M. S. Hwang, "RFID authentication protocol for anti-counterfeiting and privacy protection", in *The 9th International Conference on Advanced Communication Technology*, pp. 255-259, 2007.

[3] S. Y. Chiou, W. T. Ko, E. H. Lu, "A secure ECC-based mobile RFID mutual authentication protocol and its application," *International Journal of Network Security*, vol. 20, no. 2, pp. 396–402, 2018.

[4] L. Gong, R. Needham and R. Yahalom, "Reasoning about belief in cryptographic protocols," in *IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 234–248, 1990.

[5] M. S. Hwang, E. F. Cahyadi, S. F. Chiou, C. Y. Yang, "Reviews and analyses the privacy-protection system for multi-server," *Journal of Physics: Conference Series (JPCS)*, vol. 1237, no. 1, pp. 022–091, 2019.

[6] S. Maheshwari, "Detection of amplitude shift keying signals using current mode scheme," *International Journal of Electronics and Information Engineering*, vol. 11, no. 2, pp. 73–80, 2019.

[7] M. M. Nabi, F. Nabi, "Cybersecurity mechanism and user authentication security methods," *International Journal of Electronics and Information Engineering*, vol. 14, no. 1, pp. 1–9, 2022.

[8] H. Rezk, H. El-Bakry, M. El-Mikkawy, "Adaptive intelligent decision support system for enhancing higher education quality assurance," *International Journal of Electronics and Information Engineering*, vol. 13, no. 4, pp. 180–195, 2021.

[9] Z. Sun, M. Wei, "PUF-based anonymous RFID system ownership transfer protocol," in *IEEE. 38th Chinese Control Conference, CCC 2019, July 27-30, 2019*, pp. 6367–6373, Guangzhou, China. New York: IEEE, 2019.

[10] J. Q. Wang, Y. F. Zhang, D. W. Liu, "Provable secure for the ultra-lightweight RFID tag ownership transfer protocol in the context of IoT commerce," *International Journal of Network Security*, vol. 22, no. 1, pp. 12–23, 2020.

[11] L. Wang, F. Li, Y. Ji, *et al.*, "PUF-based anti-physical cloning RFID security authentication protocol," *Netinfo Security*, vol. 20, no. 8, pp. 89–97, 2020.

[12] Y. Wei, J. Chen, "Tripartite authentication protocol RFID/NFC based on ECC," *International Journal of Network Security*, vol. 22, no. 4, pp. 664–671, 2020.

[13] H. Xia, W. Yang, "Security access solution of cloud services for trusted mobile terminals based on trustzone," *International Journal of Network Security*, vol. 22, no. 2, pp. 201–211, 2020.

[14] R. Xie, J. Ling, D. W. Liu, "A wireless key generation algorithm for RFID system based on bit operation," *International Journal of Network Security*, vol. 20, no. 5, pp. 938–950, 2018.

[15] F. Zhu, P. Li, H. Xu, *et al.*, "A lightweight RFID mutual authentication protocol with PUF," *Sensors*, vol. 19, no. 13, pp. 2957–2978, 2019.

[16] S. H. Zhan, C. Q. Yu, "Mobile RFID authentication protocol based on permutation cross synthesis for anti counterfeit attack," *International Journal of Network Security*, vol. 24, no. 2, pp. 305–313, 2022.

Biography

Zhao-hui Xu received a master's degree in School of Computers from Guangdong University of Technology (China) in June 2011. Her current research interest fields include information security.

Database Storage Security Analysis of Enterprise Financial Management System

Yajie Gong, Xiaodan Xing, and Chang Zhang
(Corresponding author: Xiaodan Xing)

Shijiazhuang Posts and Telecommunications Technical College, Shijiazhuang, Hebei, China
Email: xingtun8@yeah.net

(Received July 10, 2022; Revised and Accepted June 13, 2023; First Online Aug. 25, 2023)

Abstract

The financial management system can improve the efficiency of enterprise financial management, and the secure storage of data in the database of the system is the most important. This paper briefly introduced the enterprise financial management system combined with blockchain technology and the secure storage process of financial data in the system and then simulated the financial management system in the laboratory server. The results showed that the enterprise financial management system could normally store the data; as financial data increased, the average throughput increased and then remained stable, and the average latency first remained stable and then increased; the financial management system could effectively maintain the security of financial data in the face of third-party brute force cracking, local database tampering, and financial summary information tampering.

Keywords: Blockchain; Database; Financial Management

1 Introduction

With the popularity and deepening of enterprise informatization, financial management system has become an indispensable and important part of modern enterprises. In the financial management system, the database stores all kinds of financial data of the enterprise, such as accounts, income and expense details, reports, and so on. Enterprises should be fully aware of the importance of database storage security [21], strengthen security awareness training, increase investment and efforts to improve system operation and maintenance capabilities to ensure the security and sound development of enterprise financial data. Therefore, database storage security has become one of the most important issues in the financial management system [4, 10, 11, 25]. It can be said that database storage security is directly related to the security protection of enterprise information by the financial management system.

The traditional protection methods include data encryption, access rights management, regular backup and security audit. The above methods can protect the financial data in the database to a reasonable extent [1], but with the increase of the business mode of modern enterprises, cascading access rights and security audit mode will reduce the timeliness, and once the data in the database is tampered with or lost, it is not only impossible to trace the source, but also difficult to recover [19, 20, 23]. The emergence of blockchain provides a new way for the secure storage of the database of financial management system [2, 3, 5, 6, 18, 24]. With the decentralized and distributed data storage feature of blockchain, the database of financial management system can realize the synchronization and traceability of stored data, and use the data stored in other nodes for data recovery.

Shen *et al.* [26] proposed a new paradigm called data integrity audit without private key storage to ensure the integrity of data stored in the cloud, and verified its effectiveness. Zaabar *et al.* [29] proposed a new architecture to avoid the problem of centralized storage of electronic medical records using decentralized databases. The performance evaluation results and comparative analysis demonstrated the robustness and superiority of the blockchain-based healthcare system in terms of key functions and performance indicators, including various throughputs and latencies.

Tian [27] constructed an agricultural product supply chain using radio frequency identification (RFID) and blockchain technologies to ensure the authenticity of the data therein. This paper briefly introduced an enterprise financial management system incorporating blockchain technology and the process of securely storing financial data in the system, and conducted a simulation experiment of the financial management system on a laboratory server.

2 Database Security Storage of Enterprise Financial Management System Combined with Blockchain

2.1 Enterprise Financial Management System

Financial management is an important part of business operations, and an enterprise financial management system can help enterprises better manage finances, improve compliance and business efficiency. Enterprise financial management system integrates accounting, finance, budgeting and cost management operations, thus realizing the information management of revenue and expense management, accounting management, financial decision making and budget management, and improving the comprehensive monitoring and management efficiency of enterprises. In addition, the enterprise financial management system can also use visual reports and analysis tools to visualize the economic situation and trend direction of the enterprise, and the rich analysis tools can help the enterprise to carry out dynamic risk analysis and early warning to timely and effectively detect and solve financial problems [22].

For the enterprise management system, the security of data stored in the database, such as various data reports reflecting the economic status of the enterprise and information of the internal personnel of the enterprise, is of paramount importance [9]. The traditional means of ensuring database storage security include data encryption and access rights management, but in the traditional centralized database, once the key or permission is leaked, the data in the database will be damaged, and it is neither traceable nor recoverable [12, 13]. Moreover, as it also takes some time to verify data encryption and access rights, it is difficult to synchronize the transmission of financial information from different departments in the enterprise, and the resulting information gap may be exploited [14].

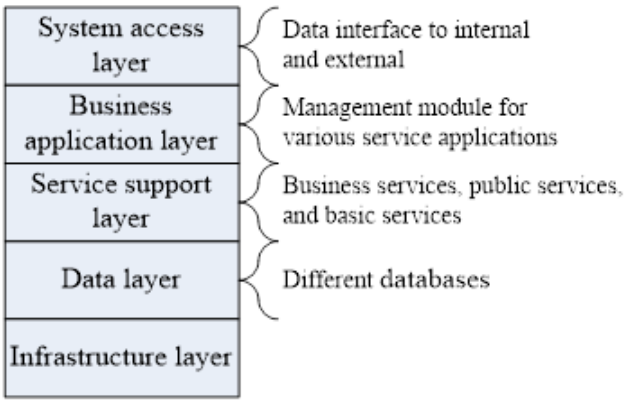


Figure 1: Basic architecture

Blockchain technology can realize the decentralized distributed storage of data, and the use of technologies such as timestamps, hash algorithms, and digital signatures can realize the traceability and authenticity guarantee of stored data, while its distributed storage method can easily recover data under the premise of ensuring the authenticity and synchronization of data [28]. As shown in Figure 1, the infrastructure layer includes the basic facilities that support the operation of the management system, such as servers for processing and storing data on the physical layer and the platform for providing blockchain services.

The data layer includes databases that store various types of data, such as various business report data. The service support layer provides various service functions, including business services, public services, and basic services. Business services are various specific financial project processing services, public services are information transfer services such as workflow and announcement logs, and basic services include various management services, especially smart contract management for blockchain storage. The business application layer is a management module after the modular integration of various service applications in the service support layer. The system access layer provides internal and external data interfaces, and the interfaces have forms of web pages and application programming interfaces (APIs) [17].

2.2 Blockchain-based Secure Storage of Databases

Figure 2 shows the flow of blockchain-based database storage in an enterprise financial management system.

1) Users of the financial management system will have different levels of permissions depending on their position in the enterprise. Users will have their permissions verified before uploading financial management data (various financial statements, project reports, etc.) and will only be able to proceed to the next step if they pass the verification. If the verification fails, the users will be prompted to be verify again or stop uploading data.

2) After users pass the authorization check, they upload the financial management data to their department's local database or to a trusted third-party database [7].

3) The key corresponding to the user authority is used to encrypt the uploaded financial data and generate the corresponding summary information of the financial data, which is the hash value of the encrypted financial data. The series of operations generate the corresponding operation log data according to the timestamp technology.

4) Smart contracts are used to verify the summary information and of financial data and the operation

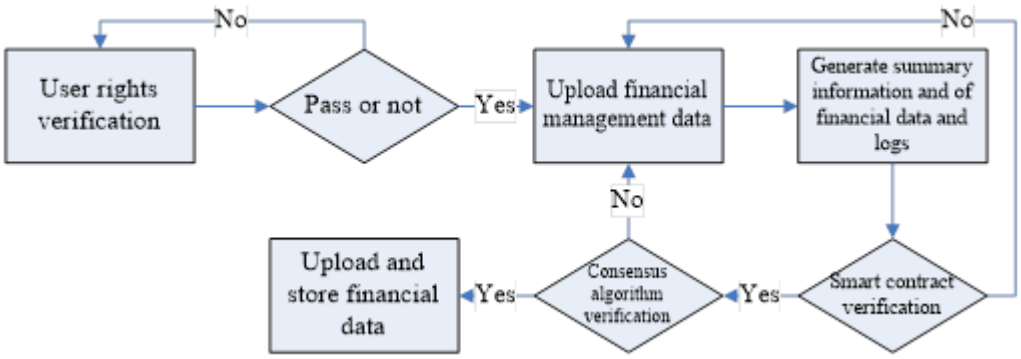


Figure 2: The storage process of blockchain-based financial management database

logs [16]. A smart contract is a program code that can be automatically executed and verified, which simply means that the text of the contract rules negotiated by the contract participants is converted into a contract code that can be recognized by a computer. After the smart contract code is generated, it is sent to the blockchain network in the form of P2P to reach consensus. After consensus is reached, the smart contract is written to the blockchain for subsequent automated verification of the financial data, reducing the waste caused by manual verification. If the smart contract passes verification, it proceeds to the next step; if not, the financial data is uploaded again.

5) After passing the smart contract's validation, the financial data is broadcast to all nodes in the blockchain network, and each node checks the financial data using a consensus algorithm; if most nodes pass the validation, i.e., reach consensus, the financial data is stored in a newly generated block. The consensus algorithm used in this paper is the practical byzantine fault tolerance (PBFT) algorithm [15], which first selects a master node in the blockchain network for consensus and the other nodes as slaves. The master node selection formula is:

$$\begin{cases} p = v \mod |n| \\ \theta_1 = \alpha \cdot \theta \\ \theta_{GC} = \begin{cases} \frac{\theta}{n} - c & \text{node cooperation} \\ \frac{\theta_1}{n} & \text{node non-cooperation} \end{cases} \end{cases} \quad (1)$$

where p is the master node number, v is the view number, $|n|$ is the total number of nodes in the view, θ is the total service fee paid by the user to the blockchain network, α is the coefficient used for revenue lure, θ_1 is the service fee used for revenue lure, n is the number of nodes in the blockchain, c is the cost of cooperation per node, and θ_{GC} is the incentive value that each node can get after successful consensus. After the master node is selected, the financial management system sends a request to the master node to upload the financial data, and the master node uses a smart contract to verify the request and then broadcasts it to the nodes in the blockchain. The slave nodes also verify the request through the smart contract and feed back the verification result to the other nodes in the blockchain. When more than two-thirds of the nodes pass the verification, the consensus is successful and each node deposits the uploaded financial data into the local blockchain ledger to complete the secure storage of the financial data; otherwise, it returns to the financial management system to have it reinitiate the request [8].

3 Simulation Experiments

3.1 Experimental Environment

The simulation experiments were conducted in a server in the lab. The blockchain network required for the enterprise financial management system was provided by a virtual machine in Ethereum. The relevant parameters of the server in the lab were quad-core i7 CPU, 16 G memory, and 1024 G hard disk. The parameters of the virtual machines provided by Ethereum were uniformly set to single-core i5 CPU, 2.5 GHz operating frequency, and 4 G memory for the sake of simulation. The number of nodes provided by the virtual machine was 10.

3.2 Experimental Projects

1) Storage function test:
Firstly, the financial data storage function of the financial management system was tested. The steps to store financial data were:

   a. Log in to the account corresponding to the position;

   b. Upload financial data in the financial data upload interface according to the prompts;

Table 1: Test results of the data storage function of the financial management system

Step number	Correct operation result	Wrong operation result
Step 1	After entering the correct account password, successfully login and jump to the financial data upload page.	If the account password is incorrect, the page will not jump, but will return the message "account or password error".
Step 2	Enter the financial data in the input box on the data upload page, click the "upload" button, and receive the "upload successful" message.	The "upload" button cannot be clicked without entering the financial data in the input box.
Step 3	After uploading the financial data, click the "upload" button and receive the "upload successful" prompt.	The "upload" button cannot be clicked when the financial data upload fails.
Step 4	In the query interface, view the successfully uploaded data according to the number of the financial data.	Financial data that has not been successfully uploaded cannot be queried in the query interface.

c. Click the "upload" button;

d. Check the status of the financial data upload in the list.

In addition to examining the financial data upload step, the average throughput and average latency of the data storage function of the enterprise financial management system were tested under different financial data sizes.

2) Database storage security testing for enterprise financial management systems:
The database storage security of the financial management system was tested from two aspects. On the one hand, a third-party server was used to play the role of an unknown user to brute-force the encrypted financial data summary information in the database, and the size of the financial data corresponding to the summary information to be brute-force cracked was 5 MB, 10 MB, 15 MB, 20 MB, and 25 MB. The brute-force cracking time was set to 30 minutes.

On the other hand, the financial data in the local database was modified to simulate a successful attack on the server and data manipulation. Then, the financial data was queried in the query interface of the company's financial management system. There were two types of data manipulation, one is to directly manipulate the plaintext of the financial data, and the other is to modify the encrypted summary of the financial data by 1, 2, 3, 4, and 5 bits, respectively.

3.3 Experimental Results

Table 1 shows the results of testing the data storage function of the financial management system.

Step 1 is to test the login function of the management system, and there will be different feedback depending on whether the account password is correct or not.

Step 2 is to test the local data upload function of the management system, and the "upload" button can be used only when there is financial data to upload.

Step 3 is to test the data upload function of the management system. Whether the "upload" button can be used depends on whether the electronic data is successfully uploaded in the previous step.

Step 4 is to test the query function of the management system, and only when the financial data is successfully uploaded can the corresponding data be queried.

Figure 3 shows the average throughput and average latency of the financial management system when storing financial data of different sizes. From Figure 3, it was seen that the average throughput of the financial management system increased as the financial data to be stored increased, but the average throughput remained the same after the data exceeded a certain size; the average latency first remained stable after the data exceeded a certain size and then increased as the average latency increased. The reason is as follows. The average throughput of the management system increased as the financial data increased, but the processing efficiency of the system for checking and uploading the data was limited, so the average throughput remained the same after the data exceeded a certain size, and due to the limited processing efficiency, the storage tasks piled up after the data exceeded a certain size, causing the average latency to increase.

Figure 4 shows the extent to which the third-party server acts as an unknown user to brute-force crack the summary information of financial data of different sizes in the management system database. It was seen from

Table 2: Storage security test results in the face of local database data tampering

	Original financial data	Financial data after local database tampering	The financial data obtained by the system
No.	9041616552001010101	9041616552001010101	9041616552001010101
Name	low-gluten flour	low-gluten flour	low-gluten flour
Batch	20191215003	20191215003	20191215003
ID	ja2raf84gv8ag0qrf4qf90	ja2raf84gv8ag0qrf4qf90	ja2raf84gv8ag0qrf4qf90
Sales date	2019/12/17	2018/12/31	2019/12/17
Point-of-sale	Chengdu,Sichuan	Chengdu,Sichuan	Chengdu,Sichuan
Salesman	Li XX	Li XX	Li XX

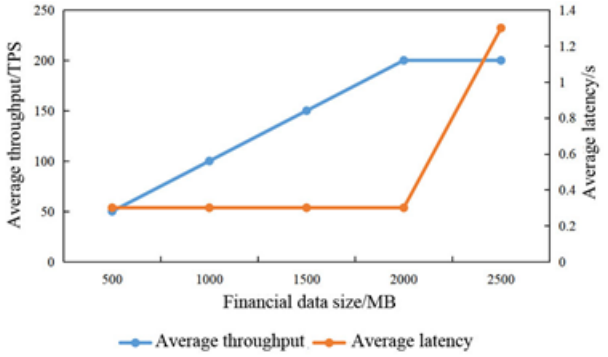


Figure 3: Average throughput and average latency of the financial management system under different financial data sizes

Figure 4 that as the financial data increased, the data integrity obtained by brute-force cracking of the data in the same time period decreased.

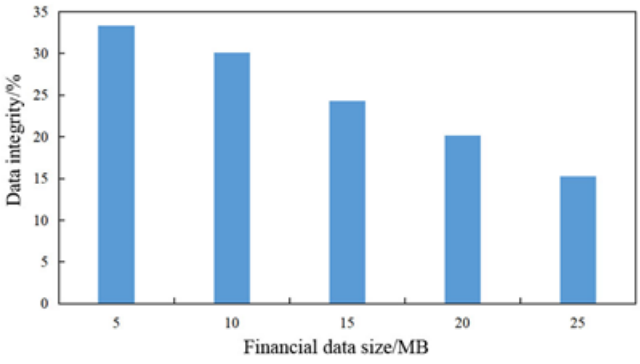


Figure 4: Level of brute-force cracking of financial data summary information by the third-party server

Table 2 shows the storage security test results of the financial management system in the face of the local database tampering. It was seen from Table 2 that after the local database was tampered with, the date in the "sales date" column was changed from "2019/12/17" to "2018/12/31". However, when the financial data report was queried through the financial management system, it not only displayed an alert that the information had been tampered with, but also restored the data in the local database.

Table 3 shows the storage security test results of the financial management system in the face of financial summary information tampering. It was noted from Table 3 that even if only 1 bit of data of the financial summary information is tampered with, it will lead to storage failure due to inconsistency with the summary information stored in other blocks, and the original financial information will remain unchanged.

4 Conclusion

This paper briefly introduced the enterprise financial management system combined with blockchain technology and the secure storage process of financial data in the system, and then simulated the financial management system in the laboratory server. The obtained results are as follows. The test results of the financial data storage function of the financial management system showed that the system could store the financial data properly. As the stored financial data increased, the average throughput of the financial management system increased and then remained stable, while the average latency remained stable and then increased. When the database of the financial management system was subjected to brute force cracking, the integrity of the data obtained by brute force cracking decreased with the increase of the financial data cracked. When the local database was tampered with, the financial management system could effectively detect if the data was tampered with and use the backup in the blockchain to recover the tampered data; when the financial summary information was tampered with, even a 1-bit data change would cause the data deposit to fail.

References

[1] A. Abdalrahman, "A cloud database based on AES 256 GCM encryption through devolving web appli-

Table 3: Results of storage security tests in the face of financial summary information tampering

Extent of modification of summary financial data information	1 bit	2 bits	3 bits	4 bits	5 bits
Store results	Failed to store falsified information, and original financial information remains unchanged	Failed to store falsified information, and original financial information remains unchanged	Failed to store falsified information, and original financial information remains unchanged	Failed to store falsified information, and original financial information remains unchanged	Failed to store falsified information, and original financial information remains unchanged

cation of accounting information system," *International Journal of Recent Technology and Engineering*, vol. 9, no. 5, pp. 268-274, 2021.

[2] P. Y. Chang, M.S. Hwang, C.C. Yang, "A blockchain-based traceable certification system", in *Security with Intelligent Computing and Big-data Services*, pp. 363-369, 2018.

[3] L. Chen, Q. Fu, Y. Mu, L. Zeng, F. Rezaeibagha, M. S. Hwang, "Blockchain-based random auditor committee for integrity verification", *Future Generation Computer Systems*, vol. 131, pp. 183-193, 2022.

[4] M. Y. Chen, C. W. Liu, M. S. Hwang, "Securedropbox: A file encryption system suitable for cloud storage services," in *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference*, pp. 1-2, 2013.

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

[6] Y. H. Chen, L. C. Huang, I. C. Lin, M. S. Hwang, "Research on blockchain technologies in bidding systems", *International Journal of Network Security*, vol. 22, no. 6, pp. 897-904, 2020.

[7] P. Dangayach, "Pharmaceutical supply chain tracking system based on blockchain technology and radio frequency identification tags," *International Journal of Business Research*, vol. 19, no. 4, pp. 37-44, 2019.

[8] H. G. Driver, T. Hartley, E. M. Price, *et al.*, "Genomics4RD: An integrated platform to share Canadian deep-phenotype and multiomic data for international rare disease gene discovery," *Human Mutation*, vol. 43, no. 6, pp. 800-811, 2022.

[9] M. S. Hwang, Chii-Hwa Lee, "Secure access schemes in mobile database systems", *European Transactions on Telecommunications*, vol. 12, no. 4, pp. 303-310, 2001.

[10] M. S. Hwang, C. C. Lee, T. H. Sun, "Data error locations reported by public auditing in cloud storage service," *Automated Software Engineering*, vol. 21, no. 3, pp. 373–390, Sep. 2014.

[11] 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.

[12] M. S. Hwang and W. P. Yang, "A two-phase encryption scheme for enhancing database security", *Journal of Systems and Software*, vo1.31, no.12, pp. 257-265, 1995.

[13] M. S. Hwang, W. P. Yang, "Multilevel secure database encryption with subkeys", *Data & Knowledge Engineering*, vol. 22, no. 2, pp. 117-131, 1997.

[14] A. A. Iyer, R. Sundar, K. S. Umadevi, "Consistency of privacy views on hippocratic and multilevel databases using smart contracts," *International Journal of Advanced Research*, vol. 9, no. 1, pp. 823-834, 2021.

[15] S. S. Kamble, A. Gunasekaran, M. Goswami, J. Manda, "A systematic perspective on the applications of big data analytics in healthcare management," *International Journal of Healthcare Management*, vol. 12, no. 3, pp. 226-240, 2019.

[16] L. Koh, A. Dolgui, J. Sarkis, "Blockchain in transport and logistics - paradigms and transitions," *International Journal of Production Research*, vol. 58, no. 7, pp. 2054-2062, 2020.

[17] M. Li, S. Shao, Q. Ye, G. Xu, G. Huang, "Blockchain-enabled logistics finance execution platform for capital-constrained E-commerce retail," *Robotics and Computer Integrated Manufacturing: An International Journal of Manufacturing and Product and Process Development*, vol. 65, pp. 1-14, 2020.

[18] C. Y. Lin, L. C. Huang, Y. H. Chen, M. S. Hwang, "Research on security and performance of blockchain with innovation architecture technology", *International Journal of Network Security*, vol. 23, no. 1, pp. 1-8, 2021.

[19] C. W. Liu, W. F. Hsien, C. C. Yang, 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.

[20] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing", *International Journal of Network Security*, vol. 18, no. 4, pp. 650-666, 2016.

[21] M. Mustafa, M. Alshare, D. Bhargava, R. Bhargava, B. Singh, P. Ngulube, "Perceived security risk based on moderating factors for blockchain technology applications in cloud storage to achieve secure healthcare systems," *Computational and Mathematical Methods in Medicine*, vol. 2022, pp. 1-10, 2022.

[22] M. Pajany, G. Zayaraz, "A robust lightweight data security model for cloud data access and storage," *International Journal of Information Technology and Web Engineering*, vol. 16, no. 3, pp. 39-53, 2021.

[23] X. Peng, J. Hu, "Big data security storage based on hybrid large-scale database," *Journal of Physics: Conference Series*, vol. 1852, no. 2, pp. 1-6, 2021.

[24] Z. P. Peng, M. L. Chiang, I. C. Lin, C. C. Yang, M. S. Hwang, "CBP2P: Cooperative electronic bank payment systems based on blockchain technology", *Journal of Internet Technology*, vol. 23, no. 4, pp. 683-692, 2022.

[25] J. S. Rauthan, K. S. Vaisla, "VRS-DB:preserve confidentiality of users5 data using encryption approach," *Digital Users: Digital Communication*, vol. 007, no. 001, pp. 62-71, 2021.

[26] W. Shen, J. Qin, J. Yu, R. Hao, J. Hu, J. Hu, "Data integrity auditing without private key storage for secure cloud storage," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1408-1421, 2021.

[27] F. Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *13th International Conference on Service Systems and Service Management (ICSSSM'16)*, pp. 1-6, 2016.

[28] Y. Yang, Y. Chen, F. Chen, "A compressive integrity auditing protocol for secure cloud storage," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1197-1209, 2021.

[29] B. Zaabar, O. Cheikhrouhou, F. Jamil, M. Ammi, M. Abid, "HealthBlock: A secure blockchain-based healthcare data management system," *Computer Networks*, vol. 200, pp.1-16, 2021.

Biography

Yajie Gong currently is an associate professor of Shijiazhuang Posts and Telecommunications Technical College. She graduated from Hebei GEO University with a master's degree in Management. Her research interests include financial management and financial analysis. She has published more than 10 academic papers and participated in more than 15 scientific research projects.

Xiaodan Xing is a lecturer at Shijiazhuang Posts and Telecommunications Technical College, and graduated from Jinan University with a master's degree in Auditing. Her research fields include enterprise financial management and auditing. She has published more than 10 academic papers and participated in more than 10 scientific research projects.

Chang Zhang is an associate professor of Shijiazhuang Posts and Telecommunications Technical College. He graduated from Beijing University of Posts and Telecommunications with a doctorate in engineering. His research interests include data mining and management science. In recent years, he has published more than 5 academic papers which have been retrieved by SCI and EI.

Effects of Shadow Fading on Energy Detection and Matched Filter Detection in Cognitive Radio Network

Hong Du[1], Dacheng Wang[2], and Long Chen[1]
(Corresponding author: Dacheng Wang)
School of Electrical and Engineering, Chongqing University of Technology[1]
Institute of Science and Technology, Chongqing University of Technology[2]
No. 69, Hongguang Avenue, Banan District, Chongqing, China. 400054
Email: wdc@cqut.edu.cn
(Received Nov. 20, 2022; Revised and Accepted June 30, 2023; First Online July 10, 2023)

Abstract

Spectrum sensing can improve the spectrum's utilization using licensed spectrum in cognitive radio. However, various security issues, such as malicious user attacks and shadow fading, affect network performance. This study focuses on how energy and matched filter detection in cognitive radio are affected by shadow fading. The formulas for the probability of detection and false alarm are derived for energy and matched filter detection, respectively. According to the simulation, matched filter detection is more susceptible to shadow fading than energy detection in an environment with a lower signal-to-noise ratio.

Keywords: Cognitive Radio; Energy Detection; Matched Filter Detection; Shadow Fading; Spectrum Sensing

1 Introduction

The Federal Communications Commission (FCC) is considering opening a portion of the authorized spectrum to unauthorized users without interfering with authorized primary users (PUs) because the allocation of fixed spectrum is no longer sufficient to meet the requirements of an increasing number of users due to the rapid development of wireless communication. With the development of cognitive radio (CR), it is now possible to use the authorized free frequency band without interfering with the primary user. Therefore, the utilization of spectrum resources is improved, and the spectrum requirements of a more significant number of wireless users are met, which is an important technology to address the problem of a lack of resources in wireless spectrum resources [23]. Cognitive radio networks are wireless communication networks with cognitive characteristics. The network can observe the surrounding wireless network environment, use environmental cognition to get information about how spectrum is used, process and learn the information, make intelligent decisions and analyses, dynamically access the available spectrum, and finally adapt and reconfigure itself to adapt to the cognitive radio network environment, which is constantly changing, to achieve optimal network performance.

A cognitive radio user is a user in a cognitive radio network, and the primary user is the cognitive radio user's counterpart. The CR user accesses to communicate when the PU is not using the channel. Once the signal of the PU returns, the CR user immediately withdraws from the channel it is communicating on and looks for other available free channels to communicate. Therefore, the CR user should first have the spectrum sensing function to detect the signal from the wireless environment, then determine the spectrum hole after analysis and adjustment, and use the spectrum hole to communicate without affecting the PU. Spectrum sensing technology, a prerequisite for operating cognitive radio networks, means that CR users collect spectrum usage information in wireless networks via various signal detection and processing methods to find spectrum holes.

Cognitive radio technology aims to solve the spectrum scarcity issue by implementing dynamic spectrum management. However, various security issues and vulnerabilities experienced can influence network performance [22]. Authors in [6] have investigated robust spectrum sensing schemes against malicious user attacks. In cognitive networks, many denial-of-service attacks will cause significant performance degradation and thus need to be detected quickly [12,14,18]. An algorithm to reduce the detection delay is presented in [21] so that a network manager can respond to an event as quickly as possible to minimize the impact of attacks.

Additionally, the structure of the cognitive radio network introduced a spectrum sensing data falsification (SSDF) attack. In such attacks, malicious users make incorrect observations of the system's fusion center, which

may cause licensed users to experience the severe quality of service degradation and disruption. The authors of [13] investigate the threat and the mitigation strategy for SSDF attacks. The Byzantine attack is one of the key issues preventing the success of cognitive radio sensor networks. The Byzantines can be avoided using the security measures suggested in [2]. A reliable sensing method has been developed to prevent the Byzantine attack. Additionally, CR users naturally face two significant security threats: jamming and primary user emulation (PUE) attacks. Machine learning has been applied to detect these attacks in [15]. The proposed deep learning-assisted detection method performs exceptionally well when spotting these threats.

Shadowing also limits the effectiveness of spectrum-sensing techniques in cognitive radio in [3, 20]. S. Kavaiya in [10] examined how an improved energy detector performs over uniformly and exponentially correlated Nakagami-m fading with imperfect channel state information (CSI). Simulation results show that user mobility and correlated fading combined affect the detection performance over imperfect CSI. In addition, H. Rasheed in [16] quantifies energy detection for spectrum sensing under shadowed conditions. A study in [5] examines performance analysis of cooperative spectrum sensing over shadowed fading. H. Huang in [9] discuss the unified performance of energy detection of spectrum sensing over generalized fading channels in cognitive radios. Fading channels will undoubtedly impact the detection performance of spectrum sensing. The results demonstrate that fading channels will impact energy detection performance, but that sensing performance can be enhanced using the appropriate channel parameters. Aulakh in [1] shows the solutions to shadow fading using two strong techniques: optimal spectrum sensing and greedy spectrum sensing. The authors of [8, 19] also investigated the sensing performance for cooperative spectrum sensing in fading channels.

The rest of the paper is organized as follows: Sections 2 and 3 looked at the spectrum sensing system model for energy and matched filter detection. Additionally, the effect of shadow fading on detection performance is discussed. Section 4 presented the simulation results and discussed the influence on the detection performance. Finally, Section 5 contains the conclusions.

2 Energy-Detection Based Spectrum Sensing Technology

Energy detection is the most commonly used method in spectrum sensing due to its simplicity. An energy detector can determine whether signals are present in a particular frequency band by detecting received signals. The steps involved in detecting energy are shown in Figure 1 as follows. After passing through an ideal Band Pass Filter (BPF), the received signal calculates the energy of signals in the band in the detecting time T, which is then compared to the threshold to determine if communications use the frequency band.

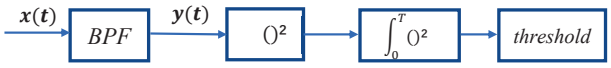


Figure 1: Block diagram of energy detection

Two presumptions can be made for spectrum sensing. H_0 denotes no primary user signal in a certain spectrum band as in Equation (1). H_1 means that a primary user signal exists in that band as in Equation (2).

$$H_0 : y(t) = n(t) \tag{1}$$

$$H_1 : y(t) = h(t)s(t) + n(t) \tag{2}$$

y(t) is the received signal by the CR user at time t. $n(t)$ is the Additive White Gaussian Noise (AWGN). We assume that $n(t)$ has a variance of 1 and an expectation of 0 under a standard normal distribution. $s(t)$ is the primary user's transmitting signal. h is the channel coefficient.

In energy detection, the sampling time is T, the signal bandwidth is W, and the number of sampling points is $N = 2TW$. It is assumed that the decision threshold is K, the signal-to-noise ratio is $\lambda_0 = \frac{E_s}{N_0}$. The detection probability, false alarm probability, and missed detection probability are shown as in Equation (3), in Equation (4) and in Equation (5):

$$P_{fa} = Q(\frac{K-N}{\sqrt{2N}}) = \frac{1}{2}erfc(\frac{K-N}{\sqrt{4N}}) \tag{3}$$

$$P_d = Q(\frac{K-N-N\lambda_0}{\sqrt{2N+N\lambda_0}}) = \frac{1}{2}erfc(\frac{K-N-N\lambda_0}{\sqrt{4N+2N\lambda_0}}) \tag{4}$$

$$P_{md} = 1 - P_d \tag{5}$$

Shadow fading will impact the primary user's signal during transmission. The normal log component of the shadow loss and the m power of the wave propagation distance r is typically used to calculate shadow fading. Here, we consider the impact of shadow loss. ζ is the log loss (in dB) caused by shadow, which follows a lognormal distribution with zero mean and variance of 1dB.

The probability of false alarm and the probability of detection can be expressed by as in Equation (6) and in Equation (7).

$$P_{fa} = \frac{1}{2}erfc(\frac{K-N}{\sqrt{4N}}) \tag{6}$$

$$P_d = \frac{1}{2}erfc(\frac{K-N-N\lambda_2}{\sqrt{4N+2N\lambda_2}}) \tag{7}$$

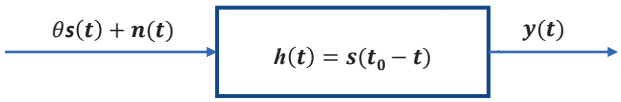


Figure 2: Block diagram of matched filter detection

3 Matched Filter Based Spectrum Sensing Technology

Matching filter detection necessitates prior knowledge of the primary user signal's modulation method, pulse waveform, timing, and packet format, among other things [4, 7, 11, 17]. We presume that BPSK is being used as a modulator in this case. The block diagram is displayed in Figure 2. The noise signal $n(t)$ satisfies the Gaussian distribution, and $\theta = 1$ indicates the primary user's presence, while $\theta = 0$ indicates its absence.

$y(t)$ is the received signal by the CR user which can be expressed by as in Equation (8)

$$y(t) = \int_0^t [\theta s(\tau) + n(\tau)]h(t-\tau)d\tau \tag{8}$$

When $t = t_0$, the CR user receives the signal. When the primary user's signal exists H_1, $s_1 = s(t) + n(t)$, then

$$y(t_0) = \int_0^{t_0} [s(\tau) + n(\tau)]s(\tau)d\tau = E_1 + Z \tag{9}$$

The received signal's probability distribution is given as in Equation (10):

$$p(y|s_1) = \frac{1}{\sqrt{\pi N_0 E_1}} exp[-\frac{(y-E_1)^2}{N_0 E_1}] \tag{10}$$

When the primary user's signal does not exist H_0, $s_2 = n(t)$, then

$$y(t_0) = \int_0^{t_0} n(\tau)s(\tau)d\tau = Z \tag{11}$$

The probability distribution of the received signal is expressed as in Equation (12):

$$p(y|s_2) = \frac{1}{\sqrt{\pi N_0 E_1}} exp[-\frac{y^2}{N_0 E_1}] \tag{12}$$

Define the judgment threshold as V . The probabilities of false alarm, detection, and missed detection can be expressed as follows:

$$P_{fa} = p(y > V|H_0) = \frac{1}{2} erfc(\frac{V}{E_1}\sqrt{\frac{E_1}{N_0}}) \tag{13}$$

$$P_d = p(y > V|H_1) = 1 - \frac{1}{2} erfc(\sqrt{\frac{E_1}{N_0}} - \frac{V}{E_1}\sqrt{\frac{E_1}{N_0}}) \tag{14}$$

$$P_{md} = p(y < V|H_1) = \frac{1}{2} erfc(\frac{E_1 - V}{\sqrt{N_0 E_1}}) \tag{15}$$

The effect of shadowing loss ζ is also taken into account here. When $E_2 = E_1 10^{\frac{\zeta}{10}}$, the probabilities of false alarm and detection for matched filter detection can be expressed as in Equation (16) and Equation (17):

$$P_{fa} = \frac{1}{2} erfc(\frac{V}{E_2}\sqrt{\frac{E_2}{N_0}}) \tag{16}$$

$$P_d = 1 - \frac{1}{2} erfc(\sqrt{\frac{E_2}{N_0}} - \frac{V}{E_2}\sqrt{\frac{E_2}{N_0}}) \tag{17}$$

4 Simulation and Discussion

The influence on the detection performance from the probability of false alarm and shadow fading is simulated and discussed in energy and matched filter detection, respectively. The sampling point is N=1024.

4.1 Influence on Detection Performance from the Probability of False Alarm

Figure 3 shows that when the number of sampling point N is selected, the higher the probability a false alarm, the higher the probability of detection, and the lower the probability of missed detection.

In matched filter detection, the effect of the false alarm probability on the detection performance is depicted in Figure 4. It is evident that the higher the signal-to-noise ratio, the greater the probability of detection, and the lower the possibility of missed detection, the greater the probability of a false alarm. This is because when the false alarm probability is increased, equivalent to a lower limit on the system, the probability of missed detection will be reduced accordingly.

It can also be seen that the threshold has the same increasing and decreasing properties for the probability of detection and false alarm. The higher the probability of a false alarm, the higher the probability of detection, and the lower the probability of missed detection when the signal-to-noise ratio is calculated. The higher the signal-to-noise ratio, on the assumption that the probability of a false alarm is calculated, the greater the probability of detection.

4.2 Influence on Detection Performance from Shadow Fading

The following simulations examine how shadow fading affects the performance of energy detection and matched filter detection, respectively, under the assumption that the probability of false alarm is 0.01.

According to Figure 5, the probability of detection affected by shadow fading is lower than the probability of detection without shadow fading for the same probability of a false alarm; The probability of missed detection when

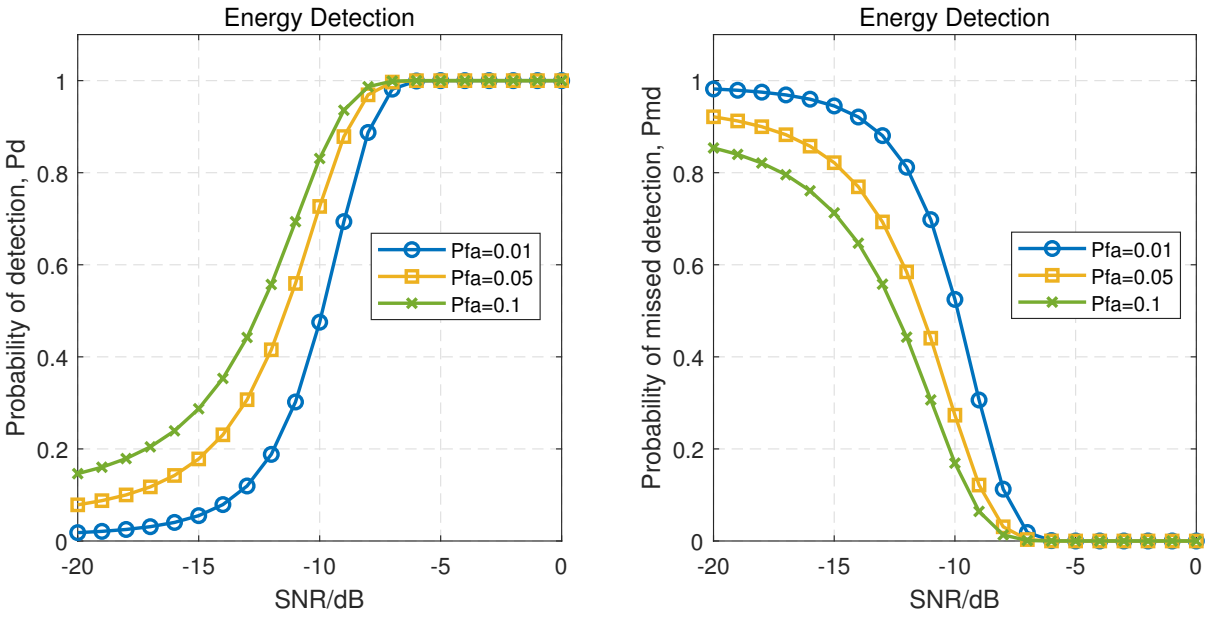


Figure 3: Influence on detection performance from different probability of false alarm in energy detection

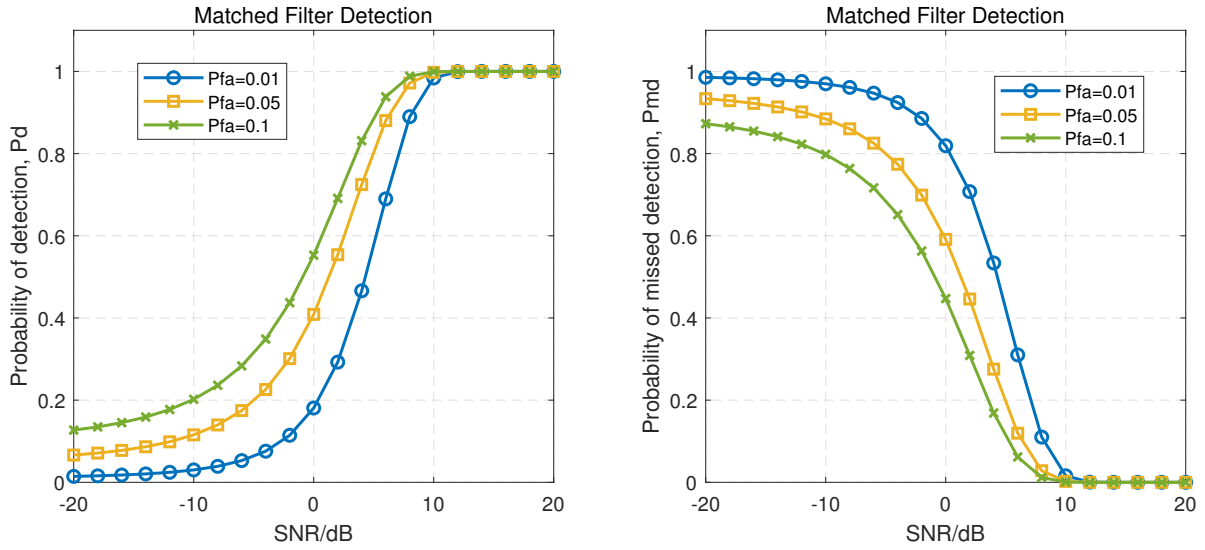


Figure 4: Influence on detection performance from false alarm probability in Matched Filter detection

shadow fading is present is greater than the probability of missed detection when shadow fading is not present. Other approaches (like the multi-node cooperative detection algorithm) are required to reduce since shadow fading impacts detection performance significantly.

According to Figure 6, the probability of detection affected by shadow fading is lower than that of detection without shadow fading for the same probability of a false alarm. In comparison, the probability of missed detection affected by shadow fading is higher than that of missed detection without shadow fading. The impact of shadow fading on detection performance is very large, so other methods, such as the multi-node cooperative detection method, are needed to overcome the impact of shadow fading.

Table 1 displays the comparison study for the energy and matched filter detection under various shadow fading when the SNR=-10dB. As seen, matched filter detection's sensing performance is more susceptible to shadow fading than energy detection.

5 Conclusions

Cognitive radio aims to solve the spectrum scarcity issue by implementing spectrum sensing technology. In fact, various security issues and vulnerabilities experienced can influence network performance, such as malicious user attacks and shadow fading. Shadow fading's effects on energy detection and matching filter detection are examined in this research. The formulas for detection probability and false alarm probability are constructed for the energy detection and matching filter detection technologies. The probability of detection and missed detection under different false alarm probabilities and shadow fading is simulated and discussed, respectively. Shadow fading has a more significant impact on matched filter detection than energy detection, as the simulation results show.

Acknowledgments

This work was supported by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJQN202101123, KJQN202101110, KJQN202101128). This work was supported by 2021 Chongqing Doctoral Through Train scientific research project (No.CSTB2022BSXM-JCX0118, (No.CSTB2022BSXM-JCX0119). This work was supported by scientific research start-up fund project of Chongqing University of Technology (No.2020ZDZ013) (No.2022ZDZ013).

References

[1] I. K. Aulakh and N. Kaur, "Optimal sensing simulation in crns under shadow-fading environments," in *3rd International Conference on Computing for Sustainable Global Development (INDIACom'16)*, pp. 901–905, 2016.

[2] M. A. Aygül, H. M. Furqan, M. Nazzal, and H. Arslan, "Deep learning-assisted detection of pue and jamming attacks in cognitive radio systems," in *IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pp. 1–5, 2020.

[3] D. Bera, I. Chakrabarti, S. S. Pathak, and G. K. Karagiannidis, "Another look in the analysis of cooperative spectrum sensing over nakagami- m fading channels," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 856–871, 2017.

[4] A. Brito, P. Sebastião, and F. J. Velez, "Hybrid matched filter detection spectrum sensing," *IEEE Access*, vol. 9, pp. 165 504–165 516, 2021.

[5] G. Chandrasekaran and S. Kalyani, "Performance analysis of cooperative spectrum sensing over $\kappa-\mu$ shadowed fading," *IEEE Wireless Communications Letters*, vol. 4, no. 5, pp. 553–556, 2015.

[6] J. C. Clement and K. C. Sriharipriya, "Robust spectrum sensing scheme against malicious users attack in a cognitive radio network," in *International Conference on Electrical, Computer and Energy Technologies (ICECET'21)*, pp. 1–4, 2021.

[7] S. Dhananjaya and B. N. Yuvaraju, "A novel method in matched filter spectrum sensing to minimize interference from compromised secondary users of cognitive radio networks," in *International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT'18)*, pp. 228–231, 2018.

[8] H. Guo, N. Reisi, W. Jiang, and W. Luo, "Soft combination for cooperative spectrum sensing in fading channels," *IEEE Access*, vol. 5, pp. 975–986, 2017.

[9] H. Huang and C. Yuan, "Cooperative spectrum sensing over generalized fading channels based on energy detection," *China Communications*, vol. 15, no. 5, pp. 128–137, 2018.

[10] S. Kavaiya and D. K. Patel, "On the performance of an improved energy detector over shadow fading channels for vehicular networks," in *14th International Conference on COMmunication Systems and NETworkS (COMSNETS'22)*, pp. 275–279, 2022.

[11] R. T. Khan, S. Zaman, M. I. Islam, and M. R. Amin, "Performance evaluation of cognitive radio network under matched filter detection," in *International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC'17)*, pp. 1196–1201, 2017.

[12] M. Lebepe and M. Velempini, "Mitigation of denial of service attacks in software-defined cognitive radio networks," in *International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD'21)*, pp. 1–5, 2021.

[13] M. Y. Morozov, O. Y. Perfilov, N. V. Malyavina, R. V. Teryokhin, and I. Chernova, "Combined approach to ssdf-attacks mitigation in cognitive radio networks," in *Systems of Signals Generating and*

Table 1: The probability of detection with the different shadow fading. (SNR=-10dB)

shadow fading	Pd of energy detection	Pd of matched filter detection
0	0.83	0.03
-1	0.72	0.02
-5	0.26	0.01
-10	0.15	0.007

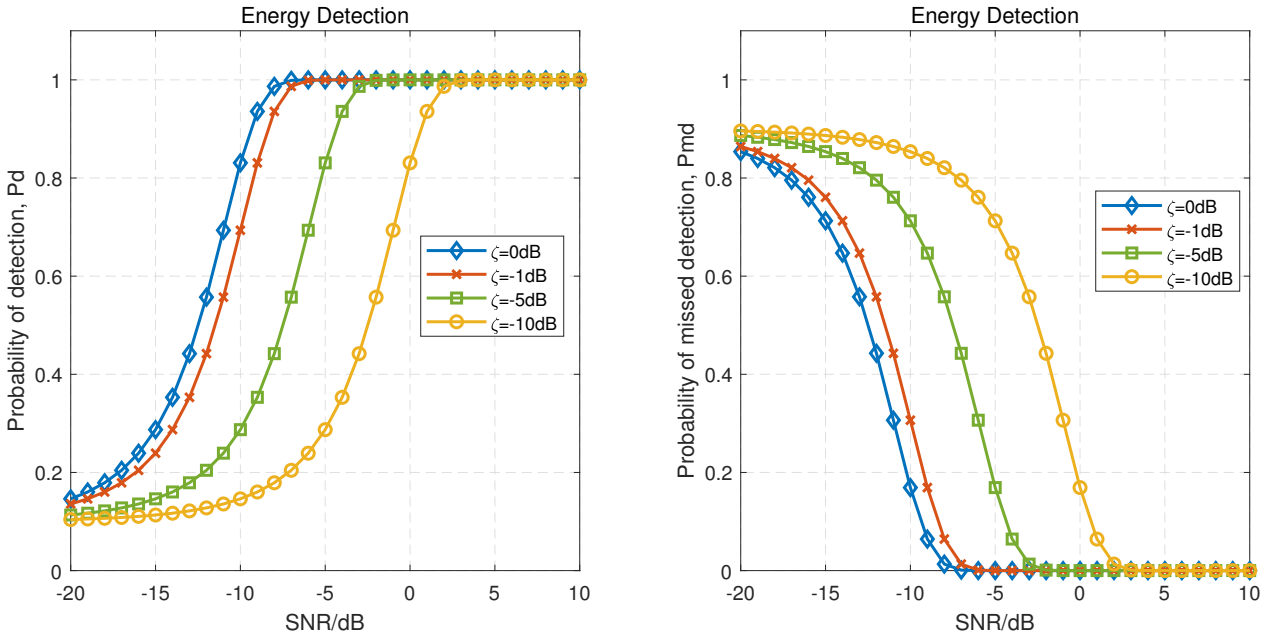


Figure 5: Influence on detection performance from the different shadow fading in energy detection

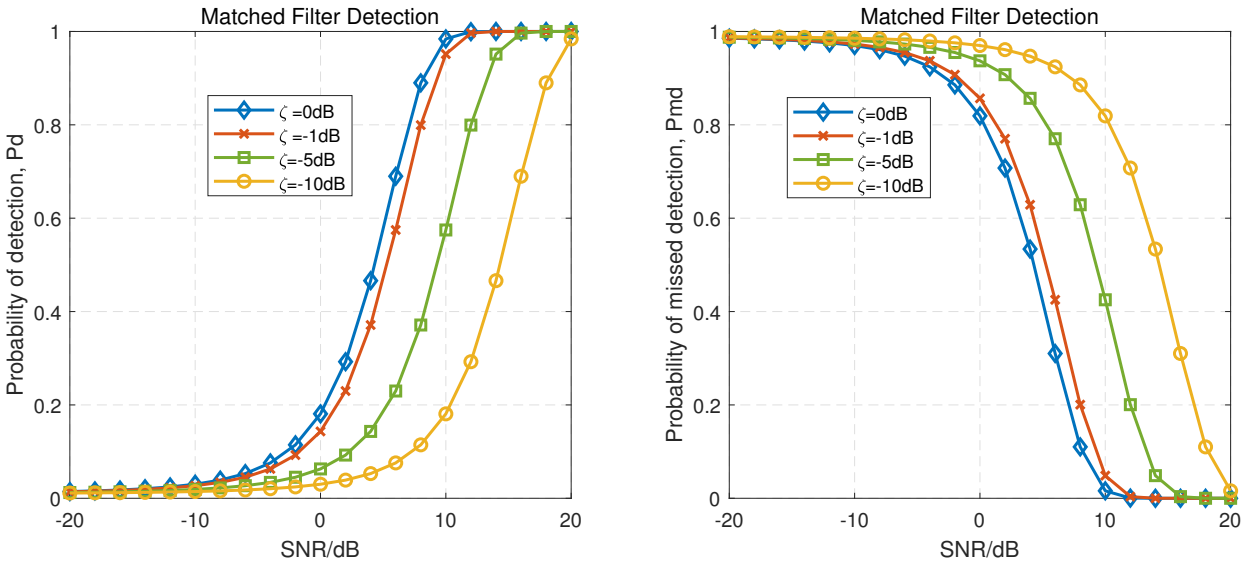


Figure 6: Influence on detection performance from the different shadow fading in Matched Filter detection

Processing in the Field of on Board Communications, pp. 1–4, 2020.

[14] N. P. Mwanza and J. Kalita, "Detecting ddos attacks in sdn using deep learning techniques: A survey," *International Journal of Network Security*, vol. 25, no. 2, pp. 360–376, 2023.

[15] S. R. Patil, R. Rajashree, and J. Agarkhed, "A survey on byzantine attack using secure cooperative spectrum sensing in cognitive radio sensor network," in *6th International Conference on Computing Methodologies and Communication (ICCMC'22)*, pp. 267–270, 2022.

[16] H. Rasheed, N. Rajatheva, and F. Haroon, "Spectrum sensing with energy detection under shadow-fading condition," in *IEEE 5th International Symposium on Wireless Pervasive Computing*, pp. 104–109, 2010.

[17] N. Reddy, P. S. Poojitha, M. Kumar, and K. Ramya, "Reducing the sensing errors by adopting the effective matched filter threshold estimation in lower snr conditions," in *International Conference on Emerging Trends in Science and Engineering (ICESE'19)*, vol. 1, pp. 1–4, 2019.

[18] S. Sedaghat, "The forensics of ddos attacks in the fifth generation mobile networks based on software-defined networks," *International Journal of Network Security*, vol. 22, no. 1, pp. 41–53, 2020.

[19] G. Sharma and R. Sharma, "Distributed cooperative spectrum sensing over different fading channels in cognitive radio," in *International Conference on Computer, Communications and Electronics (Comptelix'17)*, pp. 107–111, 2017.

[20] P. C. Sofotasios, A. Bagheri, T. A. Tsiftsis, S. Freear, A. Shahzadi, and M. Valkama, "A comprehensive framework for spectrum sensing in non-linear and generalized fading conditions," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8615–8631, 2017.

[21] C. Sorrells and L. Qian, "Quickest detection of denial-of-service attacks in cognitive wireless networks," *International Journal of Network Security*, vol. 16, no. 6, pp. 468–476, 2014.

[22] K. Sudha, K. A. Kumari, and D. Varunika, "A critical survey on security issues in cognitive radio networks," in *International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS'23)*, pp. 292–297, 2023.

[23] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

Biography

Hong Du is a Lecturer/ Master supervisor in Chongqing University of Technology, she received PhD from Beijing University of Post and Technology in 2012 and worked at School of Electrical and Engineering in Chongqing University of Technology. Her research interests include cognitive network and information security.

Dacheng Wang is a Senior Engineer, he received PhD from Northeast Petroleum University in 2021, and now worked in Chongqing University of Technology. His research interests include network and information security.

Long Chen is a Lecturer, he received PhD from Chongqing University in 2018, and then worked at School of Electrical and Engineering in Chongqing University of Technology. His research interests include network and information security, wireless communication technology.

Research on SDN Security Detection Algorithm Based on Bayesian Network

Zhiyong Luo, Shuyi Wang, Haifeng Xu, and Yu Zhang
(Corresponding author: Zhiyong Luo)
School of Computer Science and Technology, Harbin University of Science and Technology, China
Email: luozhiyongemail@sina.com
(Received Nov. 27, 2022; Revised and Accepted July 30, 2023; First Online Aug. 31, 2023)

Abstract

This paper proposes a security detection algorithm for Software-Defined Networks (SDN) based on Bayesian networks, which constructs SDN six-tuple, calculates attack cost, attack benefit, vulnerability value, and attack preference, and uses PageRank to calculate device importance to establish an attack detection model that predicts intrusion paths. It fills the gap in current SDN detection methods that do not consider the impact of controller vulnerabilities and attack costs. Through experimental comparison, the proposed model can more accurately calculate the probability of device attacks and intrusion paths, effectively providing a basis for SDN monitoring and protection.

Keywords: Attack Detection Model; Intrusion Paths; PageRank; SDN Security Detection

1 Introduction

Software-defined network SDN [9] is a new trend architecture that separates the controller and forwarding plane, which can use software programming to define and control the network, and use the characteristics of open programmability to improve the agility and efficiency of the network. It is considered to be a historic revolution in the network field, which provides a new experimental approach for the research on the new internet architecture, and speeds up the development of the next generation network.

SDN separates the control plane, which implements network decisions, from the data plane, which transmits packets. The control layer uses the SDN controller to communicate with the infrastructure layer (such as network switches/routers) through the southbound interface, and communicate with the application through the northbound interface, thereby obtaining the overall network status, and managing the programmable network to dynamically adjust the network. However, the security issues, risks, and threats become more complicated [11] on account of the SDN system framework. The introduction of new network components in SDN to support new network functions, such as SDN controllers and switches, opens up vulnerabilities in SDN that were not considered in traditional networks. In the SDN architecture, the controller is one of the core components, so the security of the controller is the key to the normal operation of the entire SDN. Therefore, this paper focuses on the analysis of the security of the controller. Previous research on controller security includes that the network is vulnerable to DDoS attacks in SDN controllers, the PATGEN [2] protocol is proposed and an advanced GENetic algorithm is used to reduce the impact of attacks, thereby significantly improving the efficiency of SDN controllers; Sahand *et al.* [14]. proposed a meta-heuristic algorithm (GSOCCPP) to solve the problem of network communication delay in the background of SDN, which realizes the shortest execution time of the algorithm, and proves that it is highly efficient in terms of other algorithms with controller placement through sufficient experiments; Miriyala Suneel *et al.* [12]. proposed a two-key method based on fully homomorphic encryption and a complex hybrid structure scheme, which includes a double decryption method with fully homomorphic encryption in the software-defined network (SDN) controller to ensure the security of encrypted data and reduce computing and communication costs; In the study of controller placement, Li Yi *et al.* [5]. proposed a comprehensive delay controller placement model to reduce the delay between the controller and the switch, as well as between the controller; Lu Sungho *et al.* [10]. proposed a defense method for HTTP DDoS flood attacks based on SDN, which can effectively defend web servers, protect network resources and resist HTTP DDoS flood attacks and so on.

The PageRank (PR) algorithm [4] is proposed to evaluate the importance of a device, which considers not only the effect of the relationship between the number of node neighbors on the node but also the effect of its location on the importance of the node. It facilitates the analysis of the cost of member attacks, moreover its calculation of the importance of nodes in the network has been applied in practice.

The traditional Intrusion Detection System (IDS) can only analyze the dependencies between node vulnerabilities after the attack occurs, and then monitor the attack behavior, which belongs to passive defense [7]. It is impossible to conduct a systematic security risk assessment on the network, and effectively protect against potential risks in the unknown network, such as multi-part attacks with hidden trajectories [16]. Therefore, attack graph [6] is used in this paper to predict the network attack path, as well the Bayesian attack graph is one of the most commonly used prediction methods. There are many papers on Bayesian attack graphs, including Asvija *et al.* [3], proposed utilizing the concept of Bayesian attack graph, aiming to investigate the Bayesian attack graph of the IaaS model to reveal sensitive areas for protecting high-risk components within stack; Shailendra *et al.* [8]. explored the visualization of attack graphs in public cyberspace to predict paths, as well created network attack maps by combining the graph adjacency matrix to identify gray areas and research points, so as to realize network security and management; Shawly Tawfeeq *et al.* [13]. proposed a new solution to address the challenges of modeling and detecting complex network attacks. The utilization of Hidden Markov Models (HMMs) to detect and track the progress of attacks, coupled with an analysis of attack risk probability and detection error rate aims to enhance the accuracy and efficiency of intrusion detection systems in tackling complex network attacks. Aaron Zimba *et al.* [18]. proposed a Bayesian network-based weighted attack path modeling technique to simulate attack paths by using quantitative induction to represent weighted total paths, considering marginal and conditional probabilities jointly to represent multiple paths from the attack point to the target node.

In this paper, the Bayesian attack graph is used to predict the attacker's attack intention for SDN, the PageRank algorithm is used to obtain the device criticality, coupled with the SDN attack graph is constructed by combining the vulnerability value, attack cost, attack benefit and preference function. A risk assessment model is established to predict the intrusion path. Finally, the effectiveness of the model is verified by simulation experiments.

The contributions of this paper are mainly in the following three aspects:

1) Considering the purpose of the attacker's aggressive behavior, use the PR algorithm to calculate the importance of the device, followed by calculating the attack cost based on the attack experience. Subsequently, the device attack probability is determined by considering four indicators: attack revenue, attack preference, and device vulnerability in order to predict the attackers' collective paths more accurately.

2) The concept of preference function is proposed, which makes the path conditional probability from the parent node to the child node more detailed, improves the accuracy of transfer probability between device nodes, and avoids the omission of path transfer behavior.

3) Perform risk assessment on the network and generate an intrusion path, calculate the reachability probability of each device in the path to realize the prediction of the intrusion path, furthermore improve the accuracy of the prediction.

2 SDN Attack Graph Establishment

The SDN attack graph enhances the device information from the SDN controller based on the traditional Bayesian attack graph, resulting in more comprehensive device attributes. Additionally, it utilizes attribute attack graphs within the attack graph framework, providing better adaptability to complex scenarios in SDN. To more accurately calculate the intrusion probability and possible intrusion path of each vertex in the SDN attack graph, this paper uses the PR algorithm to calculate the importance of the device and conducts a risk assessment to predict the intrusion path.

Definition 1. *SDN attack graph: It is a directed acyclic graph, which consists of the device information set I, node attribute set B, attack node intent F, attack process set E, node relationship combination S and possible probability P of the six-tuple, which is expressed as $SDNBAG = (I, B, F, E, S, P)$ in this paper, and its definition is as follows:*

1) I is the device information set, $I = \{I_i | i = 1, 2..., n\}$, where $I_i = (h, W, P, C)$, h is the device name, W is the device importance, Q is the influence degree of the device, C is the scope effect size; B is the node attribute set, which can be divided into three categories, where B_{start} represents the starting node of the attacker's offensive action, $B_{process}$ represents the process node in the attacker's attack path, and B_{target} represents the attacker. The target node to be attacked can be expressed as $B = B_{start} \cup B_{process} \cup B_{target}$.

2) F is the intention of the attacking node, which is expressed as the attacker's preference for attacking a child node of a device.

3) E is the attack process set, $E = \{e_i | i = 1, 2, ..., n\}$, e_i is the attack process of the attacker exploiting the system equipment vulnerability to attack from one node to another node, which expressed as $e_i \in B_{pre} \rightarrow B_{next}$, and E belongs to the set of directed edges.

4) S is the combination of node relationships, and the relationship reaching the target node can be expressed as a two-tuple $\langle B_j, d_j \rangle$, where $B_j \in B_{target}$, $d_j \in \{AND, OR\}$, $d_j = AND$ means that all parent nodes of device B_j can reach B_j. Similarly, $d_j = OR$ means

that the attack can be successful when a certain parent node of B_j can reach B_j.

5) *P is the reachability probability, that is, the reachability probability of an attacker from invading the parent device node to the child device attribute node.*

3 SDN Security Assessment

To evaluate the security of SDN more comprehensively in the experiment, this paper defines the vulnerability value of SDN equipment and the criticality of SDN equipment, and uses the attack cost and attack benefit combined with the PR algorithm to assess the importance of each network device in SDN.

3.1 Vulnerability Value Analysis

The vulnerability value of a node is related to the difficulty of its vulnerability being exploited by an attacker as well the impact of this vulnerability on the node itself, and the vulnerability is usually quantified by using the Common Vulnerability Scoring System (CVSS). This paper uses CVSS to measure six indicators [1], namely attack vector (AV), attack complexity (AC), privileges required (PR), confidentiality (C), integrity (I) and availability (A). The corresponding specific scoring standards are shown in Table 1.

The vulnerability value is quantified according to the above indicators, then the corresponding vulnerability score is obtained. The corresponding calculation formula is shown in Equation (1):

$$\text{Grade} = \text{Min}(Exp + Impact, 10) \tag{1}$$

Among:

$$Exp = 8.22 * AV * AC * PR$$
$$Impact = 6.42 * ISCbase$$
$$ISCbase = 1 - ((1 - C) * (1 - I) * (1 - A))$$

Among them, $Impact$ represents the vulnerability impact factor, where the default scope is fixed, Exp represents the vulnerability exploit factor, which size is the difficulty of being attacked, and $ISCbase$ represents a temporary intermediate variable.

Definition 2. *Vulnerability Value: This paper uses $Worth$ to quantify the vulnerability value since the material range of the CVSS scoring standard is [0,10], and the calculation formula is shown in Equation (2):*

$$Worth = Grade / 10 * 100\% \tag{2}$$

3.2 Analysis of the Importance of SDN Devices

The control plane and data plane in SDN are separated since SDN is different from the traditional network structure, so the calculation method of SDN importance is also different. According to the characteristics of SDN, it can be known that each device has specific function, hence the importance varies from light to heavy. Since the controller plays a core role in the entire SDN, its importance is the highest, followed by switches, servers, and the host. In this paper, the PageRank algorithm is used to analyze the criticality of the equipment.

Definition 3. *SDN device criticality PR: Indicates the criticality of the device in the entire SDN, which is determined by the role it plays in the given network topology. Due to the particularity of SDN, this paper sets the criticality of the initial device as an integer between [1,10].*

In the initial stage, set the PR value of the host to 4, the PR value of the switch and server to 7, and the PR value of the controller to 10, moreover to recalculate the device criticality according to the PR algorithm. The $PR(B_j)$ of network devices B_j $(j = 1,_{...}n)$ is shown in Equation (3):

$$PR(B_j) = \frac{1-d}{N} + d \sum_{c=1}^{M(B_j)} \frac{PR(B_i)}{O(B_i)} \tag{3}$$

Among them, the damping coefficient d, which the default is d=0.85, N represents the number of devices, $PR(B_i)$ represents the device criticality of the parent node B_i of the device B_j; $O(B_i)$ represents the number of devices B_i connected to the device B_j and $M(B_j)$ is the total number of devices from the parent node to device B_j.

3.2.1 Detailed Steps of PR Algorithm

The initialization matrix $S_{n\times n}$ of the device node is constructed according to the SDN topology relationship, which represents the attack probability from the parent node x to the child node y, and e is a column vector with all components equal to 1. The calculation formula of the obtained transition matrix K is shown in Equation (4):

$$K = dS + \tfrac{(1-d)}{N} ee^T \tag{4}$$

Set the unit column vector X, and perform the iteration. End the iteration to obtain the final criticality of all equipment if the values of X and PR are similar or the same, end the iteration to obtain the final criticality of all equipment, that is, $|PR - X| \leqslant \tau$, which τ is an infinitesimal quantity. The formula is shown in Equation (5):

$$PR = K * X \tag{5}$$

Calculate the equipment importance EIM_j according to the equipment criticality and vulnerability value, and then quantify it according to the calculation needs of this paper. The calculation formula of the quantized equipment importance EIM_j is shown in Equation (6):

$$\text{EIm}(B_j) = PR(B_j) \times Worth(B_j) / 10 \tag{6}$$

Table 1: CVSS Indicator Score

Index	Property measure	Factor score
Attack vector (AV)	Network relationship(N)	0.85
	Adjacent equipment (A)	0.62
	Local device (L)	0.55
	Physics (P)	0.20
Access Complexity (AC)	Low complexity (L)	0.71
	Medium complexity (M)	0.61
	High complexity (H)	0.35
Privileges Required(PR)	No permission required (N)	0.85
	Low permission requirements (L)	0.64
	High permission requirements (H)	0.33
Influence (C, I, A)	None (N)	0.00
	Low(L)	0.22
	High(H)	0.56

Algorithm 1 SDN device importance algorithm

Input: damping coefficient d, number of devices N, N-dimensional unit column vector X, N-dimensional unit column vector e, vulnerability value $Worth$, infinitesimal τ, initial matrix S.
Onput: SDN device importance EIM.
Step1: Initialize S, $d = 0.85$, N=20, τ=0.000001, $X = (1, \ldots, 1)^T$, $Worth$, $e = (1, \ldots, 1)^T$
Step2: $E = ee^T$
Step3: $C = (1 - d)/N$
Step4: Finding transition matrix
Step5:$K = dS + CE$
Step6: while $|PR - X| > \tau$
Step7: record $U = PR$
Step8: update $PR = K * X$
Step9: record $X = U$
Step10: record $EIM = PR \times Worth/10$

3.3 Attack Cost Analysis

The attacker not only pays human resources and material resources but also bears the attack cost caused by the attack when the attacker attacks offensive the device in the network. The cost of an attack is primarily determined by two factors: the probability coefficient (referred to as the risk coefficient β) of the security software detecting the attack when launched by the attacker against the targeted network, and the attacker's experience with attacking the specific node (referred to as the attack experience ξ). The risk factor β is determined by the importance $EIM\,(B_j)$ of the attacked device, the greater the importance of the node, the higher the probability of being discovered, and the greater the risk factor. The quantification of attack experience ξ is shown in Table 2.

From the above analysis, the calculation formula of equipment risk coefficient $\beta\,(B_j)$ can be obtained as shown in Equation (7):

$$\beta\,(B_j) = \mathrm{EIm}\,(B_j) * \zeta\,(B_j) \tag{7}$$

Definition 4. *Equipment attack cost (B_j): It means the cost that the attacker needs to pay when he launches an attack on the equipment in the target network. The attack experience will gradually increase during an attacker attacks a device node, so increase the coefficient f, which is given by experts according to different network environments. In this paper, the attack cost of human and material resources required to attack each device by default is $HrA\,\mathrm{Cos}\,t\,(B_j) = 0.01$, the calculation formula of the equipment attack cost can be obtained as shown in Equation (8):*

$$\mathrm{cos}\,t\,(B_j) = f^{n-1} * \beta\,(B_j) + HrA\,\mathrm{Cos}\,t\,(B_j) \tag{8}$$

3.4 Attack Benefit Analysis

Definition 5. *Device attack profit $profit$: It means the profit obtained by the attacker when he launches an attack on the device in the target network. The device attack profit index is shown in Table 3.*

The *profit* will make a profit value judgment based on the measurement information, and obtain an accurate profit value according to the amount of information leakage. The more important the leaked information, the higher the profit, and the specific situation is set by the administrator according to the network conditions on which the device is located.

3.5 Attack Preference Analysis

Definition 6. *Preference Function (PF): It represents the preference degree of the attacker to attack the target device node, the higher the preference degree, the higher the possibility of the attacker attacking the target device node.*

The preference function is mainly judged by the attack cost and the attack benefit. The size of the preference function is judged according to its ratio, that the higher the ratio, the lower the preference function and the lower

Table 2: Quantitative Criteria for Attack Experience

Serial number	Experience	Description of experience information
A1	0.1	The attacker's internal information does not have a record of past attacks on the device, but it may find the device's vulnerability
A2	0.2	The attacker's internal information has no previous attack records on the device, but knows its vulnerabilities and does not know the attack method.
A3	0.3	The attacker's internal information does not have any previous attack records on the device, but will refer to possible attack methods.
A4	0.4	The attacker's internal information has a rough attack method
A5	0.5	There are approximate attack steps, but no attack tools
A6	0.6	There are detailed attack steps, but no attack tools
A7	0.7	There are attack codes and detailed tool steps, but no attack tools
A8	0.8	There are attack codes, detailed tool steps and attack tools
A9	0.9	All are available and ready

Table 3: Attack Profit Scoring Metrics

Income level	Measure information	Profit value
PL1	Internal information leakage	0.3-0.55
PL2	Log in to the device remotely	0.55-0.7
PL3	Authentication bypass	0.7-0.85
PL4	Temporary visit	0.85-0.95
PL5	Get Root permissions	1.0

the possibility of attacking the target device node; on the contrary, the probability is higher. Using λ to represent the ratio of cost to benefit, the calculation formula is shown in Equation (9):

$$\lambda = \frac{\cos t\left(B_j\right)}{\text{profit}\left(B_j\right)} \tag{9}$$

Then the calculation formula of preference function PF is as shown in Equation (10):

$$PF\left(B_j\right) = \begin{cases} 0, \lambda \geq 1 \\ 1-\lambda, 0 < \lambda < 1 \\ 1, \lambda = 0 \end{cases} \tag{10}$$

It can be seen from Equation (10) that $PF\left(B_j\right) \in [0,1]$, when $\lambda \geq 1$, the preference function is 0, and the cost is much greater than the benefit at this time, an attacker can't invasion the device; When $\lambda = 0$, it means that the benefit is far greater than the cost, and the preference function is 1, at this time, the attacker will attack the device.

3.6 Device Attack Probability Analysis

According to the above analysis of the importance of equipment, attack cost, attack cost, and the results of quantifying them, the probability of the attacker launching an offensive on the device to be attacked can be obtained, That is, the probability of the device being attacked, its range is [0, 1], the higher the probability, the greater the possibility of being attacked.

Definition 7. *Device attack probability: It indicates the probability that an attacker attacks the target device and successfully occupies the target node. For the device B_j, the calculation formula of its device attack probability $p\left(B_j\right)$ is shown in Equation (11):*

$$P\left(B_j\right) = \min\left(\frac{worth\ \left(B_j\right) * \text{profit}\left(B_j\right) + PF\left(B_j\right)}{\text{cost}\left(B_j\right)}, 1\right) \tag{11}$$

3.7 Conditional Probability Analysis

Definition 8. *Conditional probability: It indicates the possibility of being attacked by each device attribute node under the influence of its device parent node. Among them, the conditional probability of each node is called the local conditional probability distribution function. For the device attribute node B_j, its local conditional probability can be expressed as $P_c\left(B_j \mid P_a\left(B_j\right)\right)$, and the set of device parent nodes is expressed as $P_a\left(B_j\right)$, v_j represents the attack from the device parent node to the device child node.*

1) When $S = \langle B_j, d_j = AND \rangle$, all parent nodes of the device are capable of reaching its child nodes to ensure a successful attack. The calculation formula is shown in Equation (12):

$$P_c\left(B_j \mid P_a\left(B_j\right)\right) = \begin{cases} 0, \exists B_j \in P_a\left(B_j\right), B_j = 0; \\ \prod_{j=1}^{n} P_a\left(v_j\right), otherwise; \end{cases} \quad (12)$$

2) When $S = \langle B_j, d_j = OR \rangle$, as long as there is one device parent node that can reach the device child node, the attack can be successful. The calculation formula is shown in Equation (13):

$$P_c(B_j|P_a(B_j)) = \begin{cases} 0, \forall B_j \in P_a\left(B_j\right), B_j = 0; \\ 1 - \prod_{j=1}^{n}\left[1 - P_a\left(v_j\right)\right], otherwise; \end{cases} \quad (13)$$

Definition 9 *Device reachability probability: It represents the reachability probability of each device attribute node in the SDN, which is the joint conditional probability of the node passed by the attacked initial node in the current device attribute node, that is, $B_j \in B_{process} \cup B_{targot}$, the calculation formula for the reachability probability of the device attribute node B_j is shown in Equation (14):*

$$P_c\left(B_j\right) = \prod_{j=1}^{n} P_c\left(B_j \mid P_a\left(\ \mathrm{B}_j\right)\right) \quad (14)$$

Among them, $P_a\left(B_j\right)$ is the device parent node set of the device attribute node B_j, and $P_c\left(B_j \mid P_a\left(\ \mathrm{B}_j\right)\right)$ represents the reachability probability when the device attribute node B_j is attacked.

4 SDN Risk Assessment and Prediction of Intrusion Path

4.1 SDN Risk Assessment and Analysis

To more accurately predict the attack path of the attacker's aggressive on the SDN device, the conditional probability and prior probability of the device attribute node are obtained by calculation. The parameters in the SDN attack graph are assigned to obtain the final SDN attack graph, incorporating Bayesian theory, the proposed attack cost, as well as the attack benefit and preference function. The algorithm steps are as follows.

1) Initialize the device information set I, node attribute set B, attack process set E, and node relationship combination S in the SDN attack graph.

2) Use Equation (11) to calculate the probability of the attacker attacking the device node from the directed edge.

3) Use Equation (10) to calculate the attack device intention, Equations (12) and (13) to calculate the conditional probability of the attacker attacking from the parent device node to the child device node, and use Equation (14) to calculate the reachability probability.

4) Copy the obtained attack intent and probability into the parameter P, and return the final SDN attack graph.

Algorithm 2 SDN Risk Assessment Algorithm

1: Input: The device information set I in the SDN attack graph, the node attribute set B, the attack node intent F, and the node relationship combination S.
2: Onput: $SDNBAG = (I, B, F, E, S, P)$.
3: Step1: Initialize the device information set I, node attribute set B, attack node intent F, and node relationship combination S in the SDN attack graph;
4: Step2: for (each directed edge in $SDNBAG$)
5: Step3: Calculate $P(B_i)$ using Equation (11)
6: Step4: end for (2)
7: Step5:for (each attribute node B_i in $SDNBAG$)
8: Step6: use Equation (10) to calculate the attack node intent F
9: Step7: if($F = 1$)
10: Step8: $P_c(B_i) = P$
11: Step9: else
12: Step10: Calculate $P_c\left(B_i \mid P_a(B_i)\right)$ using Equations (12) and (13)
13: Step11: Calculate the reachability probability $P_c(B_i)$ using Equation (14)
14: Step12: end if (7)
15: Step13: end for (5)
16: Step14 Copy $P_c(B_i)$ into the parameter P
17: Step15: return $SDNBAG = (I, B, F, E, S, P)$;

4.2 Intrusion Path Analysis

Definition 9. *According to the SDN attack graph obtained in 4.1, it can be known that the path taken by the attacker from the starting device attribute node B_{start} to attack the target device node B_{target} is called the intrusion path. Among them, the path composed of device attribute nodes is the intrusion path IP of the SDN attack graph, and the intrusion path algorithm is shown in Algorithm 3.*

According to all the intrusion paths obtained in the above algorithm, find the path with the highest probability risk, which is the path of the device assaulted by the attacker.

5 Simulation Experiment

5.1 Experimental Environment Construction

To verify the effectiveness of the SDN attack graph for predicting attack paths, this paper uses two virtual machines and builds a simulated network environment as shown in Figure 1 for experiments.

Algorithm 3 Intrusion Path Generation Algorithm

Input: Parameters in the SDN Attack Graph
Onput: $IntPath = \{IP_1, \ldots, IP_n\}$
Step1: Initialize Parameters in the SDN Attack Graph;
Step2: for (each target node $B_i \in B_{target}$)
Step3: Add B_i to IP_i
Step4: if ($P_a(B_i)$!=None)
Step5: if ($d_i = OR$)
Step6: n=len ($P_a(B_i)$)
Step7: copy IP_i to $(IP_{i-1}, \ldots, IP_{i-n})$
Step8: for (each node $B_j \in P_a(B_j)$)
Step9: Add B_j to IP_{i-j};
Step10: end for (8)
Step11: else
Step12: Add $P_a(B_i)$ to IP_i;
Step13: $B_i \in P_a(B_i)$
Step14: end if (5)
Step15: else
Step16: return IP_i;
Step17:end if (4)
Step18: Add IP_i to $IntPath$;
Step19: end for (2)
Step20: return $Intpath$;

Figure 1 includes the SDN control plane and data plane as well as the external network. The control plane includes three controllers, namely S1, S2 and S3. The data plane includes five servers, six switches, two virtual machines, and a firewall. The intrusion detection system snort is installed on the firewall to monitor network traffic. Simulate the business network environment by using Web server, FTP server, VPN server, SSH server and DB server to. Create virtual switches by using Open vSwitch software for simulating Open Flow switches, represented by $W = \{W_i \mid i = 1, 2, \ldots, 6\}$. The external network consists of the host used by the attacker and the external router.

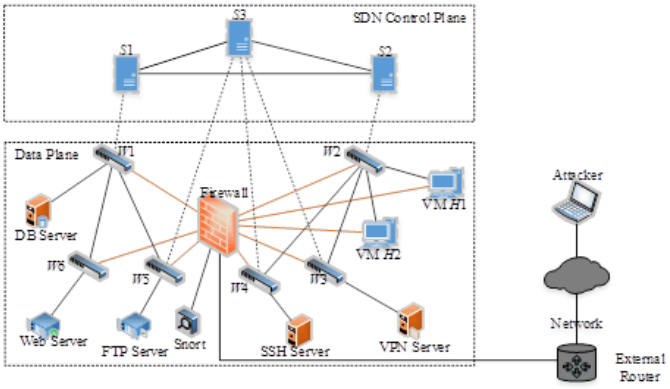


Figure 1: Lab environment

In this paper, OpenVAS is used to scan the device nodes, acquiring the vulnerability information of each device node in the SDN, and determining the vulnerability worth of the corresponding device node based on Equation (1) and Equation (2). The above information is summarized to obtain as shown in Table 4.

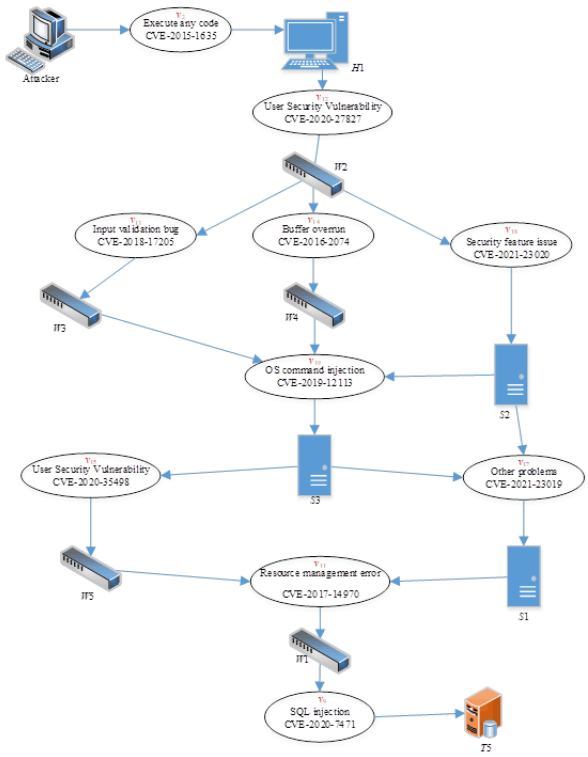


Figure 2: Attack graph in an experimental environment

5.2 Generation of Attack Graph and Intrusion Path

For the attacker, breaching the DB server of the target host yields the highest gains, as it contains a vast amount of information stored in its database. Therefore, this experiment sets the attacker to offensive the virtual machine H1 first, and sets the DB server as the target device for the final attack. This paper examines the comprehensive attack path by analyzing the relationship between the controller in the control plane and the device in the data plane, so an attack graph is generated using data such as vulnerability information and the SDN network topology, as shown in Figure 2. It can be observed that the attacker can invade the destination device child node from any one of the multiple device parent nodes successfully when there are multiple device parent nodes in the device child node based on Figure 2, so $d_j = OR$.

According to the attack intent analysis method proposed in this paper, the attack graph is predicted by using intrusion path algorithm of Figure 2, next seven feasible intrusion paths are generated. The specific information is shown in Table 5.

Table 4: Atomic Attack strategy

Device name	Device ID	Operating system or program	Vulnerability description	Vulnerability ID	CVE ID	Worth(vi)
VM H1	H1	LINUX	Buffer overrun	v1	CVE-2014-1443	0.41
		LINUX	Execute any code	v2	CVE-2015-1635	0.43
VM H2	H2	LINUX	Access restriction bypass	v3	CVE-2015-8467	0.35
		LINUX	Cross-site scripting	v4	CVE-2015-8622	0.39
VPN server	T1	Web VPN	Post-Auth Heap Overflow Vulnerability	v5	CVE-2018-13383	0.43
FTP server	T2	Titan FTP Server6.0.3	Arbitrary command execution vulnerabilities	v6	CVE-2014-8517	0.43
SSH server	T3	LINUX	Login Authentication Bypass Vulnerability	v7	CVE-2018-10933	0.57
Web server	T4	LINUX	Vulnerability recurrence	v8	CVE-2018-8715	0.39
DB server	T5	Postgre SQL	SQL injection	v9	CVE-2020-7471	0.52
		Postgre SQL	Buffer overrun	v10	CVE-2014-1669	0.41
W1 switch	W1	Open vSwitch	Resource management error	v11	CVE-2017-14970	0.55
W2 switch	W2	Open vSwitch	User Security Vulnerability	v12	CVE-2020-27827	0.62
W3 switch	W3	Open vSwitch	Input validation bug	v13	CVE-2018-17205	0.57
W4 switch	W4	Open vSwitch	Buffer overrun	v14	CVE-2016-2074	0.41
W5 switch	W5	Open vSwitch	User Security Vulnerability	v15	CVE-2020-35498	0.62
W6 switch	W6	Open vSwitch	Buffer overrun	v16	CVE-2017-9265	0.41
S1 controller	S1	NGINX	Other problems	v17	CVE-2021-23019	0.37
S2 controller	S2	NGINX	Security feature issue	v18	CVE-2021-23020	0.52
S3 controller	S3	ONAP SDNC	OS command injection	v19	CVE-2019-12113	0.55

Table 5: Intrusion path

path ID	Intrusion path	Number of intruded devices
IV1	<H1, W2, W3, S3, S1, W1, T5>	7
IV2	<H1, W2, W3, S3, W5, W1, T5>	7
IV3	<H1, W2, S2, S3, S1, W1, T5>	7
IV4	<H1, W2, S2, S1, W1, T5>	6
IV5	<H1, W2, S2, S3, W5, W1, T5>	7
IV6	<H1, W2, W4, S3, W5, W1, T5>	7
IV7	<H1, W2, W4, S3, S1, W1, T5>	7

5.3 Simulation Risk Calculation

To obtain the device attack probability of the device attribute node in each intrusion path, this paper first uses the PR algorithm to obtain the device criticality, and then uses the attack cost, attack benefit and attack preference to calculate the device attack probability.

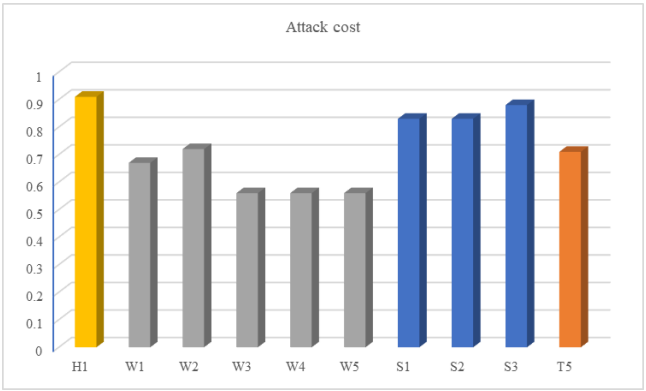


Figure 3: Device Attack Costs

Among them, the attack cost needs to be composed of attack experience and equipment importance. Find the attack experience of the equipment from Table 2 and use Algorithm 1 to calculate the equipment importance, at that moment, bring it into Equation (7) and Equation (8) to get the attack cost. The result is shown in Figure 3.

Calculate the attack probability of each device by substituting the attack cost of each device from Figure 3, the vulnerability value of each device from Table 4, the attack benefit of each device from Table 3, and the attacker's attack preference into Equation (11). The result is shown in Figure 4.

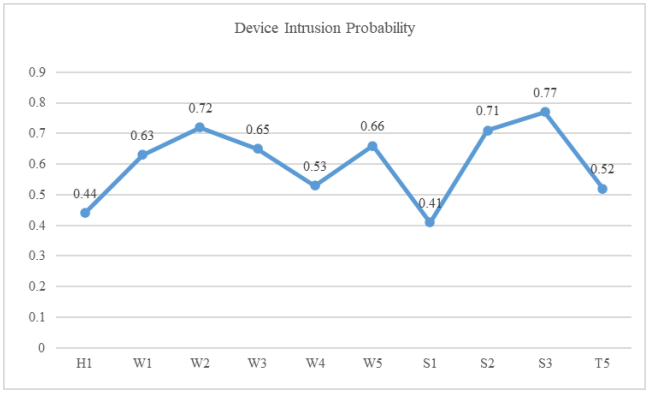


Figure 4: Device Intrusion Probability

According to Equation (14), the prior probability of each intrusion path is calculated as shown in Figure 5. Since the number of intrusion devices in the path IV_3 is lower than that of other paths, which has an impact on the experimental results, it has been omitted from the experimental results in this paper, and only the remaining paths are predicted. It can be seen from the figure that the intrusion risk of intrusion path IV_5 is Highest. Therefore, it is inferred that the intrusion path is $\langle H1, W2, S2, S3, W5, W1, T5\rangle$, and the higher the vulner-

ability value and preference of the device passed through, the higher the probability of being intruded.

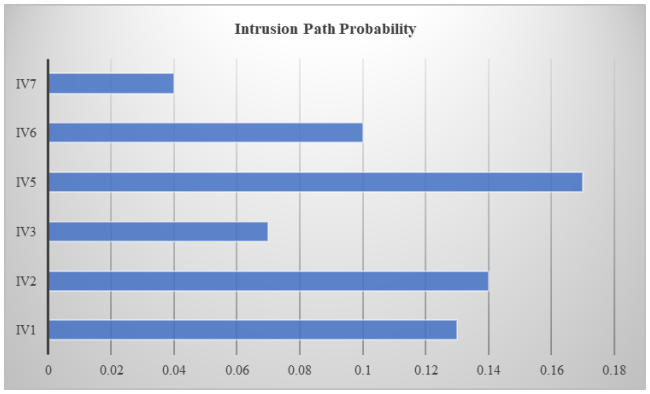


Figure 5: Intrusion Path Probability

5.4 Model Comparison

Since the prior probability of device nodes is the key to SDN security risk assessment, the prediction of intrusion paths can provide SDN administrators with a reliable basis for intrusion prevention. To verify the validity and superiority of the model proposed in this paper, it is compared with the experimental model proposed by Wang H *et al.* [15] and Yin Y S *et al.* [17] under the same SDN environment.

Reference 17 and 18 also uses a Bayesian attack graph to describe the attack behavior between SDN devices. However, because only the PR algorithm and vulnerability utilization are used for evaluation, the cost and benefits of attacking the device and the impact of the attacker's preference on the device are not considered. Therefore, it leads to inaccurate prediction of intrusion path. The reflection of its model on the intrusion probability of each device is shown in Figure 6. It can be seen that the model proposed in this paper is superior and the evaluation is more accurate.

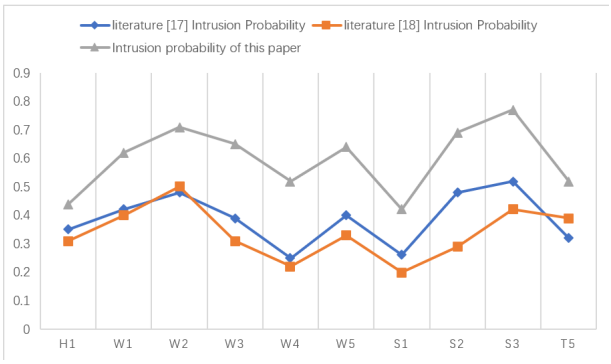


Figure 6: Device Intrusion Probability Comparison

After comparing the device attack probabilities of the three models, according to the intrusion path probability is the product of the probability of the Device Intrusion all devices on a certain attack path, the intrusion path probability of the attack path selected by these three algorithms is calculated, and the intrusion path probability of the algorithm with the highest probability can be obtained by Figure 7.

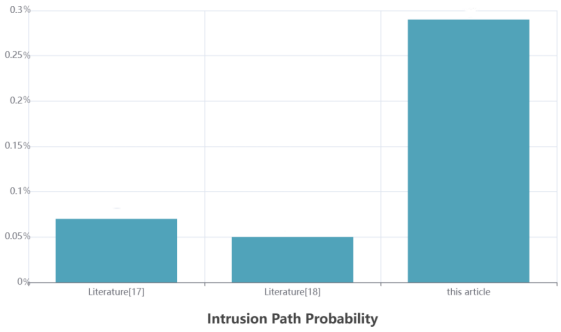


Figure 7: Intrusion Path Probability

It can be seen that the probability of intrusion path is improved, which verifies the superiority of the algorithm in this paper.

6 Conclusion

To protect the security of all information and devices in the SDN network, quantify network security risks for attack intent, and provide security policy support for SDN network security administrators. An SDN attack intent analysis model based on a Bayesian attack graph is proposed. First, use the PageRank algorithm to find the criticality of the device, and then combine it with the vulnerability value, attack cost, attack benefit and preference function to construct an attack graph as well as establish a risk assessment model to predict the intrusion path. The possible invasion paths are predicted, and the feasibility of this study is confirmed by comparing it with previous experiments. In a real SDN network, the correlation between vulnerabilities will also affect the probability of device attack. The next step that we will study this and optimize the algorithm and model of SDN intrusion risk assessment.

Acknowledgments

This study was supported by Heilongjiang Provincial Natural Science Foundation of China:LH2021F030. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. Abhishek, S. Sangeeta, and N. Sushama, "A hybrid scoring system for prioritization of software vulnerabilities," *Computers & Security*, vol. 129, 2023.

[2] I. Amir and R. N. Hamid, "A protocol for cluster confirmations of sdn controllers against ddos attacks," *Computers and Electrical Engineering*, vol. 93, 2021.

[3] B. Asvija, R. Eswari, and M. B. Bijoy, "Security threat modelling with bayesian networks and sensitivity analysis for iaas virtualization stack," *Journal of Organizational and End User Computing (JOEUC)*, vol. 33, no. 4, pp. 44–69, 2021.

[4] U. Jain, A. Mishra, B. Jaganathan, and P. Shukla, "Study and analysis of category based pagerank method," *Wireless Networks*, vol. 27, no. 8, pp. 1–16, 2021.

[5] Y. Li, S. Guan, C. Zhang, and W. Sun, "Parameter optimization model of heuristic algorithms for controller placement problem in large-scale sdn," *IEEE ACCESS*, vol. 8, pp. 151 668–151 680, 2020.

[6] X. Liu and et al, "A network attack path prediction method using attack graph," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–8, 2020.

[7] Z. Y. Luo, X. Yang, J. Liu, and R. Xu, "Network intrusion intent analysis model based on bayesian attack graph," *Journal on Communications*, vol. 41, no. 09, pp. 160–169, 2020.

[8] S. Mishra and et al, "Cyber defence using attack graphs prediction and visualisation," vol. 29, no. 3, pp. 268–289, 2023.

[9] Y. M. Nura, B. K. bin Abu, I. Babangida, O. A. Hamza, and N. Maged, "Adaptive path selection algorithm with flow classification for software-defined networks," *Mathematics*, vol. 11, no. 6, 2023.

[10] S. Park, Y. Kim, H. Choi, Y. Kyung, and J. Park, "HTTP DDoS flooding attack mitigation in software-defined networking:regular section," *IEICE Transactions on Information and Systems*, vol. 104, no. 9, pp. 1496–1499, 2021.

[11] M. Priyadarsini, P. Bera, S. K. Das, and M. A. Rahman, "A security enforcement framework for sdn controller using game theoretic approach," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, 2023.

[12] M. Suneel and S. M. Satya, "Improving privacy in sdn based manet using hybrid encryption and decryption algorithm," *Microprocessors and Microsystems*, p. 103501, 2020.

[13] S. Tawfeeq, E. Ali, K. Jason, and G. Arif, "Architectures for detecting interleaved multi-stage network attacks using hidden markov models," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 1–1, 2019.

[14] S. Torkamani-Azar and M. Jahanshahi, "A new gso based method for sdn controller placement," *Computer Communications*, vol. 163, pp. 91–108, 2020.

[15] H. Wang, J. Zhang, Y. Zhao, K. Liu, and W. F. Feng, "Network risk assessment method of a new type of bayesian model," *Journal of Chinese Computer Systems*, vol. 41, no. 09, pp. 1898–1904, 2020.

[16] K. Wang, W. Feng, and X. Li, "A new method of network risk assessment based on bayesian model," 2020.

[17] Y. S. Yin, T. P. Suo, L. G. Dong, and X. Jiang, "Sdn security prediction method based on bayesian attack graph," *Telecommunication Science*, vol. 37, no. 11, pp. 75–85, 2021.

[18] A. Zimba, H. Chen, and Z. Wang, "Bayesian network based weighted apt attack paths modeling in cloud computing," *Future Generation Computer Systems*, vol. 96, pp. 525–537, 2019.

Biography

Luo Zhiyong biography. Luo Zhiyong, male, was born in Shandong, China in July 1978.He is a professor at the School of Computer Science and Technology, Harbin University of Science and Technology. Research direction: computer network and information security, network optimization.

Wang Shuyi biography. Wang Shuyi, female, was born in Heilongjiang, China in June 2000. She is a postgraduate student in the School of Computer Science and Technology, Harbin University of Science and Technology. Research direction: network security.

Xu Haifeng biography. Xu Haifeng, male, was born in March 1999 in Jilin, China. He is a postgraduate student in the School of Computer Science and Technology, Harbin University of Science and Technology, Research direction: network security.

Zhang Yu biography. Zhang Yu is a postgraduate student in the School of Computer Science and Technology, Harbin University of Science and Technology, Research direction: network security.

Research on Key Management of Network Communication Protocols Based on IoT Security

Hongbo Yao and Li Yang
(Corresponding author: Li Yang)
School of Artificial Intelligence, Thangshan University, Tangshan, Hebei, China
Email: ry8349@163.com
(Received July 22, 2022; Revised and Accepted June 18, 2023; First Online Aug. 25, 2023)

Abstract

This paper briefly introduced the key management schemes based on the certificate public key and identity private key, and the key management scheme based on the certificateless public key evolved from the identity private key in Internet of Things (IoT) communication protocols. The communication overhead and time overhead of the three key management schemes were tested in the simulation experiments. The results showed that the communication overhead and time overhead of the key management scheme based on the certificate public key were the highest, the scheme based on the identity private key was the second highest, and the scheme based on the certificateless public key was the lowest.

Keywords: Certificateless Public Key; Communication Protocol; Internet of Things; Key Management

1 Introduction

The Internet of Things (IoT) is a network system that connects various sensors, devices, and equipment together for data exchange and communication over the Internet [27]. By using IoT technology, different devices can be connected to each other, allowing users to use and manage devices more intelligently, improving productivity and comfort [22]. It is also used in industrial automation, healthcare, transportation, and agricultural production. At the same time, the amount of data exchanged in the IoT is increasing [19], as is the amount of sensitive information it contains. The use of keys in communication protocols enables authentication between different devices in the IoT and encryption of data information [5, 7].

Kandi *et al.* [9] proposed a new multipurpose key management protocol for IoT. The experimental results verified that the solution ensured forward and backward secrecy and, unlike most existing group key management protocols, it guaranteed the secure coexistence of multiple services in the network. Tong *et al.* [21] proposed a threshold-based spatial key management scheme and applied it to the Consultative Committee for Space Data System (CCSDS) telecommunication control protocol. The results showed that the storage overhead of the scheme was lower than that of the ECC-based key management scheme [6, 8, 25].

Guermazi *et al.* [3] proposed an efficient and scalable key management and distribution framework, KMMR, for large-scale wireless sensor networks. The security analysis showed that KMMR could resist several known attacks. This paper briefly introduced the key management schemes based on certificate public key and identity private key in IoT communication protocols and a key management scheme based on certificateless public key evolved from identity private key. The communication overhead and time overhead of the three key management schemes were tested in the simulation experiments.

2 Key Management Schemes in IoT Communication Protocols

2.1 Certificate Public Key-based and Identity Private Key-based Key Management Schemes

An IoT communication protocol is a protocol used to enable different IoT devices to communicate and exchange information with each other. Similar to Internet protocols, specific IoT communication protocols are required to be used when transferring information between different IoT devices [24]. These protocols can specify information such as the communication method between devices, data format, and encryption method of communication protocols [12, 13, 15]. There may be many different options of IoT communication protocols for different types of devices [2, 20].

When selecting a communication protocol for an IoT device, the traffic and security that the communication protocol can handle must be considered to support seamless integration and efficient operation of the IoT device.

IoT communication protocols include the conversion of data formats between different devices [26], the form of data transmission between devices, and the encryption of transmitted data using keys. Among them, the function of using keys to encrypt or authenticate the transmitted data is the core of ensuring the security of IoT data transmission, so the key management in communication protocols is equally important. The key management scheme includes key generation, key distribution, key update, and key deletion [14, 17]. The early key management scheme in communication protocols is that both communication parties negotiate a key together and both parties use the key for encryption and decryption, but in this scheme, once the key is leaked, the encrypted communication data in IoT will also be leaked.

The commonly used key management scheme is based on the certificate public key mechanism management scheme [10], which has a third-party certificate authority (CA) responsible for issuing digital certificates, and the digital certificate contains the user's identity information and public key, and the private key corresponding to the public key is kept by the user himself. In the process of using this scheme, the sending user first applies for or verifies the digital certificate of the receiving user with the CA. After the application or verification is passed, the sending user can obtain the public key of the receiving user from the CA and use it to encrypt the information to be sent [23], and the receiving user decrypts the encrypted information with the private key kept by himself.

In the identity private key-based management scheme, no digital certificate is required, and the core of this key management scheme lies in the key server that generates the private key. In the process of using this scheme, the key server generates the user's public key based on its own public parameters and the user's ID information, generates the private key in combination with the key server's master key, and sends the private key to the user. When the user needs to send encrypted information, the user uses the public parameters of the key server and the ID information of the receiving user to generate the public key of the receiving user, and uses the public key to encrypt the information sent. The receiving user uses the private key given by the key server to decrypt the encrypted information after receiving it. Compared with the certificate public key-based key management scheme, the identity private key-based key management scheme greatly improves the management efficiency because it does not require the application and verification of digital certificates [16].

2.2 Certificate-Free Public Key-based Key Management Scheme

The previous text briefly introduced the certificate public key-based and identity private key-based key management schemes. The public key mechanism-based management scheme uses digital certificates issued by a CA to ensure the reliability of users' identity information and keys, but in the process of use, users need to apply for or verify digital certificates with the CA, and too many users will overburden the CA [18]. The CA stores the public keys of all users, which is cumbersome to maintain. The identity private key-based management scheme uses user ID information as the public key, and the key management server only stores and distributes the private key, so users do not need to submit digital certificates to the server when encrypting information, which improves the efficiency of key management. However, the scheme also has shortcomings, one of which is the escrow of the key. The server uses the user ID to generate the public key and then generates the private key, and once the server is compromised, so is the user's private information. In addition, the key needs to be updated periodically to ensure its security, and the way this scheme generates the public and private keys makes it difficult to update the key [4].

In order to solve the shortcomings of the above two schemes, this paper adopts the certificateless public key mechanism to manage the keys of IoT communication protocols. The new management scheme does not require digital certificates and makes improvements to the key escrow problem in the identity private key-based management scheme. Figure 1 shows the basic steps of the certificateless public key-based key management scheme. The improvement can be briefly summarized according to the flow of steps in Figure 1, that the private key generator no longer generates the public key and the complete private key, but only the partial private key, and the public key and the complete private key are generated by the user [11]. The specific steps are described below.

1) Parameter initialization of private key generator (PKG): Public system parameter $param$: $\{a, b, g, x, y, H_1, H_2\}$ is generated, where a and b are any large prime number satisfying $b|a-1$, g is a generating element in finite multiplication group Z_a^* of a, b is the order of g, x is a value randomly selected from finite multiplication group Z_b^* of b, which is the system private key, y is the remainder of g^x divided by b, which is the system public key, and H_1 and H_2 are the hash functions.

2) The partial private key is generated after combining the public parameters with the user ID in the PKG, and the corresponding formula is:

$$\begin{cases} h_A = H_1(ID_A) \\ D_A = h_A^* (\text{mod}) b \end{cases} \tag{1}$$

where h_A is the value obtained after calculating ID_A of user A by H_1 and D_A is part of the private key of user A.

3) The partial private key is sent to the user. Then, the user generates the secret value using the system parameters and combines the partial private key to generate the complete private key. The correspond-

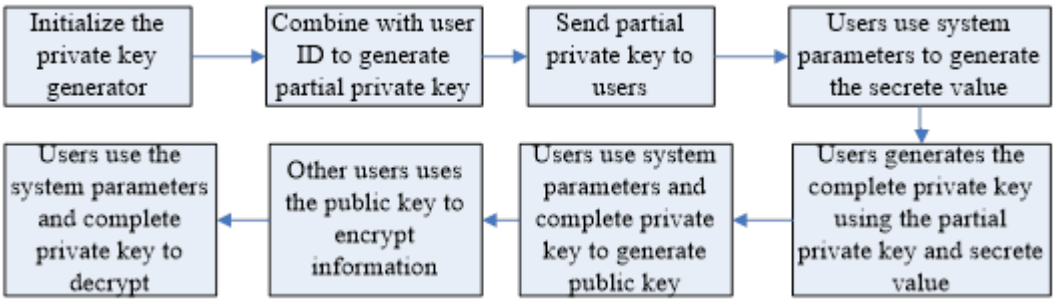


Figure 1: Flow of the certificateless public key management scheme

ing formulas are:

$$\begin{cases} \mu_A \in Z_b^* \\ S_A = \mu_A D_A \end{cases} \tag{2}$$

where μ_A is the secret value of user A, which is randomly selected from Z_b^*, and S_A is the complete private key of user A.

4) User A uses the system parameters and the complete private key to generate public key P_A, and the formula is:

$$P_A = g^{S_A} \bmod b. \tag{3}$$

5) User A's public key is published in the IOT or transmitted to other users who want to "handshake" or communicate. The other users obtain the public key of user A and send encrypted messages to it. The encryption formulas are:

$$\begin{cases} n \in Z_b^* \\ N = g^n \bmod b \\ M = m \oplus H_2((P_A^{H_1^x(ID_A)})^n) \\ C = (N, M) \end{cases} \tag{4}$$

where n is a value randomly selected from Z_b^*, N is part of the ciphertext, M is part of the ciphertext, m is the plaintext, and C is the ciphertext.

6) User A receives ciphertext C and decrypts it using the system parameters and the complete private key. The calculation formula is:

$$m = M \oplus H_2(N^{S_A}). \tag{5}$$

3 Simulation Experiments

3.1 Experimental Environment

The simulation experiments were conducted in three servers in the lab. The server parameters in the lab were quad-core i7 CPU, 32G memory, and 1,024G hard disk. One of the servers served as a third-party digital CA, one served as the server side, and the last served as the user side.

3.2 Experimental Setup

The system parameters required for the key management scheme to generate keys during the simulation experiments included an elliptic curve NIST P-192 with a 192-bit prime number field, the operations of which included addition and scalar-multiplication.

In addition to the key management scheme based on uncertificated public key, the key management scheme based on certificate public key and the key management scheme based on identity private key were also simulated and tested as a comparison. Among them, the handshake protocol constructed using the certificate public key management scheme requires the use of a third-party server as the digital CA, while the handshake protocols constructed by the other two key management schemes did not require a third-party server.

Test Item 1: Communication overhead
The three key management schemes were used to construct the respective handshake protocols. During the testing process, the server and the user performed interactive transmission of messages, and the number of message bytes generated by the handshake protocols under the three key management schemes were captured and compared.

Test Item 2: Time overhead at different packet loss rates
The connection between the server and the user was established using three key management handshake protocols with different packet loss rates, ranging from 0% to 10%. The time taken to establish the connection between the server and the user was recorded from the time the first message was sent. Each key management scheme was repeated ten times for each packet loss rate, and the results were averaged.

Test Item 3: time overhead at different attack frequencies
In the experiment, a fourth server was added as a third-party attacker, and the new server was configured in the same way as the other three servers. The attack frequency was set from 0 to 1 in intervals of 0.1. When the attack frequency was 0, the attack was

unsuccessful, and when the attack frequency was 1, the attack must be successful. For each attack frequency, the server and the user performed handshake interactions using each of the three key management schemes for 500 times, and the average time required for the handshake interactions was calculated.

3.3 Experimental Results

The server side and the user side adopted three key management schemes for the handshake connection respectively, and the communication overheads during the handshake process are shown in Table 1. During the handshake connection under the key management scheme based on certificate public key, the server and the client interacted six times and exchanged 16 messages with a total of 5,800 bytes; during the handshake connection under the scheme based on identity private key, the server and the client interacted five times and exchanged eight messages with a total of 2,630 bytes; during the handshake connection under the scheme based on certificateless public key, the server and the client interacted five times and exchanged eight messages with a total of 2,440 bytes. It can be seen that the number of interactions and the number of messages passed by the identity private key-based scheme and the certificateless public key-based schemes were smaller, and the number of bytes in the handshake messages of the scheme based on the certificateless public key was the smallest. There was no need to send and verify digital certificates in the identity-based private key and certificateless public key schemes, so the number of interactions and handshake messages were smaller.

The time overheads, i.e., the average handshake time, for different packet loss rates when the server and the user used the three key management schemes for handshake connection are shown in Figure 2. It can be seen from Figure 2 that as the packet loss rate increased during data transmission, the handshake connection time of all the three key management schemes increased, and under the same packet loss rate, the certificate public key-based scheme consumed the most time for handshake connection, followed by the identity private key-based scheme, and the certificateless public key-based scheme consumed the least time.

Figure 3 shows the time overhead of the three key management schemes at different attack frequencies. It can be seen from Figure 3 that as the frequency of third-party attacks increased, the average time spent on handshake interaction increased for all three key management schemes; the certificate public key-based management scheme increased the most, followed by the identity private key-based scheme, and the certificateless public key-based scheme increased the least. Under the same attack frequency, the average time consumption of the certificate public key-based management scheme was also the highest, the identity private key-based scheme was the second highest, and the certificateless public key-based scheme was the lowest.

4 Conclusion

This paper briefly introduced two key management schemes based on certificate public key and identity private key, and the key management scheme based on certificateless public key evolved from identity private key in IoT communication protocols. Then, the communication overhead and time overhead of the three key management schemes were tested in the simulation experiments. The results were summarized as follows. The key management schemes based on identity private key and certificateless public key required fewer interactions and exchanged less message information, and the handshake information of the certificate-free public key-based scheme contained the least number of bytes. As the packet loss rate increased, the handshake connection time increased for all three key management schemes. Under the same packet loss rate, the certificate public key-based key management scheme consumed the most time for handshake connection, followed by the identity private key-based scheme, and the certificateless public key-based scheme consumed the least time for handshake connection. As the frequency of third-party attacks increased, the average handshake time of all three key management schemes increased. For the same attack frequency, the certificate public key-based scheme had the highest average time consumption, the identity private key-based scheme had the second highest, and the certificateless public key-based scheme had the lowest.

References

[1] J. W. Lo, C. C. Lee, M. S. Hwang, Y. P. Chu, "A secure and efficient ECC-based AKA protocol for wireless mobile communications", *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 11, pp. 5249-5258, 2010.

[2] S. M. Chen, C. R. Yang, M. S. Hwang, "Using a new structure in group key management for pay-TV", *International Journal of Network Security*, vol. 19, no. 1, pp. 112-117, 2017.

[3] A. Guermazi, A. Belghith, M. Abid, S. Gannouni, "KMMR: An efficient and scalable key management protocol to secure multi-hop communications in large scale wireless sensor networks," *KSII Transactions on Internet & Information Systems*, vol. 11, no. 2, pp. 901-923, 2017.

[4] S. Gupta, V. Gupta, "Analytical modeling of RLC protocol of LTE using stochastic reward nets," *International Journal of Communication Systems*, vol. 32, no. 6, pp. e3903.1-e3903.18, 2019.

[5] J. Huang, Y. Qian, "A secure and efficient handover authentication and key management protocol for 5G networks," *Journal of Communication and Information Networks*, vol. 5, no. 1, pp. 40-49, 2020.

[6] L. C. Huang, M. S. Hwang, "Two-party authenticated multiple-key agreement based on elliptic curve discrete logarithm problem", *International Journal of Smart Home*, vol. 7, no. 1, pp. 9-18, 2013.

Table 1: Communication overhead under the three key management schemes

Key management scheme	Number of interactions/n	Number of handshake messages/n	Number of handshake message bytes/bit
Certificate public key	6	16	5,800
Identity private key	5	8	2,630
Certificateless public key	5	8	2,440

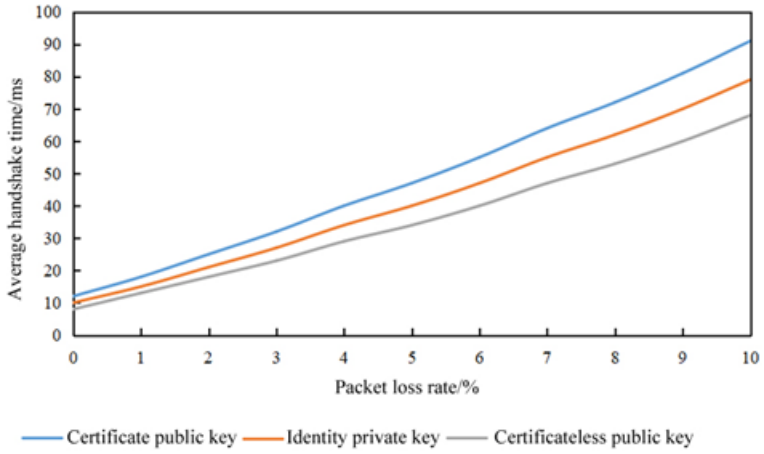


Figure 2: Time overhead of the three key management schemes under different packet loss rates under

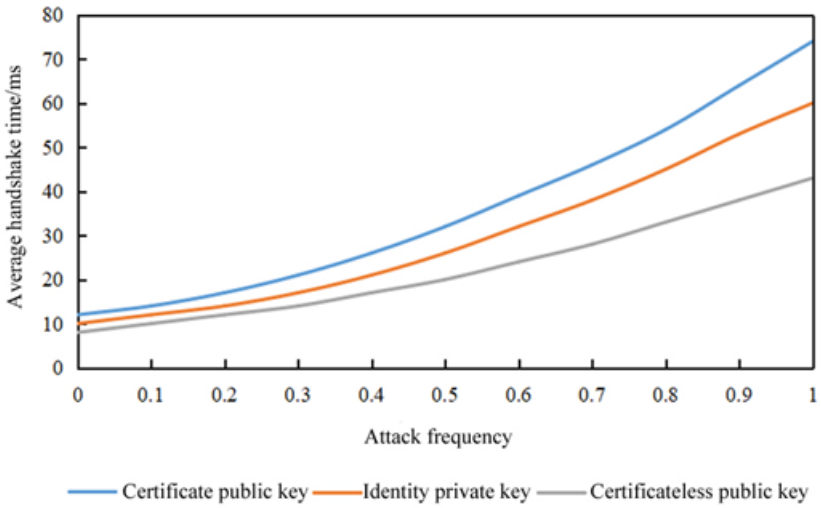


Figure 3: Time overhead of three key management schemes in the face of different attack frequencies

[7] M. S. Hwang, C. C. Lee, S. K. Chong, J. W. Lo, "A key management for wireless communications", *International Journal of Innovative Computing, Information and Control*, vol. 4, 2008.

[8] M. S. Hwang, C. C. Lee, J. Z. Lee, C. C. Yang, "A secure protocol for bluetooth piconets using elliptic curve cryptography", *Telecommunication Systems*, vol. 29, no. 3, pp. 165-180, 2005.

[9] M. A. Kandi, H. Lakhlef, A. Bouabdallah, Y. Challal, "A versatile key management protocol for secure group and device-to-device communication in the internet of things," *Journal of Network and Computer Applications*, vol. 150, no. Jan., pp. 102480.1-102480.17, 2020.

[10] S. Kumar, "Security and Privacy enforced wireless mobile communication using PI-MAKA protocol design," *Measurement and Control: Journal of the Institute of Measurement and Control*, vol. 52, no. 7/8, pp. 788-793, 2019.

[11] T. S. Kumar, S. Prabakaran, "Security and privacy enforced wireless mobile communication using PI-MAKA protocol design," *Measurement and Control*, vol. 52, no. 7/8, pp. 788-793, 2019.

[12] C. T. Li, M. S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks",*Information Sciences*, vol. 181, no. 23, pp. 5333-5347, 2011.

[13] C. T. Li, M. S. Hwang and Y. P. Chu, "An efficient sensor-to-sensor authenticated path-key establishment scheme for secure communications in wireless sensor networks", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2107-2124, Aug. 2009.

[14] I. C. Lin, H. H. Ou, M. S. Hwang, "Efficient access control and key management schemes for mobile agents", *Computer Standards & Interfaces*, vol. 26, no. 5, pp. 423-433, 2004.

[15] C. H. Ling, C. C. Lee, C. C. Yang, and M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN", *International Journal of Network Security*, vol. 19, no. 2, pp. 177-181, Mar. 2017.

[16] L. Lei, W. Zhang, Y. Wang, X. Wang, "A pairing-free identity-based handover AKE protocol with anonymity in the heterogeneous wireless networks," *International Journal of Communication Systems*, vol. 32, no. 12, pp. 1-16, 2019.

[17] I. C. Lin, M. S. Hwang, C. C. Chang, "A new key assignment scheme for enforcing complicated access control policies in hierarchy", *Future Generation Computer Systems*, vol. 19, no. 4, pp. 457-462, 2003.

[18] D. Mishra, S. Rana, "Authenticated content distribution framework for digital rights management systems with smart card revocation," *International Journal of Communication Systems*, vol. 33, no. 9, pp. e4388.1-e4388.19, 2020.

[19] Y. B. Slimane, K. B. Ahmed, Y. Benslimane, "Efficient end-to-end secure key management protocol for internet of things," *International Journal of Electrical and Computer Engineering*, vol. 7, no. 6, pp. 3622-3631, 2017.

[20] T. H. Sun and M. S. Hwang, "A hierarchical data access and key management in cloud computing", *ICIC Express Letters*, vol. 6, no.2, pp. 569-574, 2012.

[21] X. Tong, J. Liu, Z. Wang, M. Zhang, J. Ma, "Threshold-based key management scheme for space network," *International Journal of Communication Systems*, vol. 34, no. 11, pp. e4841.1-e4841.22, 2021.

[22] J. L. Tsai, N. W. Lo, "A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card," *International Journal of Communication Systems*, vol. 28, no. 13, pp. 1955-1963, 2015.

[23] T. Umer, M. H. Rehmani, Z. G. Ding, B. S. Kim, S. U. Khan, "IEEE access special section editorial: Resource management in vehicular adhoc networks: Energy management, communication protocol and future applications," *IEEE Access*, vol. 5, pp. 7839-7842, 2017.

[24] G. Xu, X. B. Chen, Z. Dou, Y. X. Yang, Z. Li, "A novel protocol for multiparty quantum key management," *Quantum Information Processing*, vol. 14, no. 8, pp. 2959-2980, 2015.

[25] C. C. Yang, T. Y. Chang, M. S. Hwang, "A new anonymous conference key distribution system based on the elliptic curve discrete logarithm problem," *Computer Standards & Interfaces*, vol. 25, no. 2, pp. 141-145, 2003.

[26] W. Zafar, B. M. Khan, "A reliable, delay bounded and less complex communication protocol for multicluster FANETs," *Digital Communications & Networks*, vol. 3, no. 1, pp. 30-38, 2017.

[27] G. Zhai, L. Tian, Y. Zhou, Q. Sun, J. Shi, "A computing resource adjustment mechanism for communication protocol processing in centralized radio access networks," *China Communications*, vol. 13, no. 12, pp. 79-89, 2016.

Biography

Hongbo Yao graduated from Tangshan University in 1993. Master degree is achieved at e-Media Engineering from Group T - Leuven Engineering School Belgium in 2003. Now he is a senior experimentalist at Thangshan University and his current research scope includes IOT and information security techniques.

Li Yang was born in Tangshan, Hebei., P.R. China, in 1979. He is currently an associate professor at Tangshan University. He received his master's degree in Software Engineering from Yanshan University in 2011. His current research interests include artificial intelligence and big data.

Centered-Ranking Learning Against Adversarial Attacks in Neural Networks

Benjamin Appiah[1], Adolph S. Y. Adu[1], Isaac Osei[1], Gabriel Assamah[2], and Ebenezer N. A. Hammond[3]

(Corresponding author: Benjamin Appiah)

Department of Computer Science, Ho Technical University, Ho, Volta Region, Ghana[1]
University of Cape Coast, Cape Coast, Central Region, Ghana[2]
Building & Road Research Institute, Kumasi, Ghana[3]
Email: bappiah@htu.edu.gh

(Received Jan. 12, 2023; Revised and Accepted Aug. 1, 2023; First Online Aug. 25, 2023)

Abstract

In this paper, we present a technique for defending against adversarial attacks in neural networks. Our approach regularizer the neural network's representation space under adversarial attacks with Centered Ranking loss to enable a neural network classifier to learn a feature representation that detects similarities between adversarial and clean samples and brings similar samples close to their original class and pushes dissimilar images away from their false classes. We propose a single Projected Gradient Descent for adversarial attack sample generation during training. Training neural network classifier with our Centered Ranking regularizer combined with the single Projected Gradient Descent noise produces a high detection rate in non-adversarial and adversarial settings compared to the state-of-the-art defense methods.

Keywords: Adversarial Detection; Adversarial Training; Anomaly Detection; Metric Learning; Neural Network

1 Introduction

Initial assumptions were that adversarial attacks designed to fool neural networks would not be effective in the real world. However, Kurakin *et al.* (2017), [16], first address this challenge by using the expectation of the model gradients with respect to the inputs plus random noise. Eykholt *et al.* (2018), [6], further consider the masks and fabrication errors, and implemented adversarial perturbations on traffic signs. Recently, [3], successfully generated adversarial samples to deceive the LiDAR-based detection system, thus validating the existence of real world adversarial samples again. These discoveries have made the application of neural networks in the field of critical decision making systems a great concern. Though the problem of adversarial attacks detection has already been discussed for many years [5, 10, 19, 20, 25, 34, 36] there is a growing need to increasing the robustness of neural network, in the face of increasingly variable adversarial attacks. In this paper, we present Centered Ranking learning method against adversarial attacks.

- Our Centered Ranking learning projects the representation space of neural networks under adversarial attacks into an Euclidean space where distance can be directly used to measure the similarity of genuine samples and adversarial sample and bring similar samples close to their original class and push dissimilar samples away from their false classes.
- We further propose a single Projected Gradient Descent method for adversarial training to achieve a more eneralizable learning model. The experimental results show that our methods can achieve promising results and outperforms several state-of-the-art approaches.

The rest of this paper is organized as follows: In Section 2, we review the related works on adversarial attack detection and prevention. Our methodology is presented in Section 3 and Section 4. The experiments are presented in Section 5 and we present our discussion and conclusions in Section 6.

2 Related Work

Notations. We consider a classification task with data $x \in D$ and class labels $y \in Z_k$. We identify a model with a hypothesis h from a space H on input samples x, the model outputs class $h(x) \in R^k$. The loss function $L(.)$ is used to train the model $L((h(x), y); \theta)$, where θ is the network parameter to learn.

2.1 Adversarial Attacks

Given the adversarial sample $x^{adv} = x + \triangle x$, generated by adding tiny perturbations $\triangle x$ to x, the adversarial

objective is to forces h to falsely label malicious samples such that $h(x^{adv}) \neq y$ and the amount of perturbation is maximized ($max||x^{adv}-x||_p$, $s.t\ h(x^{adv}) \neq y$), where $||.||_p$ is the L_p ($i.e\ L_1,\ L_2\ or\ L_\infty$) norm defining the amount of perturbation.

Szegedy *et al.* [30], generated small perturbations on images for the image classification problem and fooled state of the-art neural network classifiers with high probability. Goodfellow *et al.* [7], proposed the Fast Gradient Sign method (FGSM) and also proposed a defense mechanism by training neural network model on the FGSM adversarial examples. Other effective sarial attacks includes the Iterative method (PGD) [6], C&W [4], Basic Iteration Method (BIM) [1], Jacobian based Saliency Map Attack (JSMA) [22] and DeepFool [2] which are proposed to fool deep neural network.

2.2 Defense Methods

Kannan *et al.* (2018), [10] proposed Adversarial Logit Pairing (ALP). The ALP method matches the logits from clean image x and it's corresponding adversarial image x_{adv} and provide an extra regularization term for better representation of the data. However, the loss function adopted in this method is not scalable to untarget adversarial attacks [19]. Madry *et al.* (2019), [19], demonstrated successful defense by training the model on projected gradient descent (PGD) attacks (a.k.a Adversarial Training) which randomly initialize adversarial examples with the allowed norm ball before running iterative attack. Adversarial Training tend to learning model parameters θ^* to maximize the average minimal perturbation distance $\theta^* = argmax_{\theta \in \Theta} E_{x \sim D}\ min_{h(x^{adv} \neq y)}\ ||x^{adv} - x||$.

Regularization techniques have also been introduced to counter the treat of adversarial attacks [17, 20, 33, 35]. These techniques integrates the Triplet loss term [9,15,27] to the original neural network model's Softmax Cross Entropy loss function. The goal of Ranking learning is to learn a function $h_\theta(x) : R^H \longrightarrow R^D$ which maps semantically similar points from the data manifold in R H onto metrically close points in R^D. Analogously, h_θ should map semantically different points in R^H onto metrically distant points in R^D. During training the Ranking loss approaches also adopt the Adversarial Training technique. Our approach combines the Adversarial Training and Ranking learning concepts, however, our Ranking loss avoids the complexity of constructing triplets and the necessity of mining hard samples, a major limitation in the Ranking learning process, furthermore, we propose a single Projected Gradient Descent (OPGD) to generate adversarial samples in the adversarial training stage, unlike in the Adversarial Training [19] that adopts projected gradient descent (PGD). The PGD method requires a large number of projections, leading to a high computational cost per iteration and consequently making standard Adversarial Training technique unappealing for large datasets. The OPGD method does not need intermediate projections, instead, only one projection at the last iteration is needed.

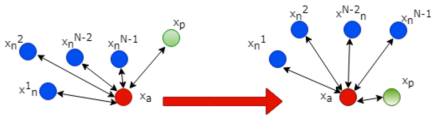


Figure 1: is an illustration of the Ranking loss with (N-1) triplets. The anchor examples (red) and positive examples (green) belong to the same class. The negative examples (blue), from a different class, is the closest image to the anchor in feature space.

3 Centered Ranking Learning

The Ranking loss consist of 3 batch input samples: an anchor (x_a), a positive (x_p), and ($N-1$) negative samples (x_n). Samples x_a and x_p are from the same class and x_a and (x_n) are from dissimilar classes. Given the batch inputs $\{x_a, x_p, x_n\}$, the objective of Ranking loss is to push away the negative point x_n from the anchor x_a by a distance margin $\alpha > \theta$ compared to the positive x_p:

$$||h(x_a) - h(x_p)||^2 + \alpha \leqslant ||h(x_a) - h(x_n)||^2 \quad (1)$$

where $x \in D$ is the input and α is a hyper-parameter for margin. A simple representation of Ranking loss function is defined as:

$$L_{RL} = ||h(x_a) - h(x_p)||_2^2 + \alpha - ||h(x_a) - h(x_n)||_2^2. \quad (2)$$

A backward propagation on Ranking loss actually assign a smaller distance between similar class samples anchor (red dot) and positive (green dot) and assign a wider distance between dissimilar anchor and negative (blue dots) as shown in Figure 1.

General issues with the Ranking loss function are that it is sensitive to selection of anchor point making it harder to train, this means improper anchors selection can result in great instability in the training stage and leads to low convergence, and the selection of an effective anchor point is still an open problem. However, to ensure a stable convergence in the training stage, we propose to replace the anchor with rather a more stable measure, the mean of the positive samples. We expect the samples belonging to same identity should locate around a common center point feature space rather than an unstable anchor. These positive samples should cling together and the samples in the negative set should move far away from the center as shown in Figure 2.

Centered Ranking loss: Our proposed Centered Ranking loss first define a center point by computing the mean of all positive samples

$$C = \frac{1}{N_p} \sum^{N_p} h(x_p) \quad (3)$$

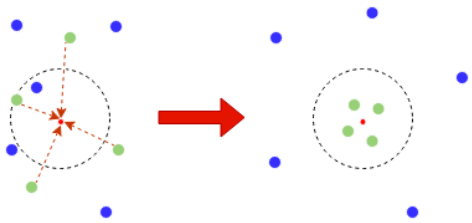


Figure 2: Illustrates our Centered Ranking loss. The Centered Ranking loss learns to pull positive examples from the true class closer towards the center (red dot), and push the (N-1) negative examples of false classes apart based on their similarity to the center.

The goal of the Centered Ranking loss is to push away the negative point x_n from the center C by a distance margin $\alpha > \theta$ compared to the positive x_p:

$$||h(x_p) - C||^2 + \alpha \leqslant ||h(x_n) - C||^2. \tag{4}$$

The Centered Ranking loss is define as

$$L_{CN_p} = \frac{1}{N}\sum_{i=1}^{N} log(1 + \sum_{j\neq 1} exp(||h(x_p) - C||_2^2 + \alpha \leqslant ||h(x_n) - C||_2^2)) \tag{5}$$

The distance between C and x in the representation space is defined using the Euclidean distance. Equation (1) is similar to Equation (4), smaller intra-class distance and wider inter-class distance. However, we measure the distance between clusters center rather than an extreme random anchor selection process. The gradients of L_{CN_p} with respect to x_p and x_n are present in Equations (6) and (7) respectively.

$$\frac{\partial L_{CN_p}}{\partial x_p} = 2(h(x_p) - C) \tag{6}$$

$$\frac{\partial L_{CN_p}}{\partial x_n} = 2(h(x_n) - C). \tag{7}$$

4 Adversarial Training with OPGD

Our robust training aim to replace the input points by their corresponding adversarial perturbations and train the network on the perturbed input all under a single projection. Given a training data x_t, perturbed inputs $\triangle x$ are introduced into the classification model, generated through our Single Projected Gradient Descent method (OPGD)

$$x_i^{adv} = x_t - \varepsilon\, sign\, \rho_t\, (\bigtriangledown h(x_t) + \lambda_t \bigtriangledown g(x_t)) \tag{8}$$

where $\bigtriangledown h$ represents the gradient of h, ε is the perturbation size, the parameter $\rho \geq 0$ is the step size, λ corresponds to Lagrangian multiplier, t is the iteration counter and $g(x) \leq 0$ is a constraint function. The key difference between Equation (8) and the projected gradient descent method in Adversarial Training [19] is the replacement of the projection step with the gradient computation of the constraint function defining the domain D. The computation of the gradient function is cheaper than projection step. Rank Learning and Projected Gradient Descents are thoroughly studied in the neural network adversarial learning literature and we do not describe them in any detail; readers are directed to [9, 15, 16, 19, 21, 27, 33].

We train the model on these augmented samples ($x^{aug} = x + x_{adv}$) under the joint supervision of Softmax Cross-entropy loss (L_{SCE}) and Centered Ranking loss (L_{CN_p}). The total training objective is presented as

$$L_{all} = L_{SCE}(x^{aug}, y; \theta) + \alpha L_{CN_p}(x^{aug}, y; \theta) + \sigma L_{norm}, \tag{9}$$

here α controls the strength of the training stability, $f(\cdot)$ is the output of the last fully connected layer of the network and σ is the weight for the feature norm decay term, to reduce the L_2 norm of the feature. C center is the center of the positive examples x_p and negative examples x_n is from mini-batch which has different label or from different class to the positive x_p and θ the parameters of the model to be optimized. In this settings, we inject the adversarial noise into the training batch and chose the center of the positives to serve as the decision boundary between the "true" class and the "false" class. Using the squared Euclidean distance function below,

$$d(h(x_n), C) = \frac{1}{2}||h(x_n) - C||_2^2 \tag{10}$$

We select negative examples as the nearest sample to the center from a false class. As a result, our model is able to learn to enlarge the boundary between the adversarial samples and their closest negative samples from the other classes. Figure 2, shows our Centered Ranking loss for adversarial training. The CN_P supervised by softmax center loss are trainable and can be optimized by stochastic gradient decent. Algorithm 1, summarizes our adversarial training with Centered Ranking loss with single Projected Gradient Descent.

Algorithm 1 Training with Centered Ranking learning loss

1: **Input** training data x; training iteration T_t; learning rate lr; a sequence of step sizes $\rho; \lambda; \theta; \gamma; t$ mini-batch K for each iteration $X^t_{k\in 1,..,K}$.
2: **for** $t = 1 : T_t$ **do**
3: Generate adversarial samples $x^{adv} = x_t - \varepsilon\, sign\, \rho_t\, (\bigtriangledown h(x_t) + \lambda_t \bigtriangledown g(x_t))$.
4: Generate training samples $a^{aug} = x + x^{adv}$.
5: Sample the positives a_p^{aug} from a^{aug}.
6: Compute the mean (C) of x_p^{aug}.
7: Sample the negatives a_n^{aug} from a^{aug} based on $d(h(x_n), C) = \frac{1}{2}||h(x_n) - C||_2^2$.
8: Compute L_{all}.
9: Update θ.
10: **end for**

5 Experiments

We analyze the effect of the CNP method on CIFAR-10 and MNIST datasets. The MNIST dataset consist of handwritten digits which 60,000 images are training set and 10,000 images a test set. CIFAR-10 is a collection of 60,000 color images in 10 classes having 6,000 images per class, 50,000 of these images are allocated for training and 10,000 for testing. We scaled the pixel values of images in both datasets to be in the range of [0,1] by dividing by the maximum pixel value of /255.

We conduct our experiments using TensorFlow on a Windows PC with Intel Core i7-2600 and a 16GB memory. On MNIST dataset we use a network consisting of two convolutional layers with {32, 64} filters respectively, each followed by 2×2 max-pooling, Batch Normalization, Dropout with sizes {0.25} and fully connected layer of size 1024. For CIFAR-10 and CIFAR-100, we use a network consisting of five convolutional layers with {32, 64, 64, 128, 256} filters, each followed by 3×3 max-pooling, Batch Normalization, Dropout with sizes {0.25}, and fully connected layer of size 1024. We used a grid search on a subspace of the hyper-parameters to select the ones which result in the best performance. In the Ranking loss experi ment, we pretrain the model using Softmax Cross Entropy loss followed by fine-tuning it with the Ranking loss. We apply hard mining strategy [26] to construct triplets in the Ranking loss method. The best value found for the hyper-parameter are $\alpha = 0.2$ and $\sigma = 0.01$.

5.1 Evaluation

To illustrate the advantage of the effectiveness of our proposed method, we compare our approach to Adversarial Training (**At**) method trained under the supervision of Softmax Cross Entropy and Ranking Learning with the Adversarial learning technique (**Np**). We represent our proposed methods in Sections 3 and 4 as (**CNp**). Since generation of attack samples are more expensive. Therefore, the last 1000 MNIST digits and CIFAR10 images are selected in our experiments. For At and Np methods, we generated adversarial examples using the same method introduced by [19]. Given a clean sample x, we generate the adversarial samples $x^{adv} = P_D(x_i - \varepsilon sign\rho \bigtriangledown F(x_i), y)$. We do not add the Ranking loss or the Centered Ranking loss term into the loss of adversarial example generation because it causes non-convergence. We evaluate performances under different adversarial attacks and under different adversarial settings whiles considering the effects against untargeted attack scenarios with different combinations of the attacking parameters through the Detection Rate DR:

$$DR = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

where TP, TN, FN, and FP denote the number of correctly detected adversarial samples (true positive), the number of correctly detected normal samples (true negative), the number of adversarial samples that are detected as normal ones (false negative), and the number of normal samples that are detected as adversarial ones (false positive), respectively.

5.1.1 Detecting FGSM and PGD Attacks

The FGSM Adversarial samples are crafted with ϵ values from 0.1 to 0.3 on MNIST and from 4/255 to 8/255 on CIFAR-10. Table 1, shows that CNp achieves an average detection rate of 99.10% on MNIST and 59.16% on CIFAR-10. Figure 3(a)(c), shows the test error rate curves w.r.t. the training time. We can see that the CNp loss induces faster convergence rate compared to the At and Np loss. We craft the Projected Gradient Descent (PGD) attack samples using the following setting: iteration size = 50 and ϵ from 0.1 to 0.3 for MNIST and 4/255 to 8/255 for CIFAR-10. Table 1, shows that CNp achieves an average detection rate of 97.49% on MNIST and 55.16% on CIFAR-10. Figure 3(b)(d), shows the test error rate curves w.r.t. the training time. Note that the CNp loss induces faster convergence rate compared to the At and Np loss.

5.1.2 Detecting Other Attacks

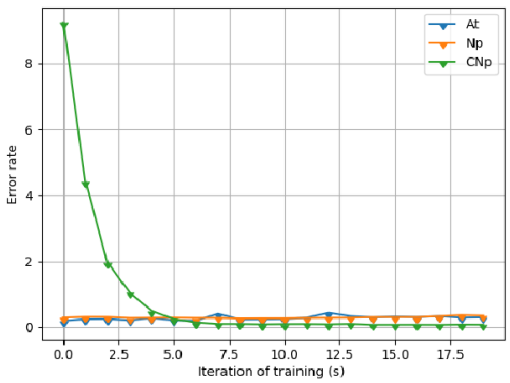
We also evaluated the detection ability of our model by evaluating it on other attacks such as C&W, JSMA, BIM and DeepFool. We set the C&W attack strength from 0.1 to 2.0 on MNIST and CIFAR-10 datasets. We craft JSMA attacks with perturbation (γ) from 0.1 to 0.3. The BIM adversarial samples are crafted from attack range β 0.1 to 0.3 on MNIST and 4/255 to 8/255 on CIFAR-10. Finally, the DeepFool attacks with a (δ) settings from 0.01 to 0.03. We can see from Table 1 that CNp slightly increase performance with an average detection rate of 96.60% on MNIST and 53.21% on CIFAR-10 against C&W attack. Similar performances are recorded on the MNIST and CIFAR-10 against JSMA and DeepFool attacks. Compared to Regularization loss approach (Np) and Adversarial Training (At), CNp method performed lesser against higher distortion rate L_2 attacks (BIM) on MNIST dataset with AT achieving higher detection rate but achieved better performance on complex CIFAR-10 dataset.

5.2 CNp vs Other Detectors

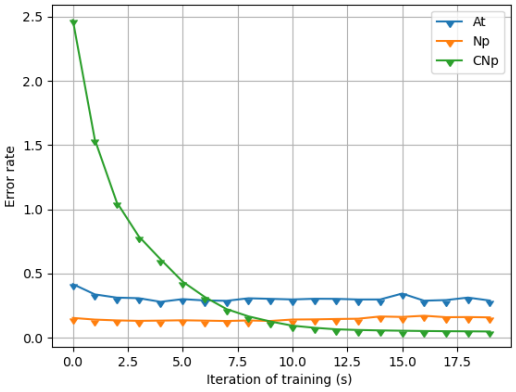
We compare the performance of CNp with methods that has demonstrated their applicability to the task of adversarial robustness in neural networks such as Triplet Loss Adversarial training (TLA) mentioned in [20], Adversarial Logit Pairing (ALP) [10], Label Smoothing and Logit Squeezing method (LLP) proposed by [28], Defence GAN (DGAN) by [26]. The detection results of our method are presented in Table 2. CNp can achieve a high detection rate of 98.62% and 97.84% on MNIST dataset. Compared with the existing methods on CIFAR-10, we can conclude that CNp method can perform considerable well

Table 1: Detection rate of CNp, At, and Np under untargeted attack on MNIST and CIFAR-10 dataset. High scores are indicated in bold. We set lower values for JSMA, C&W and DeepFool attacks since they are more expensive to generate than other attack techniques. Detection Rates are in (%).

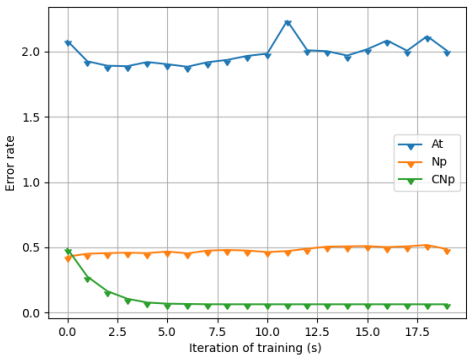
MNIST				CIFAR-10			
Attacks	At	Np	CNp	Attacks	At	Np	CNp
FGSM ($\varepsilon = 0.1$)	98.42	98.83	**99.32**	FGSM ($\varepsilon = 4/255$)	56.31	58.31	**59.02**
FGSM ($\varepsilon = 0.2$)	98.01	98.52	**99.21**	FGSM ($\varepsilon = 6/255$)	56.21	58.01	**58.78**
FGSM ($\varepsilon = 0.3$)	97.74	98.30	**98.76**	FGSM ($\varepsilon = 8/255$)	55.32	57.77	**58.38**
PGD ($\varepsilon = 1$)	96.73	97.11	**98.65**	PGD ($\varepsilon = 4/255$)	50.72	52.81	**55.48**
PGD ($\varepsilon = 2$)	96.51	97.01	**98.42**	PGD ($\varepsilon = 4/255$)	49.81	52.81	**55.21**
PGD ($\varepsilon = 3$)	96.47	96.82	**98.21**	PGD ($\varepsilon = 4/255$)	49.31	52.32	**54.78**
C&W ($k = 0.1$)	95.32	96.47	**96.72**		47.81	50.81	**59.02**
C&W ($k = 0.2$)	95.07	96.16	**96.48**		45.32	49.72	**59.02**
JSMA ($\gamma = 0.1$)	88.30	90.10	**91.36**		45.32	46.18	**48.01**
JSMA ($\gamma = 0.2$)	88.24	**89.81**	89.62		45.17	45.62	**47.93**
JSMA ($\gamma = 0.3$)	88.24	**89.32**	89.23		44.29	45.61	**47.58**
BIM ($\beta = 0.1$)	**97.55**	96.74	97.34	FGSM ($\beta = 4/255$)	49.72	53.65	**54.38**
BIM ($\beta = 0.2$)	**97.31**	96.51	97.01	FGSM ($\beta = 6/255$)	49.18	52.91	**54.17**
BIM ($\beta = 0.3$)	**96.89**	96.03	96.54	FGSM ($\beta = 8/255$)	48.72	52.12	**53.72**
DeepFool ($\delta = 0.01$)	83.21	85.71	**85.76**		47.32	47.52	**48.41**
DeepFool ($\delta = 0.02$)	83.05	**85.31**	85.24		47.28	47.51	**48.24**
DeepFool ($\delta = 0.03$)	84.96	84.96	**85.02**		47.11	47.23	**48.22**



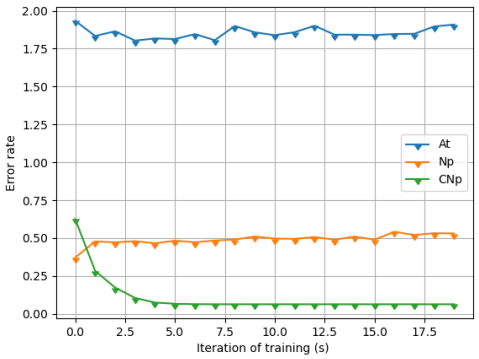
(a) FGSM attack on MNIST



(b) PGD attack on MNIST



(c) FGSM attack on CIFAR10



(d) PGD attack on CIFAR10

Figure 3: Test error rates on MNIST and CIFAR10 datasets w.r.t training time under untarget adversarial attacks. The CNp method results in better performance or at least lower error rate.

compared to baselines in strong iterative attack scenario with an improvement of 2.14%.

Table 2: Comparison to baselines under single step FGSM and 40 steps of PGD attacks on MNIST and Single step FGSM and 20 steps of PGD attacks on CIFAR-10. High scores are indicated in bold. The results for ALP were obtained from [9]. Detection Rates are in (%). Parameter ε is set to 3 and 8 on MNIST and CIFAR-10 respectively.

	MNIST		CIFAR-10	
Methods	FGSM	40PGD	FGSM	20PGD
ALP	97.34	96.62	**60.29**	48.50
TLA	98.17	97.70	58.88	51.59
LLP	94.29	78.21	74.16	49.73
D-GAN	98.38	-	-	-
CNp	**98.62**	**97.84**	58.38	**53.73**

6 Conclusion

In this paper, we proposed a Centered Ranking loss with single Projected Gradient Descent (CNp) for adversarial robustness in neural network. Our proposed method was evaluated on MNIST and CIFAR-10 datasets and under untargeted state-of-the-art adversarial attacks, including projected gradient descent (PGD), C&W, Basic Iteration Method (BIM), Jacobian-based Saliency Map Attack (JSMA), and DeepFool. The Experimental results shows that the combination of Centered Ranking loss Learning with single Projected Gradient Descent leads to high adversarial detection compared to state-of-the-art approaches. In the future, we plan to enhance CNp method by combining it with other neural networks adversarial robustness techniques such as label smoothing. Furthermore, the sensitivity analysis for choosing α to high or too low would be analysed.

References

[1] A. AkhtarZ, A. S. Bedi, K. Rajawat, "Conservative stochastic optimization with expectation constraints," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3190-3205, 2021.

[2] N. Anh, Y. Jason, C. Jeff, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, pp. 427-436, 2015.

[3] Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, "Adversarial objects against LiDAR-based autonomous driving systems," arXiv:1907.05418, 2019.

[4] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, A. Kurakin, "On evaluating adversarial robustness," arXiv preprint arXiv:1902.06705, 2019.

[5] N. Carlini, D. A. Wagner, "Adversarial examples are not easily detected: bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3-14, 2017.

[6] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625-34, 2018.

[7] I. J. Goodfellow, J. Shlens, C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations (ICLR'15)*, 2015.

[8] A. Hermans, L. Beyer, B. Leibe, "In defense of the triplet loss for person re-Identification," arXiv preprint arXiv:1703.07737, 2017.

[9] E. Hoffer, N. Ailon, "Deep metric learning using triplet network," arXiv:1412.6622, 2014.

[10] H. Kannan, A. Kurakin, I. J. Goodfellow, "Adversarial logit pairing," CoRR, abs/1803.06373, 2018.

[11] K. Kawaguchi, "Deep learning without poor local minima," arXiv:1605.07110, 2016.

[12] K.Kawaguchi, "On the theory of implicit deep learning: global convergence with implicit layers," ArXiv, abs/2102.07346, 2021.

[13] K. Kawaguchi, J. Huang, "Gradient descent finds global minima for generalizable deep neural networks of Practical Sizes," *57th Annual Allerton Conference on Communication, Control, and Computing*, pp. 92-99, 2019.

[14] K. Kawaguchi, L.P. Kaelbling, (2020). "Elimination of all bad jocal minima in deep Learning," ArXiv, abs/1901.00279.

[15] S. Kihyuk, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems (NIPS'16)*, vol. 29, pp. 1849-1857, 2016.

[16] S. Kurakin, , A. I. J. Goodfellow, S. Bengio, "Adversarial examples in the physical world," in *5th International Conference on Learning Representations (ICLR'17)*, 2017.

[17] P. Li, J. Yi, B. Zhou, L. Zhang, "Improving the robustness of deep neural networks via adversarial training with triplet Loss," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI'19)*, pp. 2909-2915, 2019.

[18] Q. Lin, R. Ma, Y. Xu, "Inexact proximal-point penalty methods for constrained non-convex optimization," arXiv: Optimization and Control. 2019.

[19] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *6th International Conference on Learning Representations (ICLR'19)*, 2019.

[20] C. Mao, Z. Zhong, J. Yang, C. Vondrick, B. Ray, "Metric learning for adversarial robustness," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 478-489, 2019.

[21] M. Mehrdad, Y. Tianbao, J. Rong, Z. Shenghuo, Y. Jinfeng, "Stochastic gradient descent with only one projection," in *Advances in Neural Information Processing Systems (NIPS'12)*, pp. 503-511, 2012.

[22] N. Papernot, P. D. McDaniel, S.Jha, M.Fredrikson, Z. B. Celik, A. Swami, "The limitations of deep Learning in adversarial settings," in *IEEE European Symposium on Security and Privacy*, pp. 372-387, 2016.

[23] H. Pathak, R. C. Paffenroth, "Non-convex optimization using parameter continuation methods for deep neural networks," *Deep Learning Applications*, vol. 2, pp 273–298, 2020.

[24] S. Pouya, K. Maya, C. Rama, "Defensegan: Protecting classifiers against adversarial attacks using generative models," arXiv preprint arXiv:1805.06605, 2018.

[25] A. Resler, Y. Mansour, "Adversarial online learning with noise," in em Proceedings of the 36th International Conference on Machine Learning, pp. 5429-5437, 2019.

[26] P. Samangouei, M. Kabkab, R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in em International Conference on Learning Representations, 2018.

[27] F. Schroff, D. Kalenichenko, J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Computer Vision and Pattern Recognition (CVPR'15)*, pp. 815-823, 2015.

[28] A. Shafahi, A. Ghiasi, F. Huang, T. Goldstein, "Label smoothing and logit squeezing: A replacement for adversarial training," ArXiv, volume=abs/1910.11585.

[29] R. Sun, "Optimization for deep learning: An overview," *Journal of the Operations Research Society of China*, vol. 8, pp. 249-294, 2020.

[30] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, F. Rob. "Intriguing properties of neural networks," CoRR abs/1312.6199, 2013.

[31] K. K. Thekumparampil, P. Jain, P. Netrapalli, S. Oh, "Projection efficient subgradient method and optimal nonsmooth frank-wolfe method," ArXiv, abs/2010.01848, 2020.

[32] K. Yadav, "A comprehensive ctudy on optimization strategies for gradient descent in deep learning," ArXiv, abs/2101.02397, 2021.

[33] Z. Yaoyao, D. Weihong, "Adversarial learning with margin-based triplet embedding regularization," in *IEEE/CVF International Conference on Computer Vision (ICCV'19)*, pp. 6548-6557, 2019.

[34] Z. Yuchen, L. Percy, "Defending against whitebox adversarial attacks via randomized discretization," in *Proceedings of Machine Learning Research*, vol. 89, pp. 684-693, 2019.

[35] D. Yueqi, L. Jiwen, Wenzhao, Z., Jie, Z. "Deep adversarial metric learning," *IEEE Transactions on Image Process*, vol. 29, pp. 2037-2051, 2020.

[36] V. Zantedeschi, M. I. Nicolae, A. Rawat, "Efficient defenses against adversarial attacks," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec'17)*, 2017.

[37] H. Zhang, H. Chen, Z. Song, D.S. Boning, I. S. Dhillon, C. Hsieh, "The limitations of adversarial training and the blind-spot attack," ArXiv, abs/1901.04684, 2019.

Benjamin A. is currently a lecturer at Ho Technical University. His research interests include machine learning, Security, Deep Learning, Data Mining, and Big Data Analysis.

Adolph S. Y. A. received his M.Sc. degree from Kwame Nkrumah University of Science and Technology, Ghana, He is currently a lecturer at Ho Technical University. His research interests include Machine Learning, Educational Technology, Human Computer Interaction and Learning Analytics.

Isaac O. received the M.Sc. degree from Kwame Nkrumah University of Science and Technology, Ghana. His research interests include machine learning, data mining, big data analysis, applied cryptography, blockchain technology, and Network Security.

Gabriel A. is currently a Ph.D. candidate at University of Cape Coast, Ghana. His research interests includes Cryptography, Network Security, data mining, and big data analysis.

Ebenezer N. A. H. received the PhD. degree from University of Electronic Science and Technology of China. He is currently a researcher at Building & Road Research Institute. His research interests include facial age estimation, family online safety, ICT management, and ICT in constructions.

A Feature-Based Network Traffic Classification Approach

Qian Mao[1], Charles O'Neill[2], and Ke Bao[3]
(Corresponding author: Qian Mao)

Mathematics & Computer Science Department, Whitworth University[1]
300 W Hawthorne Rd, Spokane, WA, USA, 99251
EH Group, Inc[2]
618 19th Ave, Tuscaloosa, AL, USA, 35401
Department of Engineering and Engineering Technology, Metropolitan State University of Denver[3]
890 Auraria Pkwy, Denver, CO, USA, 80204
Email: qmao@whitworth.edu

(Received Nov. 19, 2022; Revised and Accepted Aug. 1, 2023; First Online Aug. 25, 2023)

Abstract

A novel traffic classification approach using deep learning was proposed in this paper. Most current deep-learning-based traffic classification models use raw traffic data as the input of the neural networks. However, the amount of raw data is overwhelming. Many current models use a subset of the raw traffic data, such as the first hundred bytes of a network traffic flow. This idea limits the size of neural network input but loses some traffic information that helps classify traffic types. Instead of using raw traffic data, in this paper, we use traffic flow features as the neural network input. To efficiently classify traffic types, we analyzed what features play important roles in traffic classification and designed neural networks accordingly. Experimental results show that our approach increases classification precision by 19 percent compared to state-of-the-art methods. Meanwhile, if trained by a dataset labeled normal traffic and intrusion traffic, our models can also be applied to network intrusion detection and cybersecurity.

Keywords: Convolutional Neural Networks (CNN); Deep Learning; Multilayer Perceptron (MLP); Traffic Classification

1 Introduction

Network traffic classification categorizes network communications into various types, such as http, email, streaming, etc., and has been extensively used in Quality-of-Service control, billing, malware detection, etc. Traditional traffic classification techniques use port number to categorize traffic type. However, port-based classification methods only work well for traffic of which the port is publicly known. Nowadays, network traffic is more and more complicated, various, and camouflaged. Some traffic uses self-defined ports, while some malware disguises itself by sneaking through publicly defined ports. Using port number alone to distinguish modern traffic is not accurate anymore. Researchers also inspected network packets and found particular patterns to categorize traffic, known as Data Packet Inspection (DPI). This approach has improved traffic classification accuracy. However, the classification performance of DPI highly depends on what patterns are used to identify a traffic type. Usually, expertise in networking field is needed, and the patterns used to recognize a traffic type are different from one type to another. Furthermore, DPI only works for non-encrypted traffic.

To make the traffic classification more accurate and automated, neural networks, especially deep learning networks, have attracted researchers' interests and achieved promising performance. Theoretically, neural networks can inspect all information in the traffic, including the patterns humans do not see, and decide what traffic type it is. However, there are several challenges in network traffic classification using neural networks. First, neural networks require huge datasets for training. For network traffic classification, traffic data with labels is usually used. A crucial problem is data collection. The location that the training data could be collected includes devices in access networks (PC, laptop, smartphones, routers, etc.), local or regional Internet Service Providers (ISPs), and national or global ISPs. Generally, dataset captured in the ISPs is more diverse since it come from various clients. However, the features of traffic data collected in ISPs might have more deviation and randomness. For example, the interval time between packets might vary due to congestion, and the size of packets might have changed when packets travelled through various network protocols. Those deviation and randomness in traffic flows makes the

classification more challenging. Depending on where the data was collected, various tools could be used. For example, Wireshark can be installed on a host computer and collect internet flow and packet data from and to the host computer. Meanwhile, there are some public databases of traffic flows. For instance, ISCXCPN2016 is a 28GB network flow dataset [3]. The flow data is in pcap format and has 14 types: browsing, email, chat, streaming, file transfer, VoIP, TraP2P, and each type includes non-VPN and VPN traffic. All flow data was captured on local computers and was manually labeled.

Another challenge of traffic classification using neural networks is how to preprocess the data and feed it to a neural network. The network data been collected from internet is usually in raw format, e.g., pcap or pcapng. However, this format is not readable for most neural network programs, such as programs written in python or C++. It is important to transfer raw data of network flow/packet into a readable format for the neural network programs. There are two approaches to achieve this goal. One is to write a program/script to transfer a pcap file to a readable format. For example, Wang at al. have developed the USTC-TL2016 program, which reads raw network data file and generates required format for their neural network models [14]. The other approach is to use the third-party software. For example, the JOY software package, developed by CISCO, captures and analyzes network flow data. It extracts data features from pcap network traffic files and generates a json file [8].

The third challenge of traffic classification is the architecture design of the neural networks. Nowadays, various neural networks have been used for traffic classification, such as Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Stacked Auto Encoder (SAE), etc. [11]. Which neural network should be adopted and how to design the neural network architecture are challenging questions. Since in a network traffic flow, data tends to have temporal correlation, many researchers have chosen CNN models. An end-to-end traffic classification approach was proposed in [13], which read the first N bytes of each traffic flow or session and sent them to a one-dimensional CNN model for traffic classification. By converting the time series data into two-dimensional data structure, Chen et al. used a 2D CNN model to classify traffic type [1]. Wang proposed a traffic identification method using SAE [15]. Those models have achieved improved performance in traffic classification at the cost of relatively high computational complexity.

In this paper, we proposed a novel traffic classification method using flow features. There are two major contributions in our work. First, we analyzed various flow features and compared their impact to the traffic classification. Secondly, since feature selection significantly impacts the structure and complexity of the neural network, we designed neural networks based on the features been selected. These measures improved classification accuracy and decreased computational complexity. Our traffic classification models can also be applied to network intrusion detection. Intrusion detection is a spotlight in the field of cybersecurity [6, 7]. Researchers have proposed various models to detect intrusion, and neural network has been proved to be significantly efficient and accurate [2, 4, 10, 12]. Using a training dataset labeled as normal and malicious, our models can classify traffic as these two types, therefore detecting network intrusions.

The rest of the paper is organized as follows. Section 2 introduces how to collect traffic data and conducts a comprehensive analysis on traffic features. In Section 3, various features are selected for traffic classification, and neural networks are designed accordingly. We also implemented these traffic classification models and compared their performances to the state-of-the-art models. Finally, Section 4 draws the conclusions.

2 Flow Features for Traffic Classification

There are three major procedures in neural-networked-based traffic classification: Data collection, data preprocessing, and neural network design. We will discuss the first two steps in this section and the third one in Section 3. Particularly, we will analyze the network traffic flow features, which will be used as neural network input in Section 3.

2.1 Data Collection

Traffic classification can be implemented upon either flow or packet. A network traffic flow is defined as a sequence of network packets with the same 5-tuple: 1) source IP address, 2) destination IP address, 3) source port number, 4) destination port number, and 5) protocol number. Compared to using packet information to classify traffic, using flow offers higher accuracy for two reasons. First, a flow is directly related with a traffic type while packets are just elements of a flow. Secondly, the inter-packet and cross-packet information could be used for traffic classification, which can only by obtained from a flow.

In addition to the 5-tuple, there is abundant information related to a flow that can also be obtained, such as Transport Layer Security (TLS) information, packet information, bytes distribution, etc. Currently, using either raw traffic data [13,14] or flow/packet features as the neural network input can be found in literature. The size of raw data is overwhelming. To constrain the complexity of the neural network, only part of the flow/packet data can be used, e.g., the first N bytes. To leverage flow information comprehensively, in this paper, we extract features from a flow and use them as the neural network input.

In this paper, the data used for neural network training was provided by ISCXVPN2016 [3]. The format of the network flow is pcap. To extract features from pcap files, the JOY software was used. JOY extracts flow features from pcap network traffic files and writes features into a

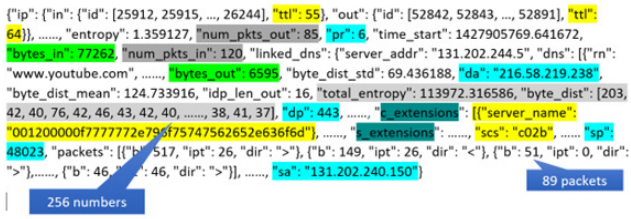


Figure 1: JSON file Generated by JOY Software

json file. Figure 1 shows parts of a json file generated by JOY.

The json file shown in Figure 1 illustrates the flow's source IP address, destination IP address, port number, packets information, etc. The total amount of feature information generated by JOY is tremendous. In the next subsection, we will analyze how to select and use these features for traffic classification.

2.2 Features of Network Traffic Flow

The features extracted by the JOY software can be divided into groups according to their characteristics. In this section, we analyze a subset of groups which will be used by the neural network for traffic classification.

2.2.1 Feature Group A: Basic Flow Information

Since a flow is defined as a sequence of packets with the same 5-tuple, the basic flow information includes the five tuples, i.e., source IP (sa), destination IP (da), source port number (sp), destination port number (dp), and protocol number (pr). The IP address is composed of four eight-bit integers for IPv4 or eight 16-bit integers for IPv6. To make it easier to be read by a neural network, we divide an IP address into four (or eight) integers, and each integer will be sent to a neuron of the neural network input layer. Table 1 shows the features we selected for Feature Group A and some example values.

Table 1: Feature Group A (Basic Information) and Example Values

Source IP (sa)	10.8.8.138
Destination IP (da)	131.202.244.3
Source Port No.	33827
Destination Port No.	443
Protocol No.	6

2.2.2 Feature Group B: Transport Layer Security (TLS) Information

Many network applications, such as email, instant messaging, etc., use TLS to provide a cyber security layer. TLS-related information can be extracted from the json file and sent to neural networks as features for traffic classification. TLS information could include Cipher Suite (CS), server name, key size, etc.

Cipher Suite specifies the cryptographic algorithms and parameters adopted by the communication. It is represented as a 4-digit hex number. For example, in Figure 1, Server CS (SCS) value is c02b, which indicates IANA name TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256, meaning the protocol is *Transport Layer Security (TLS)*, key exchange method is *Elliptic Curve Diffie-Hellman Ephemeral (ECDHE)*, Authentication method is *Elliptic Curve Digital Signature Algorithm (ECDSA)*, Encryption method is *Advanced Encryption Standard with 128bit key in Galois/Counter mode (AES 128 GCM)*, and Hash method is *Secure Hash Algorithm 256 (SHA256)* [9]. The CS solution used for the flow is related to application type, therefore can be used for traffic classification. Table 2 shows the features we selected for Feature Group B and their values as examples.

Table 2: Feature Group B (TLS) and Example Values

Server Cipher Suite (SCS)	c02b
Server Name	001200000f7777772e79 6f75747562652e636f6d
Key Length	528

2.2.3 Feature Group C: Time-to-Live (TTL) Features

TTL is a counter which limits the lifetime of a datagram in the network and prevents the datagram looping in the network indefinitely. It is set by the sender of the datagram and reduced along the route where the datagram is transferred. If a TTL value reaches zero before the datagram reaches the destination, the datagram would be destroyed. According to Internet Protocol [5], TTL is an 8-bit field. Therefore, the maximum value of TTL is 255. Table 3 shows the TTL features read from a json file and their example values, where TTL-out is set to the outgoing flow by the host computer and TTL-in is the value read from the incoming flow. TTL information reveals the distance between the sender and the receiver, which could be used as a factor in traffic classification.

Table 3: Feature Group C (TTL) and Example Values

TTL-out	64
TTL-in	55

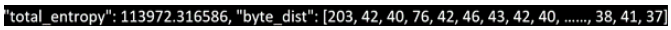


Figure 2: Byte Distribution and Entropy

Figure 3: Packet Sequence Information

2.2.4 Feature Group D: Byte Distribution and Entropy

The json file offers byte distribution, which is an array with 256 elements. Each element in byte distribution represents the number of occurrences of a specific byte value in a flow. For example, Figure 2 shows that there are 203 bytes with value 0, 42 bytes with value 1, 40 bytes with value 2, etc. Meanwhile, the total entropy of all the bytes is also available in json file, as shown in Figure 2. Since different traffic may have different byte distribution patterns, this statistic information reveals the traffic types to some extent.

2.2.5 Feature Group E: Packet Sequence Information

The json file also provides packet sequence information, as shown in Figure 3. For each packet, a 3-element tuple, {b, ipt, dir}, is provided. Parameter "b" represents the number of bytes in the packet, which is 517 bytes in the first packet in Figure 3. Parameter "ipt" is inter-packet time. Parameter "dir" represents the direction of the packet, where '>' means the packet flows from the sender to the receiver, while '<' means the opposite direction.

The number of packets of a flow varies. To make sure that the number of features remains the same for all the flow samples, only the first 50 packets information is used as neural network input. The packet information will be filled with zeros if less than 50 packets are in a flow. Table 4 shows the features we selected for Feature Group E and their example values, which illustrates the number of bytes (b), inter-packet time (ipt), and the direction of the packet (dir) of packet0 through packet 49.

Table 4: Feature Group E (Packet Sequence Information) and Example Values

packet0_b	517
packet0_ipt	26
packet0_dir	>
packet1_b	149
packet1_ipt	26
packet1_dir	<
.....	.....
packet49_dir	>

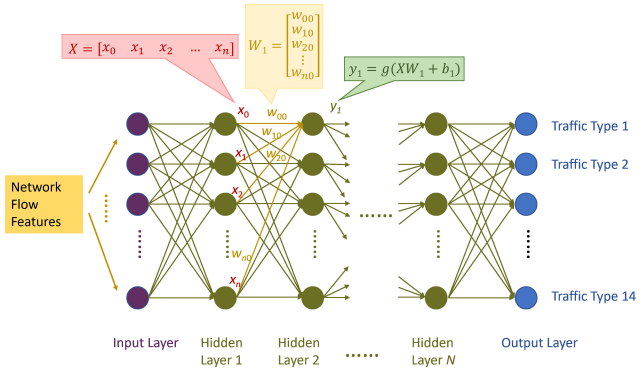


Figure 4: Traffic Classification using MLP

3 Neural Network Design and Performances

To test our traffic classification models, we used ISCXVPN2016 dataset and JOY software to extract flow information from raw traffic data. After removing irrelevant samples, we obtained about 33,000 samples of network flow, containing 14 traffic types: with and without VPN browsing, email, chat, streaming, file transfer, VoIP, and TraP2P. To achieve the classification goal, various neural networks were designed and implemented, including Multilayer Perceptron (MLP) and Convolutional Neural Networks (CNN).

3.1 Traffic Classification using Multilayer Perceptron

Multilayer Perceptron is also called deep feedforward network or feedforward neural network. MLP is a fully connected neural network. In an MLP, information flows from input layer, goes through a number of hidden layers, and reaches the output layer without feedback connections, as shown in Figure 4. For a neural in the network, its output is calculated as

$$y_i = g(XW_i + b_i), \tag{1}$$

where X is an array which contains the output of the previous layer's neurons, W_i is the array of the weights of all the connections between the previous layer's neurons to the current layer's neurons, b_i is a bias of the current layer's neurons, and g is the activation function. Figure 4 shows an example of the calculation of y_1 of the second hidden layer.

MLP is one of the earliest models of neural networks. It is straightforward in mathematics; however, compared to other neural networks, more hidden layers and neurons might be needed for an MLP to extract features layer by layer and to make a correct decision in the output layer. To leverage temporal correlation in flow data, there are few traffic classification models using MLPs. However, when very limited information of a flow is used for traffic

classification, using an MLP with less layers and less neurons can decrease computational complexity significantly.

We used Keras in Linux to implement and test our models. In this model, we use Relu as the activation function, adam optimizer, categorical crossentropy as the loss function, and classification accuracy as metrics. The key problem in neural network design includes 1) what are the inputs of the neural network, and 2) what are the neural network architectures. To answer these questions, we designed three schemes based on the choice of flow features, as shown in Table 5. For each scheme, a neural network architecture was determined after multiple rounds of optimization.

Scheme 1: 11 features. In Scheme 1, we only use features in group A as input of the neural network, i.e., source IP (sa), destination IP (da), source port number (sp), destination port number (dp), and protocol number (pr). For the four integers in an IP address, each of them will be sent to a neuron of the input layer of the MLP. Therefore, there are totally 11 features as input, requiring an input layer with 11 neurons. The simulation results are shown in Table 5. For the MLP architecture in the table, the first number represent the number of inputs, the next number represent the number of neurons in the first hidden layer, etc., and the last number represents the number of neurons in the output layer. Since we are categorizing all the flows into 14 types, the output layer always has 14 neurons.

Scheme 2: 599 features. In Scheme 2, features in groups A, B, and C are used as neural network's input. After using get_dummies function of panda library in Python to convert categorical variable into dummy/indicator variables, the total number of features is 599.

Scheme 3: 937 features. In Scheme 3, all features we mentioned in Section 2 are used as neural network's input, which provides totally 937 features.

To evaluate the classification performance, accuracy and precision are used, which are defined as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2}$$

$$precision = \frac{TP}{TP + FP}, \tag{3}$$

where TP is the number of True Positive, TN is the number of True Negative, FP is the number of False Positive, and FN is the number of False Negative. Table 5 shows the accuracy and precision for each scheme.

From Table 5 we see that, the difference between Scheme 1 and Scheme 2 is trivial, and Scheme 2 requires a more complicated MLP architecture to achieve similar precision. This is because of a larger number of input features of Scheme 2. There is a significant improvement in both accuracy and precision in Scheme 3. That means the basic flow information (IPs and port numbers) are essential for traffic classification, and adding byte distribution and packet sequence information to neural network input can improve the performance significantly.

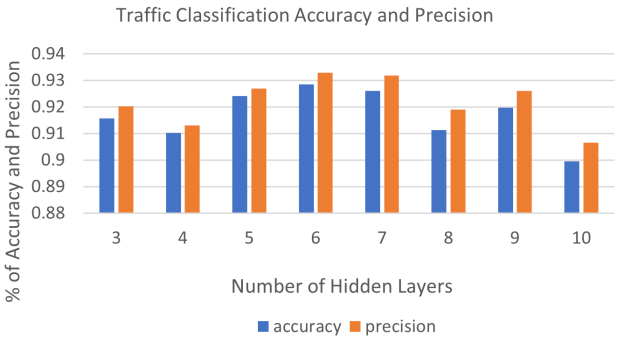


Figure 5: Performance with Various Number of MLP Layers

The accuracy and precision vary with different MLP architectures. Figure 5 shows the classification performance with various numbers of hidden layers, where scheme 3 with 937 input features had been used. The experimental results show that the accuracy and precision are above 90% even with three hidden layers. Among all the models, the MLP networks with six or seven hidden layers yield the optimal performances.

3.2 Traffic Classification using Convolutional Neural Networks

CNN are neural networks that use convolution instead of general matrix multiplication in at least one of the layers. This architecture enables CNN extracting features from a grid-like data topology, therefore performing well in time-series data, image data, etc. To implement a 1D CNN layer, first the input data need to be reshaped into a $n \times P$ array, where $n \times P$ equals to the size of the input data. Then each neuron of the CNN layer has the following output:

$$y_i = g(x * w)(i)) = g\left(\sum_{j,k=1}^{S,P} x_{i+j,k} w_{j,k}\right), \tag{4}$$

where $x_{i+j,k}$ is the input in row $i+j$ and column k, $w_{j,k}$ is the weight in row j and column k of the convolution kernel, and $g()$ is the activation function. In a 1D CNN, $\begin{bmatrix} w_{1,1} & \cdots & w_{1,P} \\ \vdots & \ddots & \vdots \\ w_{S,1} & \cdots & w_{S,P} \end{bmatrix}$ forms a kernel with size S. Through training, a kernel obtains proper values, therefore it can extract particular features from the previous layer. Usually, a 1D CNN has multiple kernels with the same size and extracts various features, producing a 2D array with

Table 5: Traffic Classification using MLP [1]

	Input Features	MLP Architecture	Accuracy	Precision
Scheme 1	Feature Group A	11 – 1600 –1200 – 880 – 640 – 220 – 14	0.862	0.870
Scheme 2	Feature Groups A, B, C	599 – 1600 – 1200 – 1200 – 880 – 640 – 360 – 220 - 80 – 14	0.844	0.870
Scheme 3	Feature Groups A, B, C, D, E	937 – 1600 – 1200 – 880 – 640 – 360 – 220 – 80 – 14	0.928	0.933

[1] *Each number of the MLP architecture represents the number of neurons in a layer, starting from the input layer and ending at the output layer.*

each column representing features generated by a kernel (e.g., filter). After that, a pooling function is applied to make the representation approximately invariant to small translations of the input. The convolution and pooling can be applied multiple rounds with various kernel size, number of kernels, and pooling rate. Finally, the 2D array is flattened to a 1D array by a fully connected dense layer.

In this work, our goal is to extract correlative feature among packet sequence, where is carried by feature group E: Packet Sequence Information. For each flow, we have extracted the first 50 packets' information, and each packet is represented as a 3-element tuple, {b, ipt, dir}. Therefore, for the 1D CNN, we first reshape the feature array to a $n \times 3$ matrix, where $n \times 3$ equals to the number of all features. Following that, various number of convolution layer are adopted. Considering that in addition to packet features, there are also features without any sequential correlation in flow data, such as features in groups A and B, several fully connected layers are also adopted. The CNN architecture for traffic classification is demonstrated in Figure 6.

Using a CNN shown in Figure 6, various options on feature selection as the CNN input were considered. For each input scheme, a CNN network was determined after multiple trails of optimizations. Table 6 shows the 1D CNN architecture used for traffic classification and their performances.

Similar to the MLP architectures, the CNN with 937 input features yields the best performance. This indicates that using the basic flow information accompanied with flow statistics and packet sequence information for traffic classification improves the classification accuracy and precision significantly. For the CNN scheme 3, i.e., using all five feature groups as input, various CNN architectures have been tested, as shown in Figure 7. Note that for the X-axis of Figure 7, (N, K) represents a 1D CNN layer with N filters and kernel size K. A single number M represents a fully connected layer with M neurons. From the chart we see that, the CNN with two convolutional layers followed by three fully connected layers yields the best performance.

3.3 Comparison

A couple of traffic classification models have been found in literature. Draper-Gil et al. have used decision tree to classify traffic type [3]. Wang et al. have proposed a 1D CNN network on raw traffic data [13]. In Table 7, we compared the accuracy and precision of those models to ours. All of them used ISCX dataset and classified the traffic into 14 categories.

Table 7 shows a significant improvement of our feature-based neural network models compared to the existing models. The classification accuracy has been improved by 7.2% and the precision has been improved by 19.2%. This is because that the features we were using were extracted from the entire flow data, therefore they are completer and more comprehensive compared to using part of the raw data of the flow as the neural network input. Meanwhile, our MLP models show better performances than the CNN models, even with a smaller number of neurons and shorter training period.

4 Conclusions

A novel traffic classification approach using neural networks was proposed in this paper. Rather than using raw traffic data as the neural network input, we used features of the traffic flow. After a comprehensive analysis on network flow features, we designed various schemes in feature selection. Based on the features been used as input, various neural networks were designed, implemented, and optimized. Compared to state-of-the-art traffic classification models, our MLP and CNN neural networks increased accuracy by 7.2% and increased precision by 19.2%. Furthermore, with the concern of computational complexity, models with significantly less input and smaller neural networks were designed in our work, which achieved relatively high accuracy with significantly lower computation. Changing the number of output layer neurons and using a training dataset labeled as normal and malicious, our models can also be used for network intrusion detection and cybersecurity.

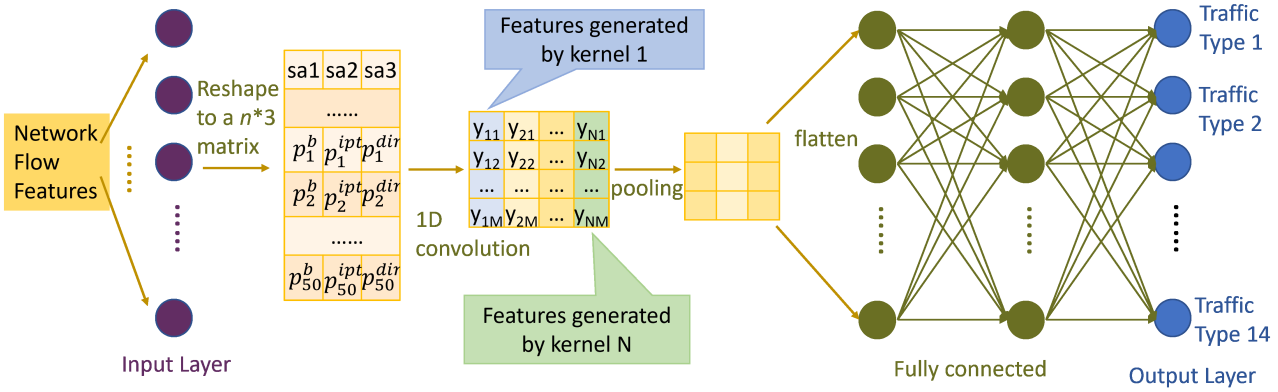


Figure 6: Traffic Classification using CNN

Table 6: Traffic Classification using 1D CNN [1]

	Input Features	CNN Architecture	Accuracy	Precision
Scheme 1	Feature Group A	11– CNN (100,2) – CNN (100,2) – 64 – 14	0.855	0.870
Scheme 2	Feature Groups A, B, C	599 – CNN (100,100) – CNN (100,80) – 220 –14	0.874	0.882
Scheme 3	Feature Groups A, B, C, D, E	937 – CNN (100,100) – CNN (100,80) – 1200 – 580 – 120 –14	0.922	0.930

[1] *CNN (N, K) represents a 1D CNN layer with N filters and kernel size K. A single number M without any letter in front represents a fully connected layer with M neurons.*

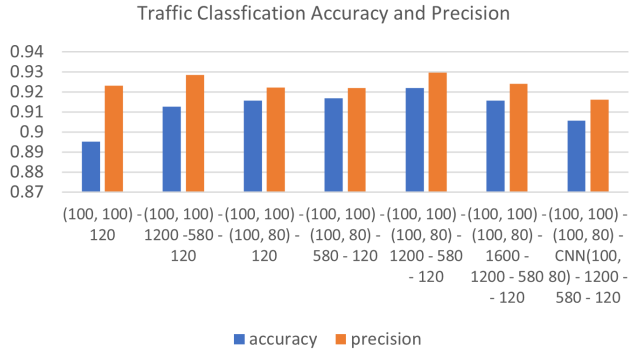


Figure 7: Performance with Various CNN Architectures

Table 7: A Comparison on Traffic Classification Performances

	Accuracy	Precision
C4.5 [3]	\	0.783
1D CNN on raw data [13]	0.866	\
Proposed MLP model	0.928	0.933
Proposed 1D CNN model	0.922	0.930

Acknowledgments

This study was supported by summer STEM research program at Whitworth University. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] Z. Chen, K. He, J. Li, and Y. Geng, "Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks," in *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, Dec 2017, pp. 1271–1276.

[2] A. Dewanje and K. A. Kumar, "A new malware detection model using emerging machine learning algorithms," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 24–32, 2021.

[3] G. D. Gil, A. H. Lashkari, M. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, Rome, Italy, Feb. 2016, pp. 407–414.

[4] W.-T. Hao, Y. Lu, R.-H. Dong, Y.-L. Shui, and Q.-Y. Zhang, "Adaptive intrusion detection model based

on cnn and c5.0 classifier," *International Journal of Network Security*, vol. 24, no. 4, pp. 648–660, 2022.

[5] Information Sciences Institute, "Internet protoco: Darpa internet program protocol specificationl," Tech. Rep. RFC791, Sept. 1981.

[6] S. S. Jajoo and K. A. Kumar, "A review on deep-learning based network intrusion detection systems," *International Journal of Electronics and Information Engineering*, vol. 13, no. 4, pp. 170–179, 2021.

[7] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 20, pp. 1–22, 2019.

[8] D. McGrew, B. Anderson, P. Perricone, and B. Hudson, "Joy software (cisco systems advanced security research group (asrg) and security and trust organization (sto))," Tech. Rep., 2016.

[9] Y. Nir, R. Salz, and N. Sullivan, "Transport layer security (tls) parameters," Tech. Rep. RFC5289, May 2023.

[10] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Ullah, A. Nazir, and A. Wajahat, "An adaptive multi-layer architecture for iot based idps for attacks using deep learning method," *International Journal of Network Security*, vol. 24, no. 5, pp. 815–827, 2022.

[11] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, 2019.

[12] P. Vanin, T. Newe, L. L. Dhirani, E. O'Connell, D. O'Shea, B. Lee, and M. Rao, "A study of network intrusion detection systems using artificial intelligence/machine learning," *Applied Sciences*, vol. 12, no. 22, pp. 61–73, 2022.

[13] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proceedings of 2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Beijing, China, July 2017, pp. 43–48.

[14] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proceedings of the 31st International Conference on Information Networking (ICOIN 2017)*, Da Nang, Vietnam, January 2017, pp. 712–717.

[15] Z. Wang, "The applications of deep learning on traffic identification," Tech. Rep. Blackhat USA 2015, 2015.

Biography

Qian Mao received her first Ph.D. degree from Tongji University (China) and second Ph.D. degree (2019) from the University of Alabama, Tuscaloosa, AL, USA. She is currently an assistant professor in the Mathematics & Computer Science Department at Whitworth University, Spokane, Washington, USA. Her research interests include wireless networks, artificial intelligence, and cyber security.

Charles O'Neill received a Ph.D. in Aerospace Engineering at Oklahoma State University in 2011. At OSU's CASELab, he supported flight testing for NASA Dryden. At Textron's Cessna Aircraft, he worked in a rapid prototyping department and developed the Textron Airland ScorpionJet. At the University of Alabama's Remote Sensing Center, he developed and operated aerial, ground, and snow-based RF systems. He is currently the Director of Research at EH Group. Dr. O'Neill holds FAA and FCC licenses.

Ke Bao received the B.S. degree in Electrical Engineering and Automation from Ocean University of China, Qingdao, China, in 2008 and the M.E. and Ph.D. degrees in Electrical and Computer Engineering from University of Alabama, Tuscaloosa, AL, USA, in 2012 and 2017, respectively. He is currently an Assistant Professor with the Department of Engineering and Engineering Technology, Metropolitan State University of Denver, Denver, CO, USA. His research interests include energy management on electric vehicle charging; cyber-physical system; software defined networking; intelligent signal processing, such as using machine learning algorithms to process sensing signals; and issues on wireless sensor network design.

A Novel Image Encryption Algorithm Based on Advanced Hill Cipher and 6D Hyperchaotic System

Mohammed Naim and Adda Ali Pacha

(Corresponding author: Mohammed Naim)

Laboratory of Coding and Security of Information & University of Sciences and Technology of Oran Mohamed Boudiaf
PoBox 1505 Oran M'Naouer 31000 Algeria

Email: mohammed.naim@univ-usto.dz, adda.alipacha@univ-usto.dz

(Received Dec. 8, 2022; Revised and Accepted Aug. 1, 2023; First Online Aug. 25, 2023)

Abstract

Digital images play an important role in the Internet and network communications era. And hence how to enhance the security of images has become a very attractive and interesting topic in information security. Therefore, image encryption is an efficient technology to protect private images. To cope with this issue, this paper introduces a novel image encryption algorithm based on a combination of advanced Hill cipher and a 6D hyperchaotic system. The proposed method used the prime number 257 as a modulo, where all the zero pixels are replaced by pixels with a value of 256. Firstly, the original image is divided into four equal parts to process each part individually. Then, each part is divided into several blocks, each consisting of four pixels. Secondly, four variables of the hyperchaotic system are used to apply the permutation operation on the blocks, where each variable is used to permute one single part. Thirdly, the remaining two variables of the hyperchaotic system are used to generate the Hill matrices. Finally, each block of each part is encrypted by the Hill cipher using one Hill matrix to obtain the final cipher image. The experimental simulation and performance analysis data demonstrated that this encryption algorithm has an extremely sensitive secret key, can resist various security attacks, and performs better than several advanced image encryption algorithms.

Keywords: Hill Cipher; Hyperchaotic System; Image Encryption; Permutation

1 Introduction

With the coming of the information age, the rapid development of the Internet, and the advancement of computer technology, there are a large number of images are transmitted on the internet every day. Therefore; the image security field received a lot of attention from researchers, governments, and companies to more enhancement and development in this field [4, 5, 16, 30].

For this reason, image encryption technology has been widely used in various fields and is considered one of the most effective and used means of image information security. In the last few years, many encryption approaches have been developed to increase the level of security of image transmissions [7, 28, 29]. Classic encryption methods, such as the Data Encryption Standard (DES), the International Data Encryption Algorithm (IDEA), the Advanced Encryption Standard (AES), and the Rivest Shamir Adleman (RSA) are unable to meet the current image encryption requirements due to their large data volume, high redundancy, Strong correlation between pixels of the image [2, 6, 17, 27, 37].

For this purpose, many recent efficient image encryption algorithms are based on different theories and techniques, where the most popular and used method is chaos theory which has interstice characteristics such as pseudo-randomness, high sensitivity to the initial value, and unpredictability. Those characteristics make it very suitable for image encryption systems [23]. The recent works on chaotic cryptosystems are based on the classical confusion and diffusion architecture proposed by Shannon. The intrinsic characteristics of chaotic systems offer many advantages such as high speed in encryption, high-security level, low computational overheads, increased flexibility, increased modularity, and relative simplicity [12].

There are too many recent works that used the chaotic system in their proposed techniques to produce efficient encryption algorithms such as [25]. which used DNA computing and chaos to secure digital medical images. And that paper [12] used the chaotic system for the permutation operation in the compressive sensing (CS) technique. Other recent works that used chaotic systems are listed in the paper.

In many recent works, there is a combination between the chaotic systems and other known symmetric cipher systems to produce an effective image encryption algo-

rithm; one of the most known symmetric and efficient cipher systems is the Hill cipher.

Hill cipher is one of the known symmetric encryption algorithms based on linear matrix transformation which is invented by Lester Hill in 1929 [14]. The Hill cipher has many advantages such as hiding letter frequencies of the plain text, using simple matrix multiplication and inversion for enciphering or deciphering, high speed, and high throughput. Although that, some problems have been noticed in the encryption scheme. The inverse of a matrix may not exist due to which encryption will not be possible. Due to its linear nature, it succumbs to known-plaintext attacks. In the application of the image encryption algorithm, there is a setback, where it reveals certain trends and does not hide all the characteristics of the image (images with a strong correlation of adjacent pixels) [8].

For increasing the Hill cipher efficiency, several image encryption algorithms have been proposed [9, 10, 15, 18, 19, 24].

Essaid *et al.* [10] proposed an image encryption scheme based on a new secure variant of Hill cipher and 1D chaotic maps. The algorithm consists of two basic processes: confusion and diffusion processes. Firstly, the confusion process is ensured by the product of a vector consisting of the key-pixel couple and a 2 ×2 Hill matrix, and the addition of another pseudo-random translation vector. While the diffusion process is ensured by a strong avalanche effect that links each encrypted pixel to its adjacent. The used chaotic sequences come from 1D chaotic maps with excellent statistical proprieties. Hraoui *et al.* [15] suggested a new cryptosystem of color images using a dynamic-chaos Hill Cipher algorithm based on the improvement of the Hill cipher by using an affine transformation applied by a three-order invertible matrix and a dynamic translation vector. Where the used vector is dynamically transformed at each iteration by an affine transformation composed of a chaotic matrix, not necessarily invertible, and a pseudo-random translation vector. Dawahdeh *et al.* [9] proposed a new image encryption technique combining an elliptic curve cryptosystem with Hill Cipher to convert Hill Cipher from a symmetric technique to an asymmetric, where the self-invertible key matrix is used to generate encryption and decryption secret key. Both sender and receiver can produce the secret key with no need to share it through the internet or unsecured communication channel. Khalaf *et al.* [18] proposed an enhancement to overcome the drawbacks of Hill Cipher by using a large and random key with a large data block, besides overcoming the Invertible-key Matrix problem. Kumar *et al.* [24] proposed a chaos and Hill Cipher-based image encryption for mammography images. The algorithm consists of a permutation and diffusion process where the input grayscale image pixel positions are permuted by using an Arnold cat map. Then, the permuted image undergoes a hill cipher (matrix multiplication) with an involuntary matrix generated from the chaotic map. Mahmoud & Chefranov [19] proposed Hill Cipher modification based on pseudo-random eigenvalues for generating a new key matrix for each plaintext block by using pseudo-random eigenvalues instead of static eigenvalues exponentiated to pseudo-random powers in the algorithm. If the sender, A, and the receiver, B, want to communicate using the algorithm, they share a secret value, SEED, that is used to generate pseudo-randomly.

Despite all previously proposed approaches, there are limitations to improving the Hill cipher. The motivation for our work is to reuse the Hill cipher in a modern fashion using new techniques while preserving the mathematical concept of the Hill cipher (using Matrix) to improve the encryption of digital images. Using the matrix in the Hill cipher to find the matrix inverse for all used matrices is our idea to reuse the Hill cipher in modern techniques. The proposed solution to avoid these limitations is the proposed new image encryption algorithm based on the advancement of Hill cipher and hyperchaotic system. As demonstrated in [1] the prime number 257 can guarantee the perfect reconstruction on the decryption side by replacing the pixel values of zero with 256.

This paper presents a novel image encryption algorithm based on a combination between an advanced Hill cipher and a 6D hyperchaotic system. To enhance the efficiency of the Hill cipher, we use the prime number 257 as a modulo and change the pixels of zero value by 256 to avoid the loss of data. The proposed encryption algorithm consists of two main processes: the confusion process and the diffusion process. The diffusion process is based on the permutation operation by using four variables of the 6D hyperchaotic system, where each variable is used to shuffle one single part. While the remaining two variables of the hyperchaotic system are used in the confusion process by applying the Hill cipher to obtain the final cipher image. In addition to digital images, the proposed algorithm is good enough for the secure encryption of satellite images too.

The remainder of the paper is organized as follows. Section 2 summarizes the preliminaries. Section 3 describes the proposed encryption algorithm in detail. The simulation results and security analyses are presented in Section 4. Finally, the conclusions are given in Section 5.

2 Preliminaries

This section introduces background knowledge on the hyperchaotic system and Hill cipher.

2.1 Hyperchaotic System

This subsection presents a 6-D hyperchaotic system that will be used in the confusion and diffusion operations in the proposed encryption algorithm. In 2020, a new 6-D hyperchaotic system is introduced by, and given by Yang

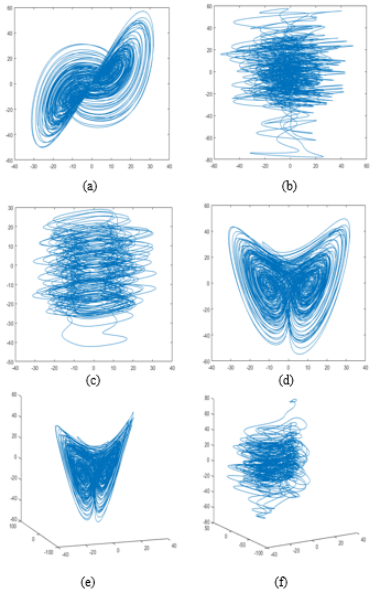


Figure 1: Phase diagram of the system: (a) x-y, (b) z-v, (c) x-w, (d) x-z, (e) x-y-z, (f) x-z-u

et al. [36], and given by

$$\begin{aligned} x' &= h(y-x)+v \\ y' &= -fy-xz+u \\ z' &= -l+xy \\ v' &= -y-w \\ w' &= ky+v \\ u' &= gx+my \end{aligned} \tag{1}$$

where $h > 0, l > 0, f, k, g$, and $m \neq 0$ are constant parameters. When $h = 10, l = 100, f = 2.7, k = 2, g = -3$, and $m = 1$; the proposed 6-D hyperchaotic system is in a hyperchaotic state, and the Lyapunov exponents for initial value (1, 1, 1, 1, 1, 1) are: $LE_1 = 1.3613, LE_2 = 0.0733, LE_3 = 0.0478, LE_4 = 0.0189, LE_5 = 0.0000, LE_6 = -14.2010$.

The phases of the 6-D hyperchaotic system are shown in Figure 1. Where Figure 1(a) represents the hyperchaotic behavior (2D) between (x and y), Figure 1(b) represents the hyperchaotic behavior (2D) between (z and v), Figure 1(c) represents the hyperchaotic behavior (2D) between (x and w), while Figure 1(d) represent the hyperchaotic behavior (2D) between (x and z), Figure 1(e) represent the hyperchaotic behavior (3D) among (x, y, and z) and Figure 1(f) represent the hyperchaotic behavior (3D) among (x, z, and u).

Compared to other chaotic systems, the proposed 6-D hyperchaotic system has the following advantages: First, four Lyapunov exponents of the 6-D hyperchaotic system are greater than zero. Second, the dynamic behavior of the 6-D hyperchaotic system (as shown above in Figure 1) is more complicated and the phase trajectories are separated in more directions. Based on these, the 6-D hyperchaotic system can improve the security of chaotic information encryption and secure communication.

2.2 Hill Cipher

Hill Cipher algorithm is one of the most famous algorithms of cryptography which is based on linear algebra. The core of the Hill cipher is matrix manipulations where the idea is a simple matrix transformation [21].

The main concept of this technique for images is based on assigning each pixel a numerical value beginning with 1 to 256 (the proposed advanced Hill cipher is based on replacing each zero-pixel value with 256). Then, the plain image is divided into blocks consisting of the same size m depending on the key matrix size mxm. After that, module 257 is taken for each block matrix element obtained by multiplication. The taken key matrix should be invertible; otherwise, decryption will not be possible.

During the inverse operation, the block matrix of the ciphered image is encrypted by the inverse of the key matrix multiplied, and finally, its module 257 is taken to obtain the original block matrix of the plain image. For example, if the block size is four ($I_{4\times1}$) then the key matrix ($K_{4\times4}$) should be of size (4×4), and the encryption process will produce a cipher text block of the image with four numerical values ($CI_{4\times1}$) as follows: For encryption,

$$\begin{aligned} \begin{bmatrix} CI_1 \\ CI_2 \\ CI_3 \\ CI_4 \end{bmatrix} &= \begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \end{bmatrix} \bmod 257 \\ &= \begin{bmatrix} (K_{11}\times I_1 + K_{12}\times I_2 + K_{13}\times I_3 + K_{14}\times I_4) \\ (K_{21}\times I_1 + K_{22}\times I_2 + K_{23}\times I_3 + K_{24}\times I_4) \\ (K_{31}\times I_1 + K_{32}\times I_2 + K_{33}\times I_3 + K_{34}\times I_4) \\ (K_{41}\times I_1 + K_{42}\times I_2 + K_{43}\times I_3 + K_{44}\times I_4) \end{bmatrix} \\ &\bmod 257 \end{aligned}$$

And for decryption, $I = K^{-1} \times CI$; where K^{-1} is the Modular Arithmetic inverse of the key matrix. $K \times K^{-1} = I$, and I is the identity matrix. The Modular Arithmetic inverse of the encryption matrix must be found. Therefore, the proposed algorithm presents the solution to find the inverse matrix always.

3 The Proposed Encryption Algorithm

In this work, we propose a new image encryption algorithm to increase encryption and transmission security. This proposed encryption algorithm is based on a combination of an advanced Hill cipher and a 6D hyperchaotic system.

3.1 The Encryption Process

Figure 2 illustrates the proposed encryption algorithm. The encryption process consists of several steps as follows:

Step 1: The original image is divided into four parts of the same size $m \times m$.

Step 2: The diffusion process (permutation operation) is applied on each part (1, 2, 3, and 4) by using four variables of the hyperchaotic system v, w, u, and z respectively, which means the permutation operation will be applied individually on each part by using one variable for each part. But the permutation operation will not permute the position of the four parts, it will permute the blocks inside each part. Because each part consists of several blocks, and each block consists of four adjacent pixels. The permutation operation is applied to change the position of each block.

Step 3: The hyperchaotic system generates six different sequences by using the initial values $x_0, y_0, z_0, u_0, v_0, and w_0$. Then, four sequences v, w, u, and z are taken to be used in the permutation operation by taking N values (where N is the number of blocks of each part) to generate four sequences V, W, U, and Z. The values of the sequences are sorted in ascending order, and the index sequences are considered as new sequences IV, IW, IU, and IZ respectively which are used for shuffling the N blocks. The new sequences are given by:

$$\begin{aligned} V &= [v_p, v_{p+1}, ...v_{p+N-1}] \\ (!, IV) &= sort(V) \\ W &= [w_p, w_{p+1}, ...w_{p+N-1}] \\ (!, IW) &= sort(W) \\ U &= [u_p, u_{p+1}, ...u_{p+N-1}] \\ (!, IU) &= sort(U) \\ Z &= [z_p, z_{p+1}, ...z_{p+N-1}] \\ (!, IZ) &= sort(Z) \end{aligned} \tag{2}$$

where p represents the starting number. The permutation operation is based on changing the positions of the blocks according to the generated index sequences to obtain the four shuffled parts.

Step 4: After the permutation operation, the vector transformation is applied on each shuffled part to obtain four different vectors $V1, V2, V3$, and $V4$.

Step 5: Each variable of the remaining two variables of the hyperchaotic system is used to generate N matrix to be used as Hill matrices ($D1$, $D2$). Each element of the used matrices is calculated by:

$$\begin{aligned} x_i &= mod(floor(x \times 10^7), 256) + 1 \\ y_i &= mod(floor(y \times 10^7), 256) + 1 \end{aligned} \tag{3}$$

$$D_1 = \begin{bmatrix} x_i & x_{i+4} & x_{i+8} & x_{i+12} \\ x_{i+1} & x_{i+5} & x_{i+9} & x_{i+13} \\ x_{i+2} & x_{i+6} & x_{i+10} & x_{i+14} \\ x_{i+3} & x_{i+7} & x_{i+11} & x_{i+15} \end{bmatrix} \tag{4}$$

$$D_2 = \begin{bmatrix} y_i & y_{i+4} & y_{i+8} & y_{i+12} \\ y_{i+1} & y_{i+5} & y_{i+9} & y_{i+13} \\ y_{i+2} & y_{i+6} & y_{i+10} & y_{i+14} \\ y_{i+3} & y_{i+7} & y_{i+11} & y_{i+15} \end{bmatrix} \tag{5}$$

Step 6: Hill cipher takes the matrix which has the highest determiner to minimize the possibility of choosing a matrix with a determiner equal to zero (in this case, the matrix is not invertible). On the other hand, if the two matrices have a determiner equal to zero, the proposed algorithm ignores these two matrices and generates another two until finding at least one matrix without a determiner equal to zero.

Step 7: The confusion process applying the Hill cipher encryption by matrix manipulations between Hill matrices and each block of the four vectors to obtain HC_1, HC_2, HC_3, and HC_4, where each block of the four vectors is encrypted with the same matrix. The obtained vectors are given as follows

$$\begin{aligned} HC_1 &= mod(HC_1, 256) + 1 \\ HC_2 &= mod(HC_2, 256) + 1 \\ HC_3 &= mod(HC_3, 256) + 1 \\ HC_4 &= mod(HC_4, 256) + 1 \end{aligned} \tag{6}$$

The main aim of using the modulo operator is to reconstruct vectors for the decryption operation.

Step 8: Finally, the obtained vectors of the previous step are combined in four parts (Ciphered parts), then the four ciphered parts are combined to obtain the final cipher image.

3.2 The Decryption Process

The decryption process is the inverse of the encryption one, where the decryption process received the secret key that is used to encrypt the original image, and it is described in the following steps:

Step 1: The cipher image is divided into four cipher parts of the same size m x m.

Step 2: Each cipher part consists of N blocks, where these blocks consist of four adjacent pixels. Then, the inverse vector transformation is applied to each cipher part to obtain four encrypted vectors.

Step 3: The inverse of Hill cipher encryption is applied to each vector by using the same inverse matrices in the encryption process.

Step 4: The matrix transformation is applied to the transformed vectors of the previous step to obtain four square parts.

Step 5: The inverse of permutation operation is applied on each part obtained in Step 4.

Step 6: The four obtained parts are combined to obtain the plain image.

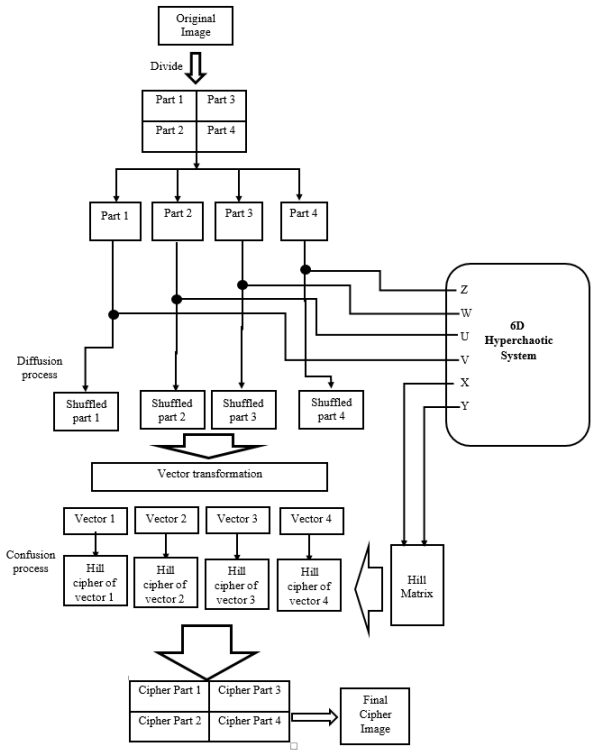


Figure 2: The proposed encryption algorithm

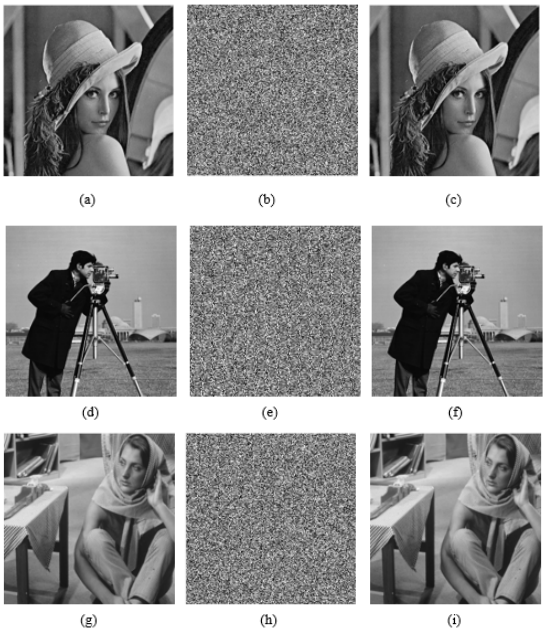


Figure 3: (a) Original image of Lena, (b) encrypted image of Lena, (c) decrypted image of Lena, (d) original image of Cameraman, (e) encrypted image of Cameraman, (f) decrypted image of Cameraman, (g) original image of Barbara, (h) encrypted image of Barbara, (i) decrypted image of Barbara.

3.3 Discussion

The proposed image encryption algorithm has the following advantages. Firstly, the proposed encryption algorithm adopts the well-known chaotic behavior by using the 6-D hyperchaotic system, which produces the desired behavior to apply in the image encryption process. Secondly, the permutation operation by the four variables of the hyperchaotic system can efficiently permute the positions of the blocks. Thirdly, the advanced Hill cipher has a high ability to encrypt and decrypt each block efficiently. Finally, the proposed algorithm has high-performance analyses and it can achieve a strong ability to resist security risks, such as differential attacks. It will be experimentally verified in Section 4.

4 The Simulation Results and Security Analyses

In this section, well-known security measures are applied to test the effectiveness of the proposed image encryption algorithm against cryptanalysis. Simulation and performance evaluation of the proposed encryption algorithm were performed on a Pentium I-3, 2.2 GHz PC with Windows 7 and 2 GB RAM. The implementation was done using Matlab 2017a software on three different images Lena, cameraman, and as shown in Figure 3. The initial parameters used in the proposed encryption algorithm to obtain the well-known hyperchaotic behavior are $x(0) = 0.25, y(0) = 0.3, z(0) = 0.5, v(0) = 0.44, w(0) = 0.06, u(0) = 0.73$.

4.1 Histogram Analysis

The histogram analysis of an image shows the distribution information of pixel values in the image by using plotting the number of observations of each brightness value. The histogram of an original image is usually unevenly distributed while it is more uniformly distributed for images encrypted by a good encryption scheme. A uniform distribution of the histogram indicates a random image and the least probability of recovering its original image and prevents the adversary from extracting any meaningful information from the fluctuating histogram of the ciphered image [31]. Figure 4 shows the histogram of the original, encrypted, and decrypted images of Lena, the cameraman, and Barbara respectively, to examine the statistical distribution by calculating and analyzing the histograms of these images.

The histogram of the ciphered images is fairly uniform and completely different from that of the plain image and decrypted one. Therefore, the proposed image encryption algorithm does not provide any clue for statistical attacks.

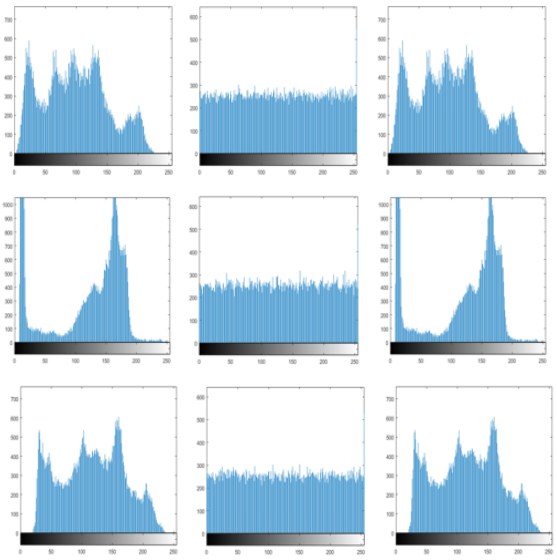


Figure 4: The histograms of Lena, Cameraman, and Barbara respectively (a) the histogram of original images, (b) the histogram of encrypted images, (c) the histogram of decrypted images

4.2 Key Space

The key space should be large enough ($>2^{100}$) to resist brute-force attacks [26]. In the proposed algorithm, the secret key includes the initial conditions of the chaotic system (x_0, y_0, z_0, v_0, w_0 and u_0). If the computing precision of the computer is 10^{-15}, the keyspace is Key $=(x(0) = 10^{15})(y(0) = 10^{15})(z(0) = 10^{15})(v(0) = 10^{15})(w(0) = 10^{15}(u(0) = 10^{15})) = 10^{90} > 2^{300} > 2^{100}$. As a result, the key space of the proposed algorithm is large enough to resist all kinds of brute-force attacks and can offer a high-security level.

4.3 Key Sensitivity Analysis

The Key sensitivity is a very essential feature for an optimal encryption system. This means that the secret key must show high sensitivity to any slight change in its values and when that happens, the encryption algorithm must produce a completely different encrypted image. The very high sensitivity to the key guarantees the security of the encryption system against attacks [20]. To test the sensitivity of the key, Lena's image is taken as the plain image and encrypted twice, one with the right secret key and the other with a tiny change (10^{-15}) in the secret key. The key sensitivity analysis results are shown in Figure 5 and Figure 6.

From Figure 5, it can be seen that the encrypted images are different when the key slightly changes. Figure 6 shows the results of the decryption process. From Figure 6, any small change in the key will result in poor decryption. Therefore, our proposed algorithm is very sensitive to the key in both encryption and decryption

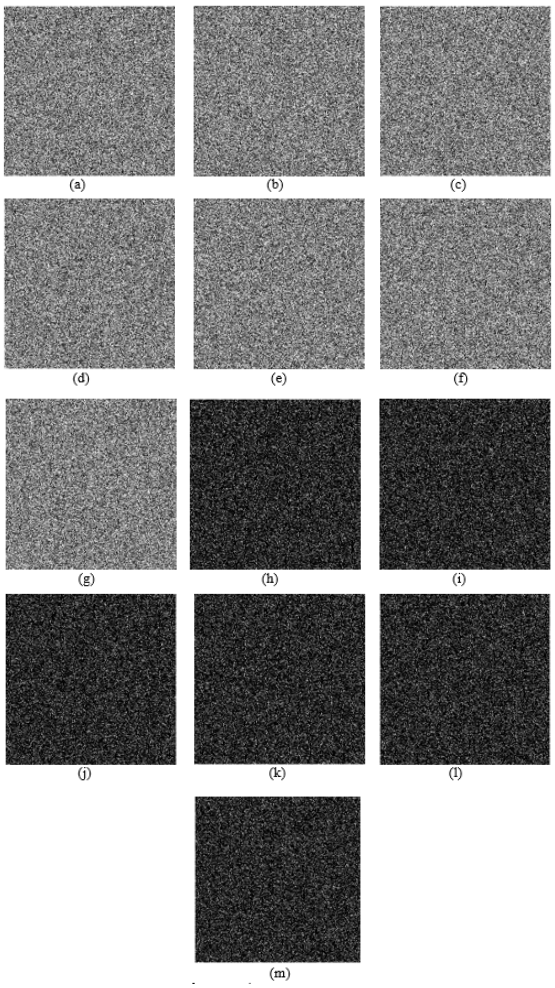


Figure 5: Secret key sensitivity in the encryption process; (a) Encrypted image with the right secret key, (b) Encrypted image with $(x(0) + 10^{-15})$, (c) Encrypted image with $(y(0) + 10^{-15})$, (d) Encrypted image with $(z(0) + 10^{-15})$, (e) Encrypted image with $(v(0) + 10^{-15})$, (f) Encrypted image with $(w(0) + 10^{-15})$, (g) Encrypted image with $(u(0) + 10^{-15})$, (h) Subtraction between (a) and (b), (i) Subtraction between (a) and (c), (j) Subtraction between (a) and (d), (k) Subtraction between (a) and (e), (l) Subtraction between (a) and (f), (m) Subtraction between (a) and (g)

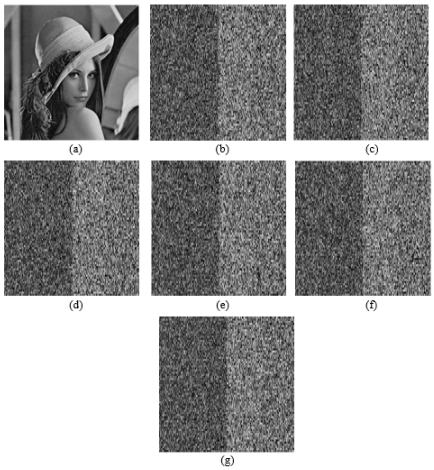


Figure 6: Secret key sensitivity in the decryption process; (a) Decrypted image with the right key. (b) Decrypted image with $(x(0) + 10^{-15})$, (c) Decrypted image with $(y(0)+10^{-15})$,; (d) Decrypted image with $(z(0)+10^{-15})$,; (e) Decrypted image with $(v(0) + 10^{-15})$,; (f) Decrypted image with $(w(0) + 10^{-15})$,; (g) Decrypted image with $(u(0) + 10^{-15})$

processes.

4.4 Correlation Coefficients Analysis

In the plain image, there is a strong correlation between the adjacent pixel values. Therefore, the image encryption algorithms should break up this high correlation between the adjacent pixels to resist the attackers' analyses [32]. To analyze the correlations of adjacent pixels of the plain image and cipher image, Lena's image is used as the test image, and we randomly pick out 5000 pairs of adjacent pixels from images in three different directions: horizontal, vertical, and diagonal. The correlation coefficient values of neighboring pixels are shown in Table 1 for Lena, Cameraman, and Barbara images of encrypted images in (Figure 3) by our proposed encryption algorithm. The correlation of adjacent pixels for the plain and cipher images (Lena image) is shown in Figure 7 in three directions: horizontal, vertical, and diagonal.

As can be seen from Figure 7 that the difference between the three directions of an encrypted image cannot be observed by the naked eye, thus for the accurate comparison, we computed the correlation coefficient measure of the plain image and the three directions of the encrypted image. For this purpose, we used the following correlation coefficient formula [3]:

$$r_xy = \frac{Con(x,y)}{\sqrt{D(X)}\sqrt{D(y)}} \tag{7}$$

$$Con(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{8}$$

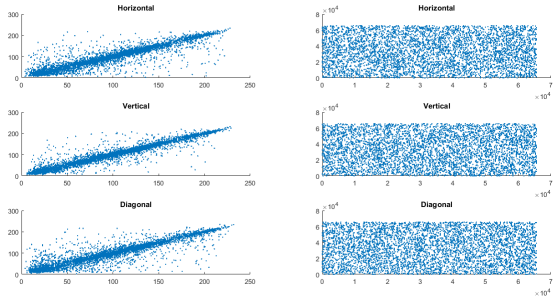


Figure 7: Adjacent pixels correlation at horizontal, vertical, and diagonal directions: (a) Plaintext image, (b) encrypted image

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i) \tag{9}$$

$$E(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i) \tag{10}$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{11}$$

$$D(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - E(y))^2 \tag{12}$$

where r_xy is the correlation coefficient of two adjacent pixels, and $E(x)$ and $D(x)$ are the expectation and variance of variable x, respectively. $E(y)$ and $D(y)$ are the expectation and variance of variable y, respectively.

Table 1: Correlation coefficients of the proposed algorithm

Images	Horizontal	Vertical	Diagonal
Lena	0.0017	-0.0027	0.0002
Cameraman	0.0028	-0.0019	0.0004
Barbara	0.0024	-0.0038	0.0065

As shown in Table 1 the correlation coefficients of the proposed algorithm are very low or practically zero. Thus, the proposed algorithm resisted statistical attacks. Table 2 shows a correlation coefficient comparison with recent works by using Lena image 512 ×512.

As shown in Table 2, the proposed encryption algorithm has better correlation coefficient values in horizontal and diagonal directions compared with other works, while for vertical direction Ref [33] has better vertical correlation coefficient values compared with others.

4.5 Differential Attack Analysis

According to the theory of cryptography, an image encryption scheme should effectively resist the differential

Table 2: Correlation coefficients of the proposed algorithm

Algorithms	Horizontal	Vertical	Diagonal
The Proposed	0.0003	-0.0029	0.00015
[10]	0.0005	0.0006	0.0029
[33]	0.0017	0.0004	0.0028
[35]	0.0056	0.0037	0.0032

attack, where the differential attack is an attack method in which the attacker chooses plaintext; then, change a part of the pixel values of the plaintext image to compare the differences between the two encrypted images to find the possibility of deciphering. Thus a good image encryption algorithm needs to be very sensitive to the original images; that is, any trivial change in the secret key should lead to a completely different encrypted image. The resistance of encryption algorithms against differential attacks can be tested by the number of pixels changing rate (NPCR), unified averaged changed intensity (UACI), and Hamming distance (HD) which are acquired as follows [11]:

$$D(i,j) = \begin{cases} 0 & \text{if } C^1(i,j) = C^2(i,j) \\ 1 & \text{if } C^1(i,j) \neq C^2(i,j) \end{cases} \quad (13)$$

$$NPCR : N(C^1, C^2) = \sum_{i=1}^{N} \frac{D(I,j)}{T} \times 100 \quad (14)$$

$$UACI : U(C^1, C^2) = \sum_{i=1}^{N} \frac{|C^1(i,j) - C^2(i,j)|}{F \times T} \times 100 \quad (15)$$

$$HD : H(C^1, C^2) = \sum_{i=1}^{N} \frac{|C^1(K) XOR C^2(K)|}{n_b \times T} \times 100 \quad (16)$$

where $C^1(i,j)$ and $C^2(i,j)$ are two-pixel values of the same position (i,j) in the two different encrypted images. F and T represent the image dimensions. The NPCR, UACI, and HD results by using three different images are listed in Table 3.

Table 3: The NPCR, UACI, and HD values of three different images

Images	NPCR	UACI	HD
Lena	99.6207	33.4907	50.1699
Cameraman	99.5819	33.6051	50.0755
Barbara	99.5667	33.6404	50.1038

The results in Table 3 indicate that the differential attack analysis tests demonstrated the sensitivity of the encrypted image with NPCR, UACI, and HD of the proposed encryption scheme are close enough to the ideal values. Table 4 shows the performance comparison in terms of differential attack analysis (NPCR and UACI) between the proposed algorithm and the recent works by using Lena image 256 ×256.

Table 4: Performance comparison in terms of Differential attack analysis (NPCR and UACI)

Algorithms	NPCR	UQCI
Proposed Algorithm	99.62	33.49
[15]	99.74	33.52
[10]	99.64	33.47
[34]	99.62	33.41

As it is illustrated in Table 4, the proposed algorithm has an NPCR value close enough to the ideal value of (99.61), while for the UACI, the proposed encryption algorithm is close enough to Ref [10] which has the closest value to the ideal value of UACI (33.46).

4.6 Information Entropy Analysis

Information entropy is a significant parameter to reflect the randomness of information. In image encryption applications, the higher the level of image confusion is, the greater the value of information entropy is [13]. The calculation formula is as follows [22]:

$$H(m) = \sum_{i=0}^{255} P(m_i) log_2 \frac{1}{P(m_i)} \quad (17)$$

where m is a set of information symbols and $P(m_i)$ represents the probability of occurrence of m_i. For grayscale images, the closer the information entropy is to 8, the stronger the randomness of the pixel value arrangement of encrypted images is. Table 5, shows the information entropy of encrypted images by applying the proposed algorithm.

Table 5: The information entropy values of three different images

Images	Entropy
Lena	7.9914
Cameraman	7.9907
Barbara	7.9896

The results of all encrypted images obtained by our proposed algorithm are close enough to the ideal value 8. Therefore, the proposed encryption algorithm demonstrates better performance, so it is enough to resist attacks. Table 6 shows the performance in terms of Entropy between the proposed algorithm and recent works.

As it is illustrated in Table 6, the proposed algorithm and Ref [15] have an Entropy value close enough to the ideal value (8).

Table 6: Performance comparison in terms of Entropy

Images	Entropy
Proposed Algorithm	7.9997
[15]	7.9998
[10]	7.9996
[34]	7.9972

4.7 Robustness Analyses

4.7.1 Noise Attack Analysis

The attackers use noise attacks such as Gaussian noise (GN), Salt and Pepper noise (SPN), and Speckle noise (SN) for disrupting the integrity of the cipher text information. Then, the receiver of the cipher text cannot decrypt the image correctly. In this paper, the proposed algorithm is used to test if it can resist noise attacks. Lena image and its cipher image without noise are shown in Figure 3, and different kinds of noise are added to it, the decrypted images of noisy images are shown in Figure 8. Moreover, to measure the quality of the decrypted image, the Peak signal-to-noise ratio (PSNR) test is used which measures the quality of the decrypted image after image processing, and its equation is given by [22]:

$$PSNR = 10log\frac{255 \times 255}{\frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(X(i,j) - Y(i,j))^2} \quad (18)$$

where M and N represent the size of the image; $X(i,j)$ and $Y(i,j)$ are the pixel values of the original image and decrypted image respectively. The higher the PSNR means that the decrypted image has less difference from the original image. The PSNR values between the decrypted with noise images and the original image are listed in Table 7. From Figure 8 and Table 7, it can be seen that the proposed algorithm has the highest resistance to noise for GS, and the decrypted image has a good visual appearance when the intensity is between $7*10^{-8}$and $4*10^{-7}$and PSNR values are more than 26.5 dB. In SN, the decrypted image has a good visual appearance when the intensity is between $5 * 10^{-6}$ and $1.5 * 10^{-5}$ and PSNR values are more than 26 dB. For SPN, our algorithm has the highest resistance, when noise intensity changes from $3 * 10^{-5}$ to $8 * 10^{-3}$ the decrypted image has the most information of an image, and PSNR is more than 22 dB. Therefore, the proposed algorithm is robust to noise to a certain degree.

4.7.2 Data Loss Attack Analysis

The encryption algorithm needs to be able to resist the data loss attack which means if there is a small loss of data in the cipher image, the decrypted image must have most information of the original image. In this paper, the encrypted Lena image loses each time $32x32, 64x64, 128x128$ blocks in the positions of the upper left corner, the middle part, and the bottom right corner. The decrypted images

Figure 8: : Decrypted images under different noise: (a) $7 * 10^{-8}$ GS,(b) $4 * 10^{-7}$ GS, (c) $8 * 10^{-7}$ GS,(d) $5 * 10^{-6}$ SN, (e) $.5 * 10^{-5}$ SN, (f) $2 * 10^{-5}$ SN, (g) $3 * 10^{-5}$ SPN, (h) $8 * 10^{-3}$ SPN, (i) $2 * 10^{-2}$ SPN.

Table 7: PSNR (dB) values between decrypted images of noisy cipher images and plain image

Noise type	Noise intensity	PSNR (dB)
GS	0.00000007	28.4610
GS	0.0000004	26.7559
GS	0.0000008	18.7691
SN	0.0000005	28.4522
SN	0.0000015	26.0495
SN	0.000002	19.8535
SPN	0.00003	28.3356
SPN	0.008	22.4118
SPN	0.02	19.2866

are shown in Figure 9 and their PSNR is listed in Table 8. From Figure 9 and Table 8, it can be seen that the recovered images have important information about the plain image. Therefore, the proposed image encryption algorithm may resist some data loss attacks.

Table 8: PSNR of data loss with different sizes and positions

Size position	Left upper	Middle	bottom right
32 x 32	26.8296	25.4114	24.4397
64 x 64	20.5066	19.1890	18.2623
128 x 128	14.6092	13.0023	12.0264

5 Conclusion

This paper introduced a new image encryption algorithm based on advanced Hill cipher and 6D hyperchaotic system. The use of the hyperchaotic system is to obtain a better diffusion process while the advanced Hill cipher is used to improve the security level of the cryptosystem in the confusion process without losing the advantages of the hyperchaotic systems. The main advantages of the proposed algorithm are a high probability of resisting a brute-force attack, high key sensitivity, the ability to defend against a differential attack, adjacent pixel correlation, and resisting robustness attack. Therefore, the proposed encryption algorithm can achieve higher security performance than several classical image encryption algorithms.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant NSC 95-2416-H-159-003. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] H. Ali Pacha, N. Hadj Said, A. Ali Pacha, and Ö.Özer, "Significant role of the specific prime number p = 257 in the improvement of cryptosystems," *Notes on Number Theory and Discrete Mathematics*, vol. 26, no. 4, pp. 213–222, 2020.

[2] F. Bao, C. C. Lee, M. S. Hwang, "Cryptanalysis and improvement on batch verifying multiple RSA digital signatures", *Applied Mathematics and Computation*, vol. 172, no. 2, pp. 1195-1200, Jan. 2006.

[3] X. Chai, J. Bi, Z. Gan, X. Liu, Y. Zhang, and Y. Chen, "Color image compression and encryption scheme based on compressive sensing and dou-

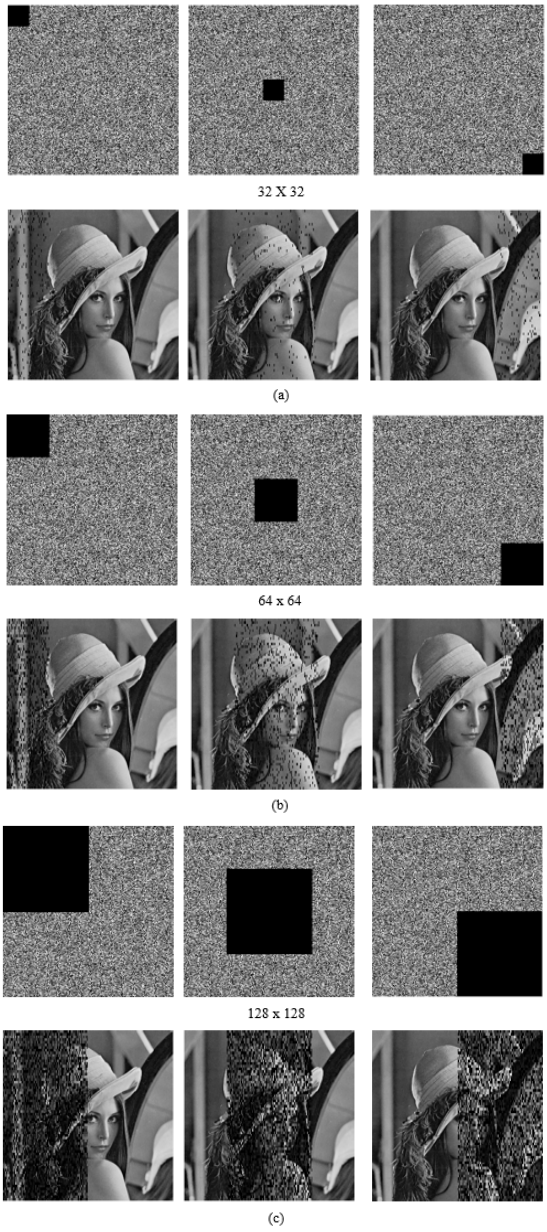


Figure 9: : Encrypted images and decrypted images with data loss of size: (a) 32 x 32 block, (b) 64 x 64 block, (c) 128 x 128 block

ble random encryption strategy," *Signal Processing*, vol. 176, p. 107684, 2020.

[4] C. C. Chang, K. F. Hwang, M. S. Hwang, "Robust authentication scheme for protecting copyrights of images and graphics", *IEE Proceedings-Vision, Image and Signal Processing*, vol. 149, no. 1, pp. 43-50, 2002.

[5] C. C. Chang, K. F. Hwang, M. S. Hwang, "A block based digital watermarks for copy protection of images," in *Asia-Pacific Conference on Communications*, vol. 2, pp. 977-980, 1999.

[6] C. C. Chang, M. S. Hwang, "Parallel computation of the generating keys for RSA cryptosystems", *Electronics Letters*, vo1. 32, no. 15, pp. 1365–1366, 1996.

[7] C. C. Chang, M. S. Hwang, T. S. Chen, "A new encryption algorithm for image cryptosystems", *Journal of Systems and Software*, vol. 58, no. 2, pp. 83–91, 2001.

[8] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.

[9] Z. E. Dawahdeh, S. N. Yaakob, and R. Razif bin Othman, "A new image encryption technique combining elliptic curve cryptosystem with hill cipher," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, p. 349–355, 2018.

[10] M. Essaid, I. Akharraz, A. Saaidi, and A. Mouhib, "Image encryption scheme based on a new secure variant of hill cipher and 1d chaotic maps," *Journal of Information Security and Applications*, vol. 47, p. 173–187, 2019.

[11] Z. Gan, X. Chai, J. Zhang, Y. Zhang, and Y. Chen, "An effective image compression–encryption scheme based on compressive sensing (cs) and game of life (gol)," *Neural Computing and Applications*, vol. 32, no. 17, p. 14113–14141, 2020.

[12] A. Hadj Brahim, A. Ali Pacha, and N. Hadj Said, "Image encryption based on compressive sensing and chaos systems," *Optics and Laser Technology*, vol. 132, p. 106489, 2020.

[13] T. Haq and T. Shah, "Algebra-chaos amalgam and dna transform based multiple digital image encryption," *Journal of Information Security and Applications*, vol. 54, p. 102592, 2020.

[14] L. S. Hill, "Cryptography in an algebraic alphabet," *The American Mathematical Monthly*, vol. 36, no. 3, p. 306–312, 1929.

[15] S. Hraoui, F. Gmira, M. F. Abbou, A. J. Oulidi, and A. Jarjar, "A new cryptosystem of color image using a dynamic-chaos hill cipher algorithm," *Procedia Computer Science*, vol. 148, pp. 399–408, 2019.

[16] L. C. Huang, L. Y. Tseng, M. S. Hwang, "The study on data hiding in medical images", *International Journal of Network Security*, vol. 14, no. 6, pp. 301–309, 2012.

[17] M. S. Hwang, C. C. Lee, Y. C. Lai, "Traceability on RSA-based partially signature with low computation", *Applied Mathematics and Computation*, vol. 145, no. 2-3, pp. 465-468,

[18] E. T. Khalaf, M. N. Mohammed, and N. Sulaiman, "Iris template protection based on enhanced hill cipher," in *ICCIS '16: 2016 International Conference on Communication and Information Systems*, pp. 53–57, Bangkok, Thailand, DEC 2016.

[19] A. Mahmoud and A. Chefranov, "Hill cipher modification based on pseudo-random eigenvalues," *Applied Mathematics and Information Sciences*, vol. 8, no. 2, pp. 505–516, 2014.

[20] A. Mansouri and X. Wang, "A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme," *Information Sciences*, vol. 520, p. 46–62, 2020.

[21] S. K. Muttoo, D. Aggarwal, and B. Ahuja, "A secure image encryption algorithm based on hill cipher system," *Bulletin EEI*, vol. 1, no. 1, pp. 51–60, 2011.

[22] M. Naim and A. Ali Pacha, "A new chaotic satellite image encryption algorithm based on a 2d filter and fisher–yates shuffling," *The Journal of Supercomputing*, 2023.

[23] M. Naim, A. Ali Pacha, and C. Serief, "A novel satellite image encryption algorithm based on hyperchaotic systems and josephus problem," *Advances in Space Research*, vol. 67, no. 7, pp. 2077–2103, 2021.

[24] S K Naveenkumar, H T Panduranga, and Kiran, "Chaos and hill cipher based image encryption for mammography images," in *2015 International Conference on Innovations in Information,Embedded and Communication Systems (ICIIECS)*, pp. 1–5, Coimbatore, India, MARS 2015.

[25] D. Ravichandran, A. Banu, B. K. Murthy, V. Balasubramanian, S. Fathima, and R. Amirtharajan, "An efficient medical image encryption using hybrid dna computing and chaos in transform domain," *Medical and Biological Engineering and Computing*, vol. 59, no. 3, pp. 589–605, 2021.

[26] S. Sun, Y. Guo, and R. Wu, "A novel image encryption scheme based on 7d hyperchaotic system and row-column simultaneous swapping," *IEEE Access*, vol. 7, pp. 28539–28547, 2019.

[27] S. Toughi, M. H. Fathi, and Y. A. Sekhavat, "An image encryption scheme based on elliptic curve pseudo random and advanced encryption system," *Signal Processing*, vol. 141, p. 217–227, 2017.

[28] M. H. Tsai, S. F. Chiou, and M. S. Hwang, "A progressive image transmission method for 2D-GE image based on context feature with different thresholds", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 2, pp. 379–386, Feb. 2009.

[29] M. H. Tsai, S. F. Chiou and M. S. Hwang, "A simple method for detecting protein spots in 2D-GE images using image contrast", *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 4617–4626, Dec. 2009.

[30] C. C. Wu, M. S. Hwang, S. J. Kao, "A new approach to the secret image sharing with steganography and authentication", *The Imaging Science Journal*, vol. 57, no. 3, pp. 140–151, 2009.

[31] Y. Xian and X. Wang, "Fractal sorting matrix and its application on chaotic image encryption," *Information Sciences*, vol. 547, p. 1154–1169, 2021.
[32] Q. Xu, K. Sun, S. He, and C. Zhu, "An effective image encryption algorithm based on compressive sensing and 2d-slim," *Optics and Lasers in Engineering*, vol. 134, p. 106178, 2020.
[33] X. Wang and N. Guan, "A novel chaotic image encryption algorithm based on extended zigzag confusion and rna operation," *Optics and Laser Technology*, vol. 131, p. 106366, 2020.
[34] X. Wang and Y. Li, "Chaotic image encryption algorithm based on hybrid multi-objective particle swarm optimization and dna sequence," *Optics and Lasers in Engineering*, vol. 137, p. 106393, 2021.
[35] J. Wu, X. Liao, and B. Yang, "Image encryption using 2d hénon-sine map and dna approach," *Signal Processing*, vol. 153, pp. 11–23, 2018.
[36] L. Yang, Q. Yang, and G. Chen, "Hidden attractors, singularly degenerate heteroclinic orbits, multistability and physical realization of a new 6d hyperchaotic system," *Communications in Nonlinear Science and Numerical Simulation*, vol. 90, p. 105362, 2020.
[37] G. Ye, "A block image encryption algorithm based on wave transmission and chaotic systems," *Nonlinear Dynamics*, vol. 75, no. 3, pp. 417–427, 2014.

Biography

M. Naim biography. Mohammed Naim is an assistant professor in both the Department of Engineering and Artificial Intelligence at the University of Palestine and the Department of Networks and Engineering at Al-Aqsa University in Gaza - Palestine. From 2018 to 2022 he was a lecturer at the Department of Electronic Engineering at USTO University and a researcher at the LACOSI laboratory in the field of information security in Oran-Algeria. He published his works in valuable research journals such as Advances in Space Research (Elsevier), The Journal of Supercomputing (Springer), and Information Security Journal: A Global Perspective (Taylor and Francis).

A. Ali Pacha biography. Adda Ali Pacha is a Professor in the Faculty of Electrical Engineering, Department of Electronics at USTO University, and responsible for the LACOSI laboratory in the field of information security in Oran-Algeria. He published his works in valuable research journals such as Advances in Space Research (Elsevier), The Journal of Supercomputing (Springer), and Information Security Journal: A Global Perspective (Taylor and Francis).

Semantic Location Privacy Protection Method Based on Differential Privacy and Temporal Association Under Continuous Query

Yonglu Wang[1], Kaizhong Zuo[2,3], Tao Pan[1], and Zhongchun Huang[1]
(Corresponding author: Yonglu Wang)

Anhui Technical College Of Mechanical and Electrical Engineering[1]
School of Computer and Information, Anhui Normal University[2]
Anhui Provincial Key Laboratory of Network and Information Security, Anhui Normal University[3]
Wuhu 241002, China
Email: 0120200013@ahcme.edu.cn

(Received Dec. 11, 2022; Revised and Accepted Aug. 5, 2023; First Online Aug. 25, 2023)

Abstract

Aiming at problems that the existing location privacy protection methods for continuous query ignore the time distribution of semantic locations and the transfer of semantic locations in the constructed dummy trajectory does not conform to the mobility model, a semantic location privacy protection method based on differential privacy and temporal association under continuous query is proposed. This method utilizes historical data to construct a transfer probability matrix. It combines the differential privacy index mechanism to comprehensively evaluate the transfer probability, query probability, and time correlation between adjacent temporal semantic locations. By selecting appropriate dummy semantic locations, a dummy trajectory that conforms to the user's mobility model is constructed, achieving the goal of confusing the user's real trajectory while protecting the privacy of the user's location. Experimental results show that the proposed algorithm enhances the degree of location privacy protection and indiscernibility of time, improves the trajectory semantic similarity, and effectively reduces the risk of location privacy and trajectory disclosure in continuous queries.

Keywords: Continuous Query; Differential Privacy; Location-based Services; Privacy Preservation; Temporal Association

1 Introduction

With the development of mobile location technology and location-based services (LBS), people can obtain convenient services by sharing location information [1, 9]. For example, people can use Meituan to deliver food and medicine by sharing location information. However, attackers can collect and analyze the location information shared by people, and then further infer their personal information [17], such as home addresses, living habits, religious beliefs, physical conditions, etc. Therefore, although people obtain convenient services, it is particularly important for them to effectively protect personal privacy when sharing location information.

Many research works have been proposed to achieve location privacy protection, including K anonymity [15], dummy location [19], mixed zones [10] and encryption [14]. Among them, dummy location is a commonly used location privacy protection method [2, 4, 6, 8, 11, 13, 16, 18, 20]. It refers to building a dummy location to replace the user's location or establish an anonymous set which contains multiple dummy locations and the user's location, so as to protect the user's location. In the continuous query, the dummy trajectory is formed by the dummy location constructed at each time, which can reduce the probability that the attacker identifies the user's trajectory.

Literature [2] proposed scheme to protect users' real location by carefully selecting a number of dummy locations based on users' query probabilities. To solve the system bottleneck caused by the single third-party trusted server, literature [18] introduced multiple trusted anonymous servers to generate pseudonyms and then sent K locations to different trusted anonymous servers to protect the location information and trajectory information of each user. Literature [8] believed that the user's mobility mode should be considered when generating dummy trajectories, and proposed a trajectory privacy protection method based on gravity mobility mode. Literature [11] proposed a trusted third-party server caching-based scheme to deal with the untrustworthy and inaccurate issues of third-party servers. The scheme introduces a dummy location generation technique that allows a query-

ing user to generate a modified(or dummy) location based on the proposed algorithm. In Literature [13], dummy locations are generated by determining transitional entropy that considers the user's current location and past location. For semantic attack [15], literature [20] proposed a spatiotemporal location privacy protection algorithm based on semantic information. Based on the problem that the semantic location transfer in the dummy trajectory does not conform to the user's mobility model, the algorithm uses the location transfer probability to select the dummy semantic location and construct the dummy trajectory. Based on the user's privacy requirement and real-time location information, literature [16] exploited greedy strategy to generate secure anonymous areas, to calculate the intersection of anonymous user sets at different times and use a dynamic pseudonym mechanism to update user's identity information.

However, when selecting dummy semantic locations to construct dummy trajectories to confuse the user's real trajectory, the above methods does not consider the similarity of 24-hour access between dummy semantic locations and the user's real semantic location, nor does it comprehensively consider the transition probability and query probability between adjacent times. As shown in Figure 1, it supposes Alice leaves the company(A) at 17:00, meets her friends at the cinema(N), and then returns home(M) after shopping at the market(I). Therefore, Alice has a trajectory from company(A)−>cinema(N)−>market(I)−>home(M) and constructs a dummy trajectory from bank(B)−>restaurant(C)−>park(O)−>hospital(K). However, the opening time of Park(O) is 7:00-18:00, and no one will be there at night. At the same time, the attacker knows that the probability that the user moves from Park(O) to Hospital(K) is 0 by analyzing historical data. Besides, for the endpoint of these two trajectories, namely hospital and home, although the query probability is similar, there are more people in the hospital during the day and more people at home at night. The attacker can infer from the time that the trajectory containing home(M) is more likely to be real, thus inferring that the trajectory from the bank(B)−>restaurant(C)−>park(O)−>hospital(K) is a dummy trajectory. Therefore, attackers can easily obtain Alice's trajectory.

To address the above problem, this paper proposes a semantic location privacy protection algorithm based on differential privacy and temporal association under continuous query, which can better protect user's location privacy and trajectory privacy by selecting the best dummy semantic location to construct a dummy trajectory. Specifically, the contribution of this paper mainly includes two aspects:

1) We use the differential privacy index mechanism to comprehensively evaluate the transition probability, query probability, and temporal association between semantic locations.

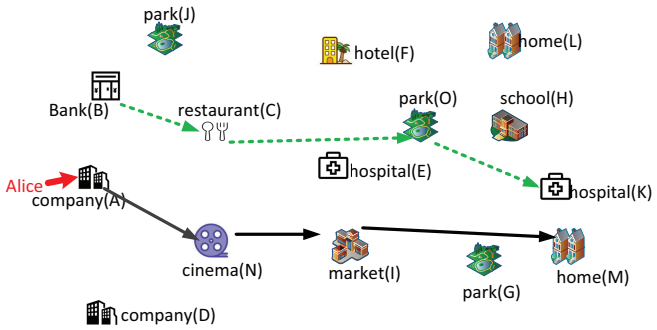


Figure 1: Motivation

2) We simulate our algorithm based on the real map and point of interest(POI) of California, compare it with other algorithms, and validate the efficacy of our algorithm.

The rest of this paper is organized as follows. In Section 2, we give the systematic framework and some basic definitions. The details of the algorithm and experimental analysis are presented in Section 3 and Section 4 respectively. Finally, we conclude our paper in Section 5.

2 Preliminaries

2.1 Related Definition

Definition 1. *(Query Request.) The query request is the content submitted by the user to the LBS server, recorded as* $req = \{id, loc, t, content, K\}$. *id is the user's identity, loc represents the semantic location, t represents the query time, content represents the query content, and K represents the minimum number of dummy semantic location set.*

Definition 2. *(Semantic Location.)* $loc = \{(x, y), type, V_{loc}\}$ *denotes the semantic location, where* (x, y) *is the coordinate of the semantic location and type is the type of the semantic location. The type of semantic location is divided into n types in total, and* $TP = \{type_1, type_2, ..., type_n\}$ *is the set of all semantic location types.* V_{loc} *represents the set of access in each hour of the day. In addition,* $N_{loc}^{t_n}$ *in* $V_{loc} = \{N_{loc}^{t_1}, N_{loc}^{t_2}, ..., N_{loc}^{t_n}, ...N_{loc}^{t_{24}}\}$ *represents the number of access at the semantic location in time* t_n.

Definition 3. *(Anonymous Semantic Location Set (AS).) The anonymous semantic location set is a set that contains the user's semantic location and meets the privacy requirements, recorded as* $AS = \{loc_{real}, loc_1, ..., loc_i, ...loc_{K-1}\}$, *where* loc_{real} *represents the user's semantic location, and* loc_i *represents the i-th dummy semantic location.*

Definition 4. *(Trajectory.) Trajectory is a set of ordered semantic locations where the user continuously*

sends query requests for a period of time, recorded as $tr = \{(loc_1, t_1), (loc_2, t_2), ...(loc_i, t_i), ..., (loc_n, t_n)\}$, *where* $tr[i]$ *represents the semantic location of the user at time* t_i, *and* $t_1 < t_2 < ... < t_i < ... < t_n$.

Definition 5. *(Cosine Similarity between Semantic Locations.) If the two semantic locations are recorded as* loc_1 *and* loc_2 *respectively,* V_{loc_i} *and* V_{loc_j} *will represent the number of access of* loc_1 *and* loc_2 *at different times respectively. The cosine similarity between semantic locations is formalized as:*

$$sim(loc_1, loc_2) = \frac{\sum_{i=1}^{n} N_{loc_1}^{t_i} \times N_{loc_2}^{t_i}}{\sqrt{\sum_{i=1}^{n} (N_{loc_1}^{t_i})^2} \times \sqrt{\sum_{i=1}^{n} (N_{loc_2}^{t_i})^2}}, \quad N_{loc_1}^{t_i} \in V_{loc_1}, N_{loc_2}^{t_i} \in V_{loc_2}. \tag{1}$$

Definition 6. *(Transition Probability Matrix (TMP).) We model the user mobility as a Markov chain of order 1 on the set of semantic locations. So, the mobility model of a given user* u *is a transition probability matrix of the Markov chain. Let* $p(u)$ *is a transition probability matrix of user* u. *The transition probability matrix* $p(u)$ *is formalized as:*

$$p(u) = (p(x^t = loc_1 | x^{t-1}), p(x^t = loc_2 | x^{t-1}), ..., p(x^t = loc_n | x^{t-1})), \tag{2}$$

where loc_i *is the actual semantic location,* n *denotes the number of semantic locations, and* $\sum_{i=1}^{n} p(x^t = loc_i | x^{t-1}) = 1$.

2.2 System Model

As shown in Figure 2, this paper adopts a central anonymous server architecture, which is composed of the user, an anonymous server, and a LBS service provider. Since the anonymous server is completely trusted, there is no privacy disclosure risk at the LBS service provider. The main workflow is as follows:

1) The anonymous server stores the city map, semantic location database, and transition probability matrix;

2) The user sends a query request to the anonymous server. After receiving the request, the anonymous server constructs an anonymous set according to the user's privacy requirement and the dummy semantic location selection algorithm to ensure that the anonymous set at adjacent times meets temporal association.

3) The LBS provider queries and returns all candidate results to the anonymous server;

4) The anonymous server filters the candidate results, and then sends the accurate results to the user.

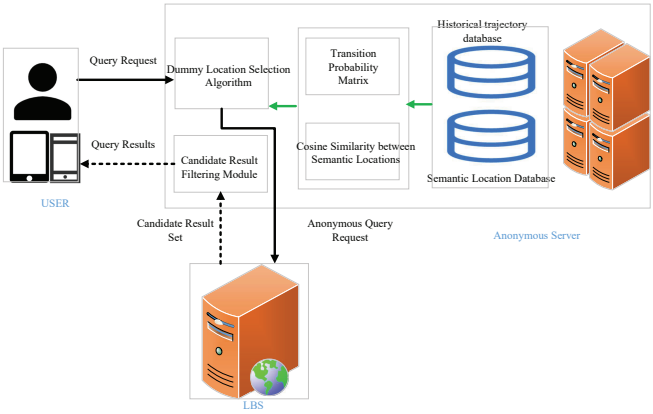


Figure 2: Server Architecture

3 Differential Privacy and Temporal Association Semantic Location Privacy Protection Algorithm

The algorithm first constructs a candidate dummy semantic location set(CDSL) based on $2K - 2$ semantic locations nearest to the user's semantic location at the current time. How to select $K - 1$ dummy semantic locations in $CDSL$ is very important. Therefore, our algorithm evaluates each semantic location in $CDSL$ by using the differential privacy index mechanism and evaluation function. $Q(loc_i^t)$ is the product of the cosine similarity between loc_i^t in $CDSL$ and the user's semantic location loc_{real}^t , the query probability of loc_i^t, and the probability that the corresponding semantic location loc_j^{t-1} in AS^{t-1} moves to loc_i^t. The formula is:

$$Q(loc_i^t) = sim(loc_{real}^t, loc_i^t) \times p(loc_i^t) \times p(loc_i^t | loc_j^{t-1}), \quad loc_i^t \in CDSL, loc_j^{t-1} \in AS^{t-1}. \tag{3}$$

After getting the score of each candidate semantic location in $CDSL$ from Equation (3), it is necessary to calculate the weight $W(loc_i^t)$ of each semantic location in $CDSL$ according to the index mechanism, and rearrange $CDSL$ from high to low. The formula is:

$$W(loc_i^t) = exp(\frac{\epsilon \times Q(loc_i^t)}{2 \times \Delta Q}), \tag{4}$$

ϵ represents the privacy budget, and ΔQ represents the sensitivity of the evaluation function, which is the maximum value of the difference between $Q(loc_{real}^t)$ and all semantic location scores in $CDSL$. The formula is:

$$\Delta Q = max||Q(loc_{real}^t - Q(loc_i^t)||_1, loc_i^t \in CDSL. \tag{5}$$

Algorithm 1 illustrates the details for dummy semantic location selection. The specific steps are as follows:

1) Initialize the input parameters;

2) Find the nearest $2K-2$ semantic locations according to the user's semantic location, which is recorded as $CDSL$;

3) Determine whether the user is the first one to submit a query request. If it is, according to the query probability and cosine similarity between semantic locations, the best $K-1$ semantic location is selected in $CDSL$, otherwise, to execute Step 4);

4) The best dummy semantic location at the current time is selected according to the transition probability, query probability and cosine similarity of each semantic location in AS^{t-1} moving to the candidate semantic location in $CDSL$ at the previous time and finally add them to AS^t;

5) Add the dummy semantic location set AS^t to the trajectory set(TS), and return to AS^t.

Algorithm 1 Differential Privacy and Temporal Association Semantic Location Privacy Protection Algorithm(DPTASP)

Input: semantic location of time t, privacy requirement req, city map map, privacy budget ϵ,transition probability matrix TMP,trajectory set TS
Output:AS^t

$AS^t \leftarrow \varnothing$
$AS^t \leftarrow AS^t \bigcup loc^t_{real}$
According to loc^t_{real} , the nearest 2K-2 semantic locations are constructed as candidate dummy semantic location set $CDSL$
if u is the first query **then**
  for each loc^t_i in $CDSL$ **do**
    $Q(loc^t_i) = sim(loc^t_{real}, loc^t_i) \times P(loc^t_i)$
    $W(loc^t_i) = exp(\frac{\epsilon \times Q(loc^t_i)}{2 \times \Delta Q})$
    descend sorted $CDSL$ by $W(loc^t_i)$
  end for
else
  $AS^{t-1} = TS.get(t-1)$
  for each loc^{t-1}_j in AS^{t-1} **do**
    for each loc^t_i in $CDSL$ **do**
      $Q(loc^t_i) = sim(loc^t_{real}, loc^t_i) \times P(loc^t_i) \times P(loc^t_i | loc^{t-1}_j)$
      $W(loc^t_i) = exp(\frac{\epsilon \times Q(loc^t_i)}{2 \times \Delta Q})$
      descend sorted $CDSL$ by $W(loc^t_i)$
    end for
    $AS^t \leftarrow AS^t \bigcup CDSL.get(0)$
    $CDSL.remove(0)$
  end for
end if
$TS.add(AS^t)$
return AS^t

4 Experiment and Analysis

4.1 Experiment Data Sets and Parameter Settings

The experiment compares and evaluates our algorithm, DPTASP, with SCLPP proposed in [20], DPSPP proposed in [16] and TTcloak proposed in [12] from the aspects of location entropy, cosine similarity, trajectory similarity, location privacy level, and scalability. The experimental environment is AMD A10-5750M CPU @2.5 GHz; 12GB RAM. The operating system is Microsoft Windows 7 Professional, and the algorithm is developed in Java based on MyEclipse.

The experimental data is based on the California map, which includes 20934 POIs(semantic locations) [7]. In addition, 778364 records are extracted from Gowalla according to the longitude and latitude of California [5], and the historical check-in data of each semantic location is constructed. There are 10000 uniform distribution users obtained from Brinkhoff based network mobile object generator [3] by introducing the highway network of California city into Brinkhoff generator. Each user contains 10 snapshot locations. For the convenience of calculation, the privacy budget ϵ is 1, and semantic location types are as office, travel, park, entertainment, school, hospital, and others respectively. All experimental parameters are shown in Table 1.

4.2 Analysis of Experimental Results

(1) Location entropy. This measures the uncertainty of the real semantic location. The higher the location entropy, the higher the degree of privacy protection, and vice versa. The calculation formula of location entropy is as follows: $H = -\sum_{i=1}^{n} q_i \times log_2 q_i$. Among them, q_i means normalizing the historical query probability of each semantic location, namely $q_i = \frac{p_i}{\sum_{i=1}^{n} p_i}$. Figure 3 compares the effect of K value on location entropy. As shown in Figure 3, with the increase of K, the curve of location entropy shows a slow upward trend, and our algorithm has achieved the best effect after $K > 20$. The reason for this phenomenon is that our algorithm considers the query probability of semantic location, so that it reduces the degree of difference in query probability between different semantic locations. Since SCLPP considers the transition probability of adjacent time, its location entropy is similar to ours. Since DPSPP and TTcloak randomly select dummy semantic locations, location entropy is lower than other algorithms.

(2) Cosine similarity. It refers to the similarity of users' access at different times between semantic locations. The higher the similarity, the higher the degree of protection. Figure 4 compares the effect of K value on cosine similarity. As shown in Figure 4, with the increase of K, our scheme achieves the highest cosine similarity compared with the other three algorithms. The main reason for this phenomenon is that compared with the other three

Table 1: Parameter Settings.

Parameters	Default values	Range
The number of users	10000	
K	30	[10,60]
The number of semantic locations	15000	[7500,17500]
Times	10	

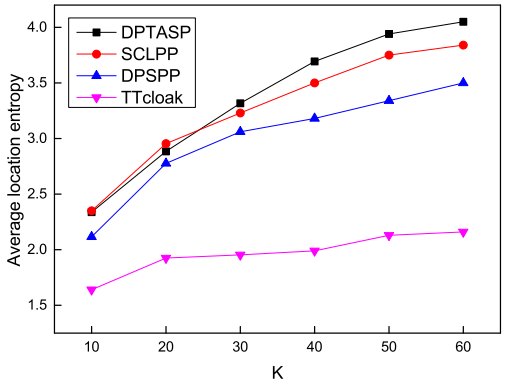


Figure 3: location entropy

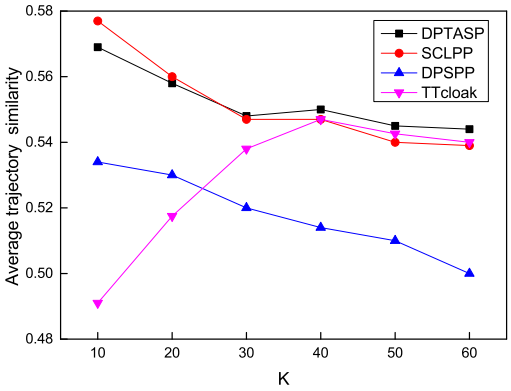


Figure 5: trajectory similarity

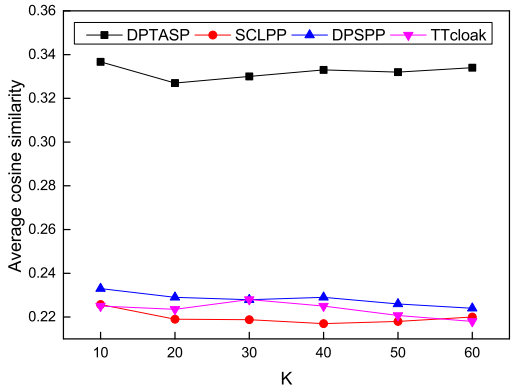


Figure 4: cosine similarity

algorithms, our algorithm selects the dummy semantic location based on users' access at different times, which improves the cosine similarity of dummy semantic locations near the request location. It is difficult for the attacker to distinguish the dummy semantic location from the temporal association. However, the other three algorithms do not consider this factor.

(3)Trajectory similarity. This measures the cosine similarity of the number of semantic locations of each type in the two trajectories. The formula is:

$$cos(tr_1, tr_2) = \frac{\sum_{type \in TP} (count(tr_1^{type}) \times count(tr_2^{type}))}{\sqrt{\sum_{type \in TP} (count(tr_1^{type})^2} \times \sqrt{\sum_{type \in TP} (count(tr_2^{type}))^2}}. \quad (6)$$

The higher the cosine similarity, the smaller the difference between the two trajectories, and the more similar the trajectories. Figure 5 compares the influence of K value on trajectory similarity. As shown in Figure 5, our algorithm achieved higher trajectory similarity than the other three algorithms, especially in the case of increased K value. With the increase of K, our dummy semantic location select method based on query probability and temporal association can select the same semantic location type each time and achieve better results than others. Since SCLPP only considers the user's mobile pattern but does not fully consider the temporal association, so its trajectory similarity is lower than ours. As DPSPP, considering semantic security, selects dummy semantic locations of different types as far as possible, its results are the worst. Since TTcloak does not consider the type of semantic location, and randomly selects a dummy semantic location, its changes are significant.

(4)Location privacy level(LPL). It refers to measure the ability of the adversary to reduce the level of privacy. We assume that the LBS server uses the background information to exclude some dummy semantic locations. The LPL can be defined as follows: $LPL = ln(size(AS^r))$, where AS^r is the set of locations that remain out of AS after excluding the locations that do not contain the target user. It is worth noting that LPL metric can be used to quantify the impact on the privacy level of any attack that aims at reducing the anonymity set by excluding some dummy semantic locations, such as the query probability and transition probability are equal to 0, or cosine similarity is less than 0.3. Figure 6 compares the influence of K value on LPL. With the increase of K, the LPL of all the algorithms presents an upward trend, and

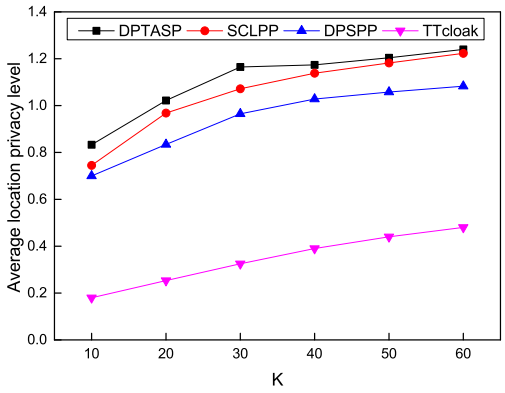


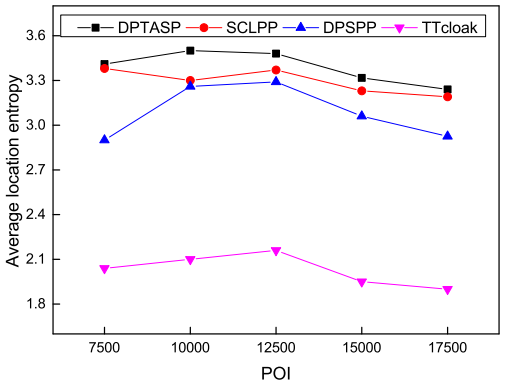
Figure 6: location privacy level

our scheme achieves the highest LPL compared with the other three algorithms. The main reason for this phenomenon is that our algorithm not only adopts the location select method based on user's mobile pattern, which improves the correlation between adjacent locations, but also improves the quality of dummy semantic location selection by combining the differential privacy index mechanism with query probability and cosine similarity. Since SCLPP considers the user's mobile pattern, good results are obtained. Since DPSPP and TTcloak do not consider both the user's mobile pattern and cosine similarity, it has lower LPL than SCLPP and DPTASP.

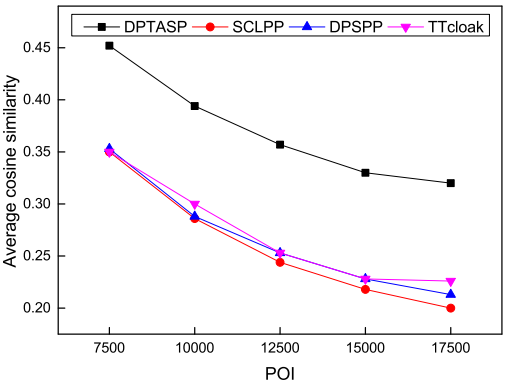
(5) Scalability. This measures the effectiveness and efficiency performances about the increasing number of semantic locations. Figure 7 shows the average location entropy, average cosine similarity, average trajectory similarity, and average location privacy level with respect to varying number of semantic locations from 7500 to 17500. In these tests, the value of K is set as 30. As show in Figure 7, with the number of semantic locations increases, our algorithm achieves the best effect. As the scale of semantic locations increases, the probability of selecting the same type of semantic location increases, which improves trajectory similarity. Meanwhile, due to the dispersion of historical query data, the location entropy, location privacy level, and cosine similarity decrease. However, in this case, our algorithm obtains better results than the other three algorithms because of selecting the dummy semantic location by combining differential privacy index mechanism with transition probability, query probability, and cosine similarity.

5 Conclusions

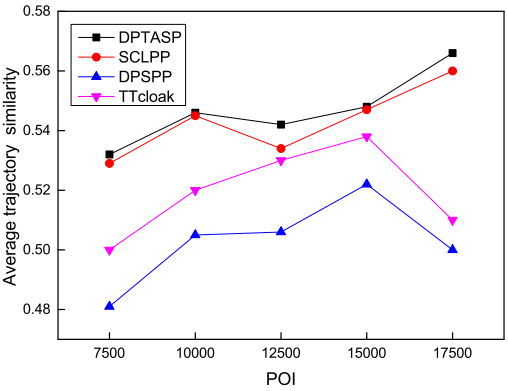
Since the transition probability between semantic locations at adjacent times and the temporal correlation between semantic locations are not fully considered when constructing dummy trajectories in the continuous query scenario, we propose a semantic location privacy protection algorithm based on differential privacy, and temporal correlation under continuous query. The algorithm



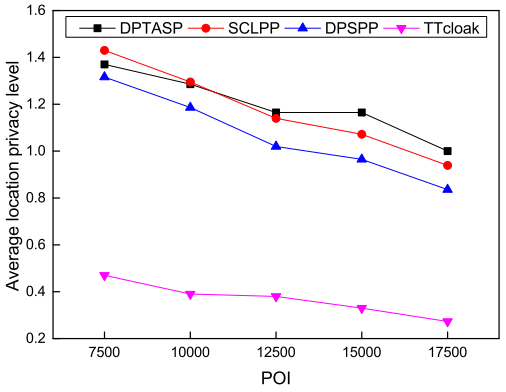
(a) location entropy



(b) cosine similarity



(c) trajectory similarity



(d) location privacy level

Figure 7: Scalability

uses first-order Markov to construct the semantic location transition probability matrix, comprehensively evaluates the transition probability, query probability and cosine similarity between semantic locations at adjacent times according to the differential privacy index mechanism, selects the optimal dummy semantic location, constructs a dummy trajectory that conforms to the mobile model, improves the similarity between the true trajectory and the dummy trajectory, and enhances the privacy protection of semantic locations.

Acknowledgments

This paper is supported by the Anhui Quality Engineering through project 2019DSGZS42, and Anhui Natural Science Foundation through project 2022AH052367.

References

[1] H. Alamleh and A. A. AlQahtani, "Architecture for continuous authentication in location-based services," in *International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, Sakheer, Bahrain, December 2020, pp. 1–4.

[2] M. Alotaibi, M. I. Ibrahem, W. Alasmary *et al.*, "Ubls: User-based location selection scheme for preserving location privacy," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, Montreal, QC, Canada, July 2021, pp. 1–6.

[3] T. Brinkhoff, "A framework for generating network-based moving objects," *Geoinformatica*, vol. 6, no. 2, p. 153–180, 2002.

[4] Y. C. Chen, W. L. Wang, M. S. Hwang, "RFID authentication protocol for anti-counterfeiting and privacy protection", in *The 9th International Conference on Advanced Communication Technology*, pp. 255-259, 2007.

[5] Gowalla, *SNAP: Network Datasets*, Aug. 15, 2023. (http://snap.stanford.edu/data/loc-gowalla.html)

[6] C. T. Li, M. S. Hwang, Y. P. Chu, "Further improvement on a novel privacy preserving authentication and access control scheme for pervasive computing environments", *Computer Communications*, vol. 31, no. 18, pp. 4255–4258, Dec. 2008.

[7] R. Liu, K. Zuo, Y. Wang *et al.*, "Location privacy-preserving method based on degree of semantic distribution similarity," in *International Conference of Pioneering Computer Scientists, Engineers and Educators*, Taiyuan, China, September 2020, pp. 118–129.

[8] X. Liu, J. Chen, X. Xia *et al.*, "Dummy-based trajectory privacy protection against exposure location attacks," in *International Conference on Web Information Systems and Applications*, Qingdao, China, September 2019, pp. 368–381.

[9] D. Lu, Q. Han, and K. Zhang, "A novel method for location privacy protection in lbs applications," *Security and Communication Networks*, pp. 1–11, 2019.

[10] I. Memon, H. Memon, and Q. A. Arain, "Pseudonym changing strategy with mix zones based authentication protocol for location privacy in road networks," *Wireless Personal Communications*, vol. 116, no. 4, pp. 3309–3329, 2021.

[11] N. Nisha, I. Natgunanathan, S. Gao *et al.*, "A novel privacy protection scheme for location-based services using collaborative caching," *Computer Networks*, vol. 213, p. 109107, 2022.

[12] B. Niu, X. Zhu, W. Li *et al.*, "A personalized two-tier cloaking scheme for privacy-aware location-based services," in *International conference on computing, networking and communications (ICNC)*, IEEE, pp. 94–98, 2015.

[13] S. Shaham, M. Ding, B. Liu *et al.*, "Privacy preservation in location-based services: A novel metric and attack model," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 3006–3019, 2020.

[14] Z. Tan, C. Wang, C. Yan *et al.*, "Protecting privacy of location-based services in road networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6435–6448, 2020.

[15] Z. Tu, K. Zhao, F. Xu *et al.*, "Protecting trajectory from semantic attack considering k-anonymity, l-diversity, and t-closeness," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 264–278, 2018.

[16] Y. Wang, K. Zuo, R. Liu *et al.*, "Dynamic pseudonym semantic-location privacy protection based on continuous query for road network," *International Journal of Network Security*, vol. 23, no. 4, pp. 642–649, 2019.

[17] V. K. Yadav, S. Verma, and S. Venkatesan, "Linkable privacy-preserving scheme for location-based services," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7998–8012, 2021.

[18] S. Zhang, X. Mao, K.-K. R. Choo *et al.*, "A trajectory privacy-preserving scheme based on a dual-k mechanism for continuous location-based services," *Information Sciences*, vol. 527, pp. 406–419, 2020.

[19] Z. Zheng, Z. Li, H. Jiang *et al.*, "Semantic-aware privacy-preserving online location trajectory data sharing," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2256–2271, 2022.

[20] K. Zuo, R. Liu *et al.*, "Method for the protection of spatiotemporal correlation location privacy with semantic information," *Journal of Xidian University*, vol. 49, no. 1, pp. 67–77(in Chinese), 2022.

Biography

Yonglu Wang He was born in 1992. He is an assistant teacher at Anhui Technical College of Mechanical and Electrical Engineering. His major research fields include

data security and privacy preservation.

Kaizhong Zuo He was born in 1974. He is a professor and a supervisor of Master's student at Anhui Normal University. His major research fields include data security and privacy preservation.

Tao Pan He was born in 1986. He is a lecturer at Anhui Technical College of Mechanical and Electrical Engineering. His major research fields include information security and RFID technology.

Zhongchun Huang He was born in 1980. He is an associate professor at Anhui Technical College of Mechanical and Electrical Engineering. His major research fields include information security and cloud computing.

Privacy-Preserving Vehicular Cloud Computing Based on Blockchain and Decentralized Identifier

Zaishuang Liu[1], Xiaoxu Ma[1], Jian Bai[1], Min Xiao[2], and Fei Tang[2]
(Corresponding author: Min Xiao)
China Electronics Technology Cyber Security Co., Ltd.[1]
Chengdu 610041, China
School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications[2]
Chongqing 400065, China
Email: xiaomin@cqupt.edu.cn
(Received Dec. 12, 2022; Revised and Accepted Aug. 5, 2023; First Online Aug. 25, 2023)

Abstract

Vehicular cloud computing (VCC) combines vehicular ad hoc networks (VANET) and cloud computing to effectively utilize vehicle computing and storage resources and meet the needs of VANET services. However, the privacy protection of vehicle users and cloud computing security are challenges in VCC. This paper presents a VCC scheme with privacy protection and strong security. First, a blockchain-based decentralized identifier (DID) model is proposed for VANET, in which vehicles can create as many DIDs and public/private key pairs locally as possible to protect the privacy of vehicle users and ensure the security of all interactions in VCC. Then, for each vehicle DID, a vehicle management department can create an anonymous resource verifiable certificate (ARVC) in advance to qualify the vehicle for VCC and thus prevent resource information forgery in VCC. Lastly, an anonymous and dynamic group key negotiation protocol is designed to achieve secure and dynamic vehicular cloud management. The security analysis demonstrates the proposed scheme can meet the security and privacy requirements of VCC, and the performance analysis and simulation results verify the feasibility of this scheme.

Keywords: Blockchain; Decentralized Identifier; Privacy-preserving; Vehicular Cloud Computing

1 Introduction

Intelligent transportation systems are designed to provide innovative applications and services relating to traffic management, as well as to facilitate the access to information for other systems and users. The compelling motivation for employing underutilized onboard resources for transportation systems and the advancements in management technology for cloud computing resources has promoted the concept of vehicular cloud (VC) [2,6–8,25,26].

NIST [29] defines cloud computing (CC) as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources, such as networks, servers, storage, applications, and services, which can be rapidly provisioned and released with minimal management effort or service provider interaction [25, 26]. The MCC forum [28] defines mobile cloud computing (MCC) as an infrastructure in which mobile devices do not process data or store data within them, and the stress on data storage and burden on computing power are transferred from mobile devices to the Cloud. The vehicular cloud computing (VCC) combines vehicular ad hoc networks (VANET) and cloud computing technology and aggregates underutilized vehicle resources to form a temporary cloud that dynamically allocates available resources to authorized vehicles. Therefore, VCC works in the opposite direction of the MCC paradigm and allows underutilized or available vehicle resources to be harvested. A detailed comparison between CC, MCC and VCC can be found in [2]. After Olariu *et al.* [30] first put forward the concept of the VC, a lot of research has focused on the VC architecture and security [17]- [19]. In VCC, the malicious vehicles may falsely report their resources or disguise themselves [27] to attack against the privacy of VC. Nkenyereye *et al.* [19] proposed a security protocol for VCC traffic data transmission and analysis, which enables anonymous vehicle authentication through pseudonym technology and identity based encryption algorithm. Limbasiya *et al.* [21] proposed a secure and efficient VC-based information search system that can securely store and retrieve data from VC, as well as perform reliable data transmission between different entities. Shao *et al.* [33] used the bilinear pairing based cryptography algorithm to outsource data computing to VC, thus reducing the computing task of vehicles. They also presented new challenges for security and privacy of VCC, but no solutions. Zhang *et al.* [45] proposed a VC creation and data transfer communication scheme with security and privacy

protection, and this scheme uses a pseudonym technology and a dynamic identity-based asymmetric group key protocol to construct a VC, where vehicles can anonymously collect and share resources and messages. However, each vehicle needs to regularly maintain the local key parameter list for the group key update and the calculation of the group public and private key is based on the bilinear pair operation, thus its calculation cost is high. In addition, the reliance on tamper-proof devices is also vulnerable to side channel attacks. Limbasiya [22] proposed a light-weight V2V communication scheme without using tamper-proof devices, which can prevent attacks like tampering, replay, simulation, password guessing, and so on. Based on this result, they proposed a VC scheme supporting batch message validation and solving the computing limitations of on-board unit (OBU) [23]. However, the scheme cannot resist session key leakage and counterfeiting, and cannot provide secure mutual authentication and privacy protection. Zhang [44] proposed a VC communication protocol, where a task manager selects a leader vehicle and other VC members to construct VC, and issues the public private key pair to the leader and all VC members. The leader vehicle issues a task through the group signature, and the VC members verify the signature and determine whether to receive the task. However, this protocol does not encrypt messages, which is easy to be eavesdropped, and the task manager would be the bottleneck of this system. Similarly, Cui [13] proposed a centralized VC certification protocol with delay problem, where legitimate vehicles can be certified through the service provider. Jiang *et al.* [18] proposed a batch authentication and key negotiation framework for VC. A authentication token is issued to an vehicle after this vehicle is successfully authenticated based on its key by the service provider, then the vehicle uses the token to request services from a VC. However, there is the token forgery risk. Li *et al.* [20] designed an effective and secure key management protocol for autonomous VC message transmission to achieve messages authentication and confidentiality.

The VC is used for localization processing and consumption of traffic sensing data to achieve highly timely intelligent traffic management. Therefore, the centralized structure is not suitable for VC, because it is difficult to meet the requirements of high timeliness. More importantly, the VC has typical self-organization characteristics, so a trust mechanism between vehicles is required to ensure the correctness and effectiveness of VC behavior. However, the existing studies have not considered the trust problem, and the vehicle re-sources are not verified and the vehicle behaviors are not audited, which cannot prevent malicious vehicles from falsely reporting the resources, disrupting the VCC environment and reducing the efficiency.

In this paper, we present a secure and efficient VCC scheme, where a blockchain supporting decentralized identifier (DID) is introduced as a trust infrastructure and VC information sharing plat-form to authenticate vehicle resources in VC and publish VC information. Meanwhile, an anonymous dynamic group key negotiation protocol is also proposed to ensure the security and privacy of VC.

2 Preliminaries

2.1 Blockchain Based DID

Blockchain technology relies on distributed ledger, consensus mechanism and digital signature technologies to ensure the security, traceability and privacy protection of the transaction process in distributed scenarios, and can effectively solve the technical bottleneck of traditional centralized methods [9–12, 24, 32]. Blockchain does not rely on consortia or governments to be a cryptographic root of trust, but uses a consensus algorithm to achieve decentralized trust. In recent years, it has been widely used in diverse areas, including finance [1], electronic health records [38], internet of things [15], smart grid [39], and so on.

A blockchain is ideal to serve as a decentralized identifier [43], and it also is a useful tool for protect data security and users' privacy [37]. With the development of information technology, traditional identity management has evolved from isolated, centralized, federated, user centric and then to decentralized self-sovereign identity (SSI) model. The SSI is owned and controlled by a user without the need to rely on any external administrative authority and without the possibility that this identity can be taken away [14].

At present, the World Wide Web Consortium (W3C) has released two standards closely related SSI, verifiable credential [35] and decentralized identifier (DID) [36]. A verifiable credential can represent all of the same information that a physical credential represents and is a tamper-evident credential that has authorship that can be cryptographically verified. A verifiable credential is a set of one or more verifiable claims made by an issuer. A claim is an assertion made about a subject, and the verifiable claim employs cryptography to enable tamper-proof and digitally signed claims. The verifiable claim is a standard way of defining, exchanging, and verifying digital credentials and usually require three parties: the claim owner, the claim verifier and the claim issuer. The strength of the claim depends on the degree of trust the verifier has in the issuer. DID is a new type of identifier that enables verifiable, decentralized digital identity. DID enables individuals and organizations to generate our own identifiers using systems we trust, and to prove control of those identifiers using cryptographic proofs (for example, digital signatures). Because we control the generation and assertion of these identifiers, each of us can have as many DIDs as we need to respect our desired separation of identities, personas, and contexts. Each DID is associated with a public and private key pair and DID document, which can express cryptographic material, verification methods, or services and provide a set of mechanisms enabling a DID controller to prove control of the DID. In the blockchain based solution, a DID can be the address of a public key

and is stored on a blockchain along with a DID document containing the public key for the DID. The identity owner controls the DID document by controlling the associated private key.

2.2 Burmester *et al.*'s Group Key Negotiation Protocol

In [3], Burmester *et al.* presented a group key negotiation protocol with a cyclic network structure, and a specific description of the protocol is given below.

The system chooses a prime p and an element $\alpha \in Z_p$ with prime order q to make the Computational Diffie-Hellman (CDH) problem difficult, that is, given $p, q, \alpha, \alpha^a, \alpha^b$, computing $\alpha^{ab} \bmod p$ is hard, where $a, b \in Z_q$.

Let $\{u_i | i = 1, 2, ..., n\}$ be a dynamic set of users who want to generate a common key. The indices are taken in a cycle: so u_{n+1} is u_1, and u_0 is u_n.

Step 1. Each u_i, $i \in [1, n]$ selects $r_i \in Z_q$, and then computes and broadcasts $z_i \equiv \alpha^{r_i} \pmod p$.

Step 2. Each u_i, $i \in [1, n]$ checks $(z_j)^q \equiv 1 \pmod p, j = 1, 2, ..., n$, then computes and broadcasts $X_i \equiv (\frac{Z_{i+1}}{Z_{i-1}})^{r_i} \pmod p \equiv \alpha^{r_{i+1}r_i - r_i r_{i-1}} \pmod p$.

Step 3. Each u_i, $i \in [1, n]$ computes the group key as follows:
$K_i = (Z_{i-1})^{nr^i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} ... X_{i-2} \pmod p = \alpha^{r_1 r_2 + r_2 r_3 + ... + r_n r_1} \pmod p$.

The group key $K = K_1 = K_2 = ... = K_n$.

It is clear that the above protocol is vulnerable to man-in-the-middle attack, and a secure message authentication mechanism is needed to resist this attack.

3 System and Security Models

3.1 System Model

As shown in Figure 1, there are three types of entities, two types of networks and many vehicle clouds (VCs). These entities include department of motor vehicle (DMV), road side unit (RSU) and vehicle. The two types of networks are VANETs and blockchain network with DID (here Internet is not considered separately). Note that in a actual system, there may be central clouds and other edge computing entities, which are not included in this system because they will not directly affect our scheme.

DMV. A DMV is response for the management of vehicles within its jurisdiction, includes the vehicle registration, anonymous resource verifiable certificate (ARVC) distribution and management and malicious vehicle tracking.

RSU. A RSU communicates directly with vehicles within its range through dedicated short-range communications (DSRC) to collect the VC information and publish these information to the blockchain.

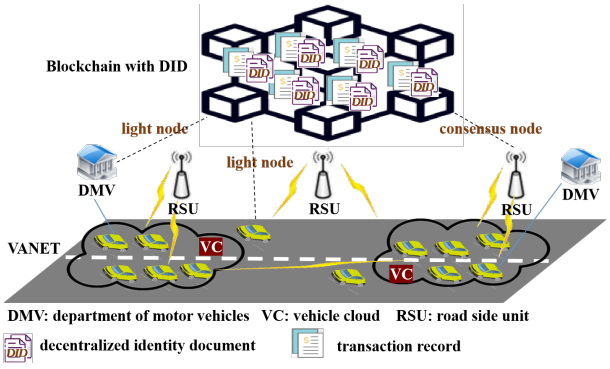


Figure 1: System model

Vehicle. Each vehicle has a certain resources, such as communication, computing, storage, and sensors, and needs to share information and resources with each other through VC technology.

VC. A VC is a self-organizing vehicle group that run a group key agreement protocol to share key, then share resources and information.

Blockchain with DID. Each entity in this system is a node of the blockchain, where RSUs are full nodes that run consensus algorithm and are responsible for maintenance of the blockchain, and the other entities are light nodes that do not participate in consensus and can access data on the blockchain. The blockchain with DID refers to a blockchain supporting decentralized identifier. In such system, each node generates many DIDs to protect users' privacy and ensure the communication security. A DID is associated with a public/private key pair and can be the blockchain address of its public key and is anonymous. The ARVC distributed by DMV is signed by the private key of its DID and can be verified by the corresponding public key.

3.2 Security and Privacy Requirements

The security assumptions in the proposed scheme are as follows. First of all, we assume that the DMVs are completely trusted. Then, the vehicles are not trusted. The vehicles may send false information or implement active attacks such as counterfeiting, forgery, or tampering. In particular, it is assumed that the member vehicles participating in the vehicle cloud computing (VCC) are semi-trusted and can guarantee the confidentiality of the negotiated VCC key, but may forge false resource information, send forged or false messages, or tamper with messages. Next, the initiator of the VCC is semi-honest, and can automatically maintain the VCC, for example, verify the ARVC of each cloud member to ensure the legitimacy of the cloud members and not broadcast false messages, but it is curious about the real identities of the vehicle users. Finally, RSUs are semi-honest and do not actively attack

VCC, but may be vulnerable, thus threatening the security and privacy of VCC.

In addition to the security of VCC process, it is also necessary to meet the security and privacy requirements in VANET [41,42]. The security and privacy requirements are listed as follows.

Preventing resource information forgery. That is to prevent the malicious vehicles from forging their own resource information when joining VC to disrupt the normal execution of VCC.

Preventing impersonation. That is to prevent the malicious vehicles from impersonating legitimate cloud members directly or by man-in-the- middle attack, to obtain illegal profits or disrupt VCC.

Information source and integrity certification. That is to ensure that the source of the message is true and the message is not maliciously tampered with.

Forward security. When a vehicle joins the VC, it can not obtain the vehicle cloud key before.

Privacy. All vehicles and RSUs cannot confirm the true identity of the message sender, and also cannot track the vehicle location information by linking messages.

4 VC Construction

The VC construction process includes four stages: system initialization, vehicle registration, VC key negotiation, and VC information release.

4.1 System Initialization

All entities in this system are registered to the blockchain with DID, and generate their own DIDs. According to the standard from W3C, a specific DID is defined as a string in the following form:

$$did : VC : unique\ identifier\ string$$

where *did* is a fixed URI scheme identifying the decentralized identifiers, VC is the did-method in the vehicle cloud environment, and *unique identifier string* is a unique value that resolves a DID to the DID document and can uniquely identify an entity.

A DID is associated with a public/private key pair (pk, sk) and a DID document, and the private key sk is securely kept by the DID owner. The DID is the blockchain address of the public key pk. Both the DID and pk are recorded in the corresponding DID document and are published on the blockchain. The basic structure of a DID document is shown in Table 1.

Generally speaking, the DIDs of DMV and RSU can remain unchanged for a long time, while the DIDs of vehicles need to be changed frequently to protect privacy. In addition, the DIDs of vehicles can provide the traceability of malicious vehicles.

Table 1: The basic structure of a DID document

{
"*id*" : $did : VC : unique\ identifier\ string$
"*controller*" : $did : VC : unique\ identifier\ string$
"*type*" : $the\ type\ of\ key\ pair\ (pk, sk)$
"*publicKey*" : pk
}

In addition, the cryptographic public parameters of this system are chosen as follows: two random large prime numbers p and q satisfying $q|(p-1)$, a cycle group G with order q, and a generator g of the group G.

4.2 Vehicle Registration

In order to achieve conditional privacy protection and obtain ARVC, a vehicle needs to be registered to a DMV in the jurisdiction when the vehicle wants to join VCC. The registration process is as follows.

Registration request. A vehicle v_i generates multiple DIDs $\{DID^1_{v_i}, DID^2_{v_i}, ..., DID^{n_i}_{v_i}\}$ and corresponding public/private key pairs $\{(pk^1_{v_i}, sk^1_{v_i}), (pk^2_{v_i}, sk^2_{v_i}), ..., (pk^{n_i}_{v_i}, sk^{n_i}_{v_i})\}$ locally (note that, $pk^j_{v_i} = g^{sk^j_{v_i}}$) and then sends a registration request message $req_{v_i} = E_{pk_{DMV}}(DID^1_{v_i} \parallel DID^2_{v_i} \parallel ... \parallel DID^{n_i}_{v_i} \parallel M \parallel t_0 \parallel \sigma)$ to the DMV in the jurisdiction, where M is the vehicle's real identity information (such as license, type and configured resource, etc), t_0 is the timestamp and $\sigma = \{\sigma_1, \sigma_2, ..., \sigma_{n_i}\}$ is the signature sequence of the request message content $M \parallel t_0$ with the private key sequence $\{sk^1_{v_i}, sk^2_{v_i}, ..., sk^{n_i}_{v_i}\}$, which proves the vehicle v_i is the owner of the identity sequence $\{DID^1_{v_i}, DID^2_{v_i}, ..., DID^{n_i}_{v_i}\}$. The request message is encrypted with the public key pk_{DMV} of the DMV, which can be searched in the DMV's DID document on the blockchain.

ARVC creation. The DMV decrypts the request message req_{v_i} and verifies the real identity information M of the vehicle v_i. If successful, it then generates an anonymous resource certificate sequence $\{ARVC^1_{v_i}, ARVC^2_{v_i}, ..., ARVC^{n_i}_{v_i}\}$ for the vehicle v_i, where the certificate $ARVC^j_{v_i}$ corresponds to the identity $DID^j_{v_i}$ for $j = 1, 2, ..., n_i$. Typically, an ARVC is a triplet (metadata, claims, proofs), where metadata includes certificate number, holder DID (i.e., a DID $DID^j_{v_i}$ of the vehicle v_i), issuer DID (i.e., DMV's DID), issue time, expiration time, etc, claims are assertions about available resources of the vehicle, such as computing, storage and network resources, and proof is the signature of the issuer DMV on the ARVC.

Registration response. The DMV encrypts each $ARVC_{v_i}^j$ with the public key $pk_{v_i}^j$ of the corresponding holder DID $DID_{v_i}^j$, which can be searched in the document of $DID_{v_i}^j$ on the blockchain, and then sends the ciphertext sequence of $\{ARVC_{v_i}^1, ARVC_{v_2}^2, ..., ARVC_{v_i}^{n_i}\}$ to the vehicle v_i. The DMV also maintains a list of the relationship between the real identity and DIDs, which is used to track malicious vehicles.

4.3 Vehicle Cloud Key Negotiation

When establishing a VC, the vehicles that want to participate in the VCC can negotiate the VC key through the group key negotiation agreement to ensure the safety of the VCC. The specific description is as follows.

VC construction initiation. Assume that the vehicle V is the temporary VC initiator, it first selects a DID_V and the corresponding $ARVC_V$ (note that, without causing confusion, for the sake of brevity, the superscripts of DID_V and its resource certificate $ARVC_V$ are omitted in the following description) and broadcasts a VC construction message $VC_{const} = (VC\ requirments, ARVC_V, \sigma_V)$, where $VC\ requirments$ are the descriptions about resource requirements and some constraints, such as the number of storage, computing, and sensing resources, the maximum number of vehicles accepted, etc., $ARVC_V$ is the resource certificate of the initiator V , and σ_V is the signature on the broadcast message with the private key sk_v of DID_V.

VC member selection. Each vehicle v_i who receives the broadcast message and wants to join VCC, uses the public key pk_{DMV} of the issuer DMV to verify the $ARVC_V$, and uses the public key pk_V of the initiator V to verify σ_V (these public keys can be searched in the corresponding DID documents on the blockchain). If successful, the vehicle v_i selects a resource certificate $ARVC_{v_i}$ of DID_{v_i}, generates a encrypted response message $VCresp_{v_i} = E_{pk_V}(ARVC_{v_i}, RSavai_{v_i}, \sigma_{v_i})$ and sends it to the initiator V, where $RSavai_{v_i}$ refers to the available resources that the vehicle v_i can provide in VCC, σ_{v_i} is the signature on $ARVC_{v_i}$ and $RSavai$ with the private key sk_{v_i} of DID_{v_i}, which demonstrates that the vehicle v_i is the owner of $ARVC_{v_i}$.

The initiator V decrypts and verifies all received response messages. The each verification process validates two signatures in the message $VCresp_{v_i} = E_{pk_V}(ARVC_{v_i}, \sigma_{v_i})$: one is the issuer's signature in $ARVC_{v_i}$ and the other is the vehicle's signature σ_{v_i}. After that, the initiator V selects n member vehicles (including itself) that have been verified successfully and meet resource requirements to establish a VC and numbers all the members in a cycle. That is, the VC members numbered 1 to n are assumed to be $v_1, v_2, ..., v_n$, we have v_{n+1} is v_1, and v_0 is v_n.

$VCmember = (DID_{v_1}, DID_{v_2}, ..., DID_{v_n}, \sigma_V^*)$, where DID_{v_i} can be obtained from $ARVC_{v_i}$ and σ_V^* is the signature of the initiator V on the member message with the private key sk_v of DID_V.

VC key negotiation.

Firstly, each VC member $v_i, i \in [1, n]$ randomly selects $s_i \in Z_q^\star$, computes $D_{i,i-1} = pk_{v_{i-1}}^{s_i}$, $D_{i,i+1} = pk_{v_{i-1}}^{s_i}$ and generates two messages $keyneg_{i\to i-1} = E_{pk_{v_{i-1}}}(DID_{v_i}, D_{i\to i-1}, \sigma_{i\to i-1}^\star)$, $keyneg_{i\to i+1} = E_{pk_{i+1}}(DID_{v_i}, D_{i\to i+1}, \sigma_{i\to i+1}^\star)$ to send to v_{i-1} and v_{i+1}, respectively, where $\sigma_{i\to i-1}^\star$ and $\sigma_{i\to i+1}^\star$ are the signatures of the member v_i on the two messages with the private key sk_{v_i} of DID_{v_i}.

Secondly, each VC member v_i can receive the message $keyneg_{i+1\to i} = E_{pk_{v_i}}(DID_{v_{i+1}}, D_{i+1\to i}, \sigma_{i+1\to i}^*)$ from the member v_{i+1} and the message $keyneg_{i-1\to i} = E_{pk_{v_i}}(DID_{v_{i-1}}, D_{i-1\to i}, \sigma_{i-1\to i}^*)$ from v_{i-1}. The member v_i decrypts the message and verifies the signatures $\sigma_{i+1\to i}^*$ and $\sigma_{i-1\to i}^*$ using the public key $pk_{v_{i+1}}$ of $DID_{v_{i+1}}$ and the public key $pk_{v_{i-1}}$ of $DID_{v_{i-1}}$, respectively. If successful, v_i computes the VC key parameters $B_{i+1} = (D_{i+1\to i})^{sk^{(-1)}{}_{v_i}} = g^{S_{i+1}}$ and $B_{i-1} = (D_{i-1\to i})^{sk^{(-1)}{}_{v_i}} = g^{S_{i-1}}$, then broadcasts $X_i = (B_{i+1}/B_{i-1})^{S_i} = g^{S_i(S_{i+1}-S_{i-1})}$.

Finally, each VC member v_i can obtain the set $\{X_1, X_2, ..., X_n\}$ and computes the VC key by the following equation.

$$\begin{aligned} K_{v_i} &= B_{i-1}^n s_i \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} ... X_{i-2} \\ &= g^{S_iS_{i+1}+S_{i+1}S_{i+2}+...+S_{i-2}S_{i-1}} \\ &= g^{S_1S_2+S_2S_3+...+S_nS_1} \end{aligned} \quad (1)$$

VC key confirmation. The VC initiator V encrypts the VC member set $VCmemberSet = \{DID_{v_1}, DID_{v_2}, ..., DID_{v_n}\}$ using the VC key K_V computed by the initiator V, signs the ciphertext using the private key sk_V of DID_V and broadcasts the VC key confirmation message $VCkeyConf = (E_{K_V}(VCmemberSet), \sigma_{sk_V}^{\star\star})$.

Each VC member v_i verifies the signature $\sigma_{sk_V}^{\star\star}$ and decrypts the $E_{K_V}(VCmemberSet)$ using the VC key K_{v_i}, then compares the decryption result and the member list in message $VCmember$, if they are the same, the two keys K_{v_i} and K_V are proved to be equal. Subsequently, the member v_i broadcasts a VC key confirmation response message, which only contains an expression of successful key agreement and a signature of the member v_i on the message.

If all members have sent the VC key confirmation response messages, the VC key negotiation is successful. That is, the VC key $K = K_{v_1} = K_{v_2} = ... = K_{v_n}$.

4.4 VC Information Release

After the VC key negotiation is successful, the VC initiator V generates a description of the VC:

$$VCdescrip = (VCmemberSet, RSavaiSet, CreateTime, EndTime, Position, other),$$

where $VCmemberSet$ gives the DIDs of all members and reveals the available resource set in the VC, which is collected according to the information in the VC member selection phase.

The initiator V broadcasts the ciphertext $E_K(VCdescrip, \tilde{\sigma}_V)$ to all the VC members, where $\tilde{\sigma}_V$ is the signature of the initiator V on $VCdescrip$ with the private sk_V of DID_V. Each VC member v_i can decrypt the message using the VC key and then uses the private key sk_{v_i} of DID_{v_i} to generate a signature $\tilde{\sigma}_{v_i}$ on $VCdescrip$ and sends back to the initiator V. The initiator V initiates a blockchain transaction and releases the VC information $VCinformation = (VCdescrip, \tilde{\sigma}_{v_1}, \tilde{\sigma}_{v_2}, ..., \tilde{\sigma}_{v_n})$ to the blockchain.

5 VC Service Access

A user searches the VC information on the blockchain and can send a VC access request to any VC member v_i in $VCmemberSet$, then the member v_i can broadcast the request message to the VC. The security of all communications between service providers and requesters can be ensured by their DIDs and the corresponding key pairs. In addition, the service records are released to the blockchain for post-audit and the digital currency mechanism inherent in the blockchain can provide incentives for VCC ecology.

6 VC Dynamic Management

6.1 Joining VC

When a vehicle u outside of VC wants to join the VC, it sends a request $VCresp_u = E_{pk_V}(ARVC_u, RSavai_u, \sigma_u)$ to the VC initiator V, then V performs the following verification:

- Verifies the resource certificate $ARVC_u$ and the signature σ_u, if successful, numbers the vehicle u v_{n+1}.
- Sends X_i, $i \in [2, n]$ of the VC to the new member v_{n+1}.
- Broadcasts the new member message $VCmember = (DID_{v_1}, DID_{v_2}, ..., DID_{v_n}, DID_{v_{n+1}}, \sigma^\star_V)$.

Next, all VC members update VC key by the following steps:

- The new member v_{n+1} randomly selects $S_{n+1} \in Z^\star_q$, computes $D_{n+1\to1} = (pk_{v_1})^S_{n+1}$ and $D_{n+1\to n} = (pk_{v_n})^S_{n+1}$, signs them using the private $sk_{v_{n+1}}$ of $DID_{v_{n+1}}$, and sends the messages $keyneg_{n+1\to1} = E_{pk_{v_1}}(DID_{v_{n+1}}, DID_{n+1\to1}, \sigma^\star_{n+1\to1})$ and $keyneg_{n+1\to n} = E_{pk_{v_n}}(DID_{v_{n+1}}, DID_{n+1\to n}, \sigma^\star_{n+1\to n})$ to the member v_1 and v_n, respectively.
- The member v_1 decrypts the message and verifies the signature $\sigma^\star_{n+1\to n}$, computes $B_{n+1} = (D_{n+1,1})^{sk_{v1}-1} = g^{S_{n+1}}$ and broadcasts $X'_1 = (B_2/B_{n+1})^{S_1}$ in VC. At the same time, The member v_1 also sends the message $keyneg_{1\to n+1} = E_{pk_{v_{n+1}}}(DID_{v_1}, D_{1\to n+1}, \sigma^\star_{1\to n+1})$ to the member v_{n+1}, where $D_{1\to n+1} = pk^{S_1}_{v_{n+1}}$. In the same way, the member v_n verifies the signature $\sigma^\star_{n+1\to n}$, computes $B_{n+1} = (D_{n+1\to n})^{sk^{-1}_n} = g^{S_{n+1}}$ and broadcasts $X'_n = (B_{n+1}/B_{n-1})^{S_n}$ in VC. At the same time, The member v_n also sends the message $keyneg_{n\to n+1} = E_{pk_{v_{n+1}}}(DID_{v_n}, D_{n\to n+1,\sigma^\star_{n\to n+1}})$ to the member v_{n+1}, where $D_{n\to n+1} = pk^{S_n}_{v_{n+1}}$.
- The member v_{n+1} decrypts the message and verifies the signatures $\sigma^\star_{1\to n+1}$ and $\sigma^\star_{n\to n+1}$, computes $B_1 = (D_{1\to n+1})^{sk_{v_{n+1}}-1} = g^{S_1}$, $B_n = (D_{n,n+1})^{sk_{v_{n+1}}-1} = g^{S_n}$ and broadcasts $X_{n+1} = (B_1/B_n)^{S_{n+1}}$.
- After receiving X'_1, X'_n, X'_{n+1}, each member v_i in VC updates the VC key according to the equation and $X'_1, X_2, X_3, ..., X'_n, X_{n+1}$.

$$K'_i = B^{N_{S_i}}_{i-1} \cdot X^{N-1}_i \cdot X^{N-2}_{i+1} ... X_{i-2} \tag{2}$$

- All members performs the VC key confirmation process and obtains the new key of the VC $K_{new} = K'_i$, $i \in [1, n+1]$. Now, the VC has $n+1$ members $v_1, v_2, ..., v_n, v_{n+1}$.

6.2 Exiting VC

When the member v_j wants to exit the VC, the following is done.

- v_j broadcasts the exit request message ciphertext encrypted by the VC key K

$$EXITreq_{v_j} = E_K(DID_{v_j}, (pk_{v_{j+1}} \cdot pk_{v_{j-1}})^{S_j}, \hat{\sigma}_{v_j})$$

to the VC, where $\hat{\sigma}_{v_j}$ is the signature of v_j on the request message.

- After receiving the message $EXITreq_{v_j}$, the member v_{j+1} computes $D_{j+1\to j-1} = (pk_{v_{j-1}})^{S_{j+1}}$, signs $D_{j+1\to j-1}$ using the private $sk_{v_{j+1}}$ of $DID_{v_{j+1}}$ and sends the message $keyneg_{j+1\to j-1} = (DID_{v_{j+1}}, D_{j+1\to j-1}, \sigma^\star_{j+1\to j-1})$ to the member v_{j-1}.

- The member v_{j-1} verifies the signature $\sigma^{\star}_{j+1\to j-1}$, then computes and broadcasts $D = (D_{j+1\to j-1})^{sk_{j-1}-1}$. Subsequently, all members update their VC keys according to the equation.

$$K_{new} = \frac{D}{(pk_{v_{j+1}} \cdot pk_{v_{j-1}})^{S_j}} K_{old} \tag{3}$$

- All members perform the VC key confirmation process. If successful, the initiator V releases the new VC member message: $VCmember = (DID_{v_1}, DID_{v_2}, ..., DID_{v_{j-1}}, DID_{v_{j+1}}, ..., DID_{v_n}, \sigma^{\star}_V)$.

7 Security and Performance Analysis

7.1 Security Analysis

The security of the above scheme includes the following 7 aspects.

The security of the VC key negotiation protocol. The literature [3] has discussed the security of the key negotiation protocol, and proved that the protocol is secure as long as the broadcast message in the negotiation process is verified by using a secure authentication method. This paper uses the digital signature to authenticate the broadcast message. As long as the signature algorithm is secure, the authentication scheme is secure, so the VC key negotiation process is secure.

Preventing resource information forgery. In the proposed scheme, the management departments DMVs issue the verifiable resource certificates ARVCs for the vehicles participating in the VCC, and any vehicle can verify the authenticity of ARVCs, which can prevent malicious vehicles from arbitrarily forging their own resource information to disrupt the normal execution of the VCC or cheat.

Preventing impersonation. Preventing direct impersonation. The polices that the attacker w directly impersonates legitimate vehicle v may include impersonating vehicle v to gain the qualification to participate in the VCC in VC member selection phase or impersonating legitimate vehicle v to participate in VC key negotiation in the VC key negotiation phase.

In the proposed scheme, the legitimate vehicle v that wants to join the VCC first needs to be registered with the DMV with its real identity, and the content of the registration request is signed by the vehicle v with its own DID private key. The DMV is trusted. Therefore, the fake vehicle w cannot impersonate v to register and obtain a legitimate ARVCs. At the same time, each ARVC is signed by DMV, thus the security of the signature algorithm makes it impossible for a malicious vehicle w to forge a legitimate resource certificate. In addition, an ARVC includes a DID public key of the legal vehicle v, and when using an ARVC to join VCC, a legal signature on the certificate with corresponding DID private key is required. That is to say, even if the malicious vehicle w intercepts an ARVC, it cannot use the certificate because it does not have a corresponding private key to generate a legal signature. Therefore, the malicious vehicle w cannot impersonate v to gain the qualification to participate in VCC.

The security of the VC key negotiation protocol ensures that the malicious vehicles cannot impersonate the legal vehicles to participate in the VC key negotiation.

Preventing man-in-the-middle attack. In all communication process, the digital signature is used to provide the authentication service. There-fore, the proposed scheme can resist man in the middle attack.

Information source and integrity certification. Messages sent by all entities in the scheme are signed to guarantee the authenticity of message sources and the integrity of messages.

Forward security. The VC information that the newly entered vehicle v_{n+1} can get includes: previous message $VC_{estab} = (VCreqquirments, ARVC_V, \sigma_V)$, $VCmember = (DID_{v_1}, DID_{v_2}, ..., DID_{v_n}, \sigma^{\star}_V)$ and X_i, $i \in [1, n]$. According to the calculation method of VC key $K_i = B_{i-1}^{nS_i} \cdot X_i^{n-1} \cdot X_{i+1}^{n-2} ... X_{i-2}$ and the security of VC key negotiation protocol, it is obviously impossible for the vehicle v_{n+1} to calculate the previous VC key unless it has a random value S_i selected by a previous VC member or can solve the CDH problem.

Privacy protection. The proposed scheme is based on the block-chain with DIDs, and all vehicles are identified with anonymous DIDs and all sensitive information is encrypted with DID keys. Therefore, the identity privacy and data privacy of vehicles are protected. Furthermore, a vehicle requests multiple ARVCs with different DIDs from DMV, and each ARVC and corresponding DID can be disposable to make different VCC processes unlinked, thus the position privacy of the vehicle can be protected.

7.2 Performance Analysis

In this section, we compare our scheme with scheme in [45] to show the performance of our scheme.

7.2.1 Computation Performance

Table 2 shows the comparison of the computation overhead in vehicle registration stage and VC key negotia-

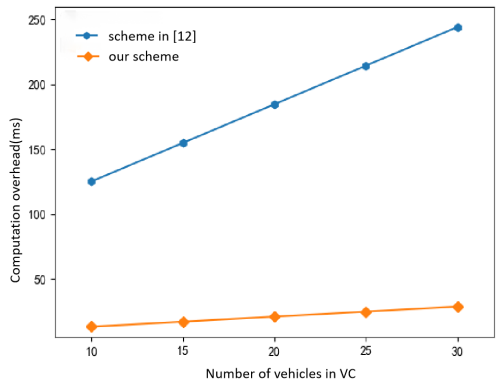


Figure 2: Comparison of the computation overhead

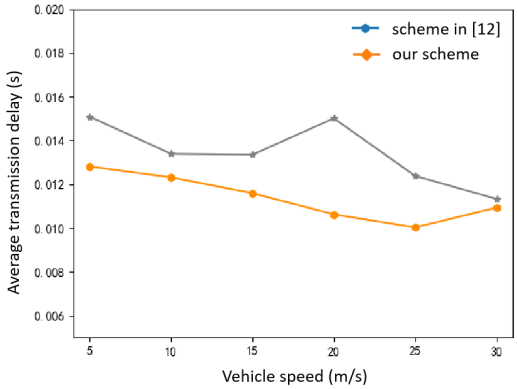


Figure 3: The relationship between average transmission delay and the speed of vehicle

tion stage between the proposed scheme and the scheme in [45].

The T_{bp} is a bilinear pair operation, T_{bpm} is the multiplication operation of a bilinear pair, T_{bsm} is the point multiplication operation of a bilinear pair, T_{ex} is a multiplication operation related to the ECC, T_{esm} is a point multiplication operation related to the ECC, T_{mtp} is Map-to-Point hashing operation, T_h is an one-way hash operation, T_{ECDSA} is an ECDSA signature and verification operation, $T_{ElGamal}$ is an ElGamal encryption and decryption operation.

Figure 2 shows the comparison of the computation overhead of the two schemes as the VC size changes, and it demonstrates the computation overhead of our scheme is significantly better than the scheme in [45]. The reason for this result is that there is no bilinear operation in our scheme.

7.2.2 Communication Delay

We compare the average communication delay of the two schemes through simulation analysis. The simulation uses the event-based net-work simulator Omnet ++ [31], the traffic simulation soft-ware Sumo [34] and the network simulation framework Veins [40], and the experimental environment is Intel(R) Core(TM) i5-8265U CPU@16GB memory and Windows 10 operating system with 1.80Ghz main frequency. The simulation parameters of the experiment are shown in Table 3, and the vehicle exit position was random in the experiment.

The equation (4) gives the calculation method of the average end-to-end delay of the message between the receiver and the sender [**?**]. In order to observe the relationship between the speed of vehicles and the end-to-end delay, the number of vehicles is set as 20, and the data packet length is set as 1088B in our scheme and 976B in the scheme in [45]. T_s^i and T_r^i are the sending time and receiving time of the message, respectively, and $T_r^i - T_s^i$ indicates the time of a one-way transmission between the recipient and the sender. In addition, n is the number of vehicles, and N_i is the number of messages received by the vehicles.

$$AvgDelay = Avg(\Sigma_{j=1}^{n} Avg(\Sigma_{i=1}^{N_i}(T_r^i - T_s^i))) \quad (4)$$

Figure 3 shows the relationship between the average end-to-end delay and the vehicle speed. As can be seen from the figure, when the vehicle speed is low, the physical distance between the vehicles is relatively far in the VC key negotiation, so the average end-to-end delay of the two schemes is relatively larger. Since the total computation time of our scheme is smaller than that of the scheme in [45], and the average end-to-end delay of our scheme is slightly lower than that of the scheme in [45]. However, with the increasing speed, the physical distance between the vehicles is relatively close, and the packet size in our scheme is larger than that in the scheme in [45], and thus the average end-to-end delay of our scheme is slightly higher than that of the scheme in [45].

8 Conclusion

The privacy and security problems are the main challenges that VCC is facing, this paper combines blockchain and DID technologies to provide a decentralized VCC solution with good privacy and security features. The decentralized characteristics of blockchain and DID make them easy to integrate. Blockchain provides a decentralized trusted platform for DID and the DID's internal cryptography mechanism, combined with the verifiable certificate model, can provide anonymous authentication capabilities, which can greatly expand blockchain based applications. Obviously, this solution is very suitable for the VANET environment with high privacy and security requirements. Future work can further study the deep integration of blockchain, DID and VANET and practicality.

Table 2: Comparison of the computation overhead

Scheme	Vehicle registration stage	VC key negotiation stage
[45]	$(n+4)T_{bpm}+(n+1)T_{tmp}$	$4T_{bp}+T_{bpm}$
Our scheme	$2T_{ECDSA}+2T_{ELGamal}+T_{ex}$	$(5+n)T_{ex}+(n-1)T_{esm}+2T_{ECDSA}$

Table 3: Simulation parameters

Parameter	Value
Simulation area	$20000 \times 20000\ m^2$
maximum jamming range	$2600\ m$
transmission range	$50\ Mw$
data rate	$6\ Mbps$
sensing capability	$-89\ dBm$
simulated time	$100\ s$

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 61702067) and in part by Chongqing Natural Science Foundation (No. cstc2020jcyj-msxmX0343).

References

[1] S. Anwar, V. K. Shukla, S. S. Rao, *et al.*, "Framework for financial auditing process through blockchain technology, using identity based cryptography," *Sixth HCT Information Technology Trends*, IEEE, pp. 099-103, 2019.

[2] A. Boukerche, R. E. De Grande, "Vehicular cloud computing: Architectures, applications, and mobility," *Computer Networks*, vol. 135, pp. 171-189, 2018.

[3] M. Burmester, Y. A. Desmedt, *Secure and efficient conference key distribution system*, Springer-Verlag, pp. 275-286, 1995.

[4] E. F. Cahyadi, C. Damarjati, M. S. Hwang, "Research on identity-based batch verification schemes for security and privacy in VANETs", *Journal of Electronic Science and Technology*, vol. 20, no. 3, pp. 1-19, 2022.

[5] E. F. Cahyadi, M. S. Hwang, "A comprehensive survey on certificateless aggregate signature in vehicular ad hoc networks", *IETE Technical Review*, vol. 39, no. 6, pp. 1265-1276, 2022.

[6] E. F. Cahyadi, M. S. Hwang, "An improved efficient anonymous authentication with conditional privacy-preserving scheme for VANETs", *Plos One*, vol. 16, no. 9, 2021.

[7] E. F. Cahyadi, M. S. Hwang, "An improved efficient authentication scheme for vehicular ad hoc networks with batch verification using bilinear pairings", *International Journal of Embedded Systems*, vol. 15, no. 2, pp. 139-148, 2022.

[8] E. F. Cahyadi, M. S. Hwang, "A lightweight BT-based authentication scheme for illegal signatures identification in VANETs", *IEEE Access*, vol. 10, pp. 133869-133882, 2022.

[9] P. Y. Chang, M.S. Hwang, C.C. Yang, "A blockchain-based traceable certification system", in *Security with Intelligent Computing and Big-data Services*, pp. 363-369, 2018.

[10] L. Chen, Q. Fu, Y. Mu, L. Zeng, F. Rezaeibagha, M. S. Hwang, "Blockchain-based random auditor committee for integrity verification", *Future Generation Computer Systems*, vol. 131, pp. 183-193, 2022.

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

[12] Y. H. Chen, L. C. Huang, I. C. Lin, M. S. Hwang, "Research on blockchain technologies in bidding systems", *International Journal of Network Security*, vol. 22, no. 6, pp. 897-904, 2020.

[13] J. Cui, X. Zhang, H. Zhong, *et al.*, " Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1654-1667, 2019.

[14] P. Dunphy and F. A. Petitcolas, "A first look at identity management schemes on the blockchain," *IEEE Security & Privacy*, vol. 16, no. 4, pp. 20-29, 2018.

[15] S. Guo, X. Hu, S. Guo, *et al.*, "Blockchain meets edge computing: A distributed and trusted authentication system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1972-1983, 2019.

[16] M. R. Jabbarpour, A. Marefat, A. Jalooli, *et al.*, "Could-based vehicular networks: a taxonomy, survey, and conceptual hybrid architecture," *Wireless Networks*, vol. 25, no. 1, pp. 335-354, 2019.

[17] J. Q. Ji, Y. Y. Yao, X. L. Chang, "A review on the security of vehicular cloud computing," *Cyberspace Security*, vol. 11, no. 6, pp. 50-56, 2020.

[18] Q. Jiang, J. Ni, J. Ma, X. Yang Land Shen, "Integrated authentication and key agreement framework for vehicular cloud computing," *IEEE Network*, vol. 32, no. 3, pp. 28-35, 2018.

[19] N. Kenyereye, Y. Park, K. H. Rhee, "Secure vehicle traffic data dissemination and analysis protocol in vehicular cloud computing," *The Journal of Supercomputing*, vol. 74, no. 3, pp. 1024-1044, 2018.

[20] C. Li, S. Ji, X. Zhang, H. Wang, D. Li, H. Liu, "An effective and secure key management protocol for message delivery in autonomous vehicular clouds," *Sensors*, vol. 18, no. 9, pp. 2896, 2018.

[21] T. Limbasiya, D. Das, "SearchCom: Vehicular cloud-based secure and energy-efficient communication and searching system for smart transportation," in *Proceedings of the 21st International Conference on Distributed Computing and Networking*, 2020.

[22] T. Limbasiya, D. Das, "Lightweight secure message broadcasting protocol for vehicle-to-vehicle communication," *IEEE Systems Journal*, vol. 14, no. 1, pp. 520-529, 2020.

[23] T. Limbasiya, D. Das, "Secure message confirmation scheme based on batch verification in vehicular cloud computing," *Pysical Communication*, vol. 34(C), pp. 310-320, 2019.

[24] C. Y. Lin, L. C. Huang, Y. H. Chen, M. S. Hwang, "Research on security and performance of blockchain with innovation architecture technology", *International Journal of Network Security*, vol. 23, no. 1, pp. 1-8, 2021.

[25] C. W. Liu, W. F. Hsien, C. C. Yang, 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.

[26] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing", *International Journal of Network Security*, vol. 18, no. 4, pp. 650-666, 2016.

[27] A. Masood, D. S. Lakew, S. Cho, "Security and privacy challenges in connected vehicular cloud computing," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2725-2764, 2020.

[28] MCC Forum, *Discover the World of Mobile Cloud Computing*, Aug. 15, 2023. (`http://http://www.mobilecloudcomputingforum.com`)

[29] P. M. Mell, T. Grance, *The NIST Definition of Cloud Computing*, National Institute of Standards & Technology, Tech. Rep. SP 800-145, 2011.

[30] S. Olariu, M. Eltoweissy, M. Younis, "Towards autonomous vehicular clouds," *ICST Transactions on Mobile Communication*, vol. 11, pp. 1-11, 2011.

[31] Omnet^{++}, *Discrete Event Simulator*, Nov. 29, 2019. (`https://omnetpp.org/`)

[32] Z. P. Peng, M. L. Chiang, I. C. Lin, C. C. Yang, M. S. Hwang, "CBP2P: Cooperative electronic bank payment systems based on blockchain technology", *Journal of Internet Technology*, vol. 23, no. 4, pp. 683-692, 2022.

[33] J. Shao, G. Wei, "Secure outsourced computation in connected vehicular cloud computing," *IEEE Network*, vol. 32, no. 3, pp. 36-41, 2018.

[34] Simulation of Urban Mobility, *SUMO User Documentation*, Nov. 29, 2019. (`https://sumo.dlr.de/docs/index.html`)

[35] M. Sporny, D. Longley, D. Chadwic, *Verifiable Credentials Data Model v2.0*, W3C Working Draft, 21 Nov. 2022. (`https://w3c.github.io/vcdata-model/#what-is-a-verifiable-credential`)

[36] M. Sporny, D. Longley, M. Sabadello, *et al.*, *Decentralized Identifiers (DIDs) v1.0*, W3C Recommendation, 19 July 2022. (`https://www.w3.org/TR/2022/REC-did-core-20220719/`)

[37] F. Tang, G. Ling, J. Shan, "Additive homomorphic encryption schemes based on SM2 and SM9," *Journal of Cryptologic Research*, vol. 9, no. 3, pp. 1-15, 2022.

[38] F. Tang, S. Ma, Y. Xiang, *et al.*, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE Access*, vol. 7, pp. 41678-41689, 2019.

[39] F. Tang, J. Pang, K. Cheng, *et al.*, "Multiauthority traceable ring signature scheme for smart grid based on blockchain," *Wireless Communications and Mobile Computing*, 2021.

[40] veins, *The Open Source Vehicular Network Simulation Framework*, Nov. 29, 2019. (`https://veins.car2x.org`)

[41] L. Wang, D. Zheng, R. Guo, C. Hu, and C. Jing, "A blockchain-based privacy-preserving authentication scheme with anonymous identity in vehicular networks," *International Journal of Network Security*, vol. 22, no. 6, pp. 981-990, 2020.

[42] X. Wang, Q. Chen, Z. Peng, Y. Wang, "An efficient and secure identity-based conditional privacy-preserving authentication scheme in VANETs," *International Journal of Network Security*, vol. 24, no. 4, pp. 661-670, 2022.

[43] P. Windley, R. D. Sovrin, *Sovrin^{TM}: A Protocol and Token for Self-Sovereign Identity and Decentralized Trust*, A White Paper from the Sovrin Foundation, Version 1.0, Jan. 2018.

[44] J. Y. Zhang, "Research on security authentication and privacy protection mechanism of vehicular cloud computing," Beijing Jiaotong University, 2018.

[45] L. Zhang, X. Meng, K. R. Choo, Y. Zhang, F. Dai, "Privacy-preserving cloud establishment and data dissemination scheme for vehicular cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 634-647, 2020.

Biography

Zaishuang Liu is currently a senior engineer of the China Electronics Technology Cyber Security Co.,Ltd. His research interest is blockchain.

Xiaoxu Ma is currently a senior engineer of the China Electronics Technology Cyber Security Co.,Ltd. His research interest is blockchain.

Jian Bai is currently a senior engineer of the China Electronics Technology Cyber Security Co.,Ltd. His research interest is blockchain.

Min Xiao is currently a professor of the College of Cyberspace Security and Law, Chongqing University of Posts and Telecommunications. Her research interests are cryptography and blockchain.

Fei Tang 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, blockchain, and privacy protection.

An Efficient Sine Cosine Algorithm for Global Complex Optimization Problems

Jing-Sen Liu[1,2], Fang-Yuan Zhao[1,2], and Ping Hu[2]
(Corresponding author: Ping Hu)

Henan International Joint Laboratory of Intelligent Network Theory and Key Technology
Henan University, Kaifeng 475004, China[1]
College of Software, Henan University, Kaifeng 475004, China[2]
Email: huping@vip.henu.edu.cn
(Received Dec. 12, 2022; Revised and Accepted Aug. 5, 2023; First Online Aug. 25, 2023)

Abstract

Optimization algorithms are widely used in the field of network security optimization. The Sine Cosine Algorithm (SCA) is an effective algorithm for solving global complex optimization problems. However, problems remain, such as insufficient solution accuracy, slow convergence speed, and difficulty jumping out of local extreme values. To improve the optimization performance and application ability of the SCA and apply it better to solve complex optimization problems in network information security, three improved strategies, namely Elite leadership, Quadratic interpolation optimization, and Self-feedback memory refresh, are introduced into the Sine Cosine Algorithm (EQSSCA). The elite leadership strategy first coordinates the algorithm's global exploration and local mining capabilities. Then, the quadratic interpolation optimization strategy is adopted to improve the algorithm's solution accuracy and enrich the population's diversity. Finally, a self-feedback memory refresh strategy is introduced to enhance the population's capability to evade local extremes and improve the algorithm's convergence rate. In addition, the EQSSCA and SCA are proven consistent in terms of time complexity by theoretical analysis. To evaluate the proposed algorithm's optimization capability, the optimization accuracy, difference significance, and convergence curves of EQSSCA and four high-performance comparison algorithms are tested and analyzed on various dimensions of the CEC2017 test suite. The test results indicate that the proposed three strategies can effectively enhance SCA's solution accuracy, robustness, adaptability, and effectiveness in solving global optimization problems. And the proposed algorithm is superior to the other four comparison algorithms.

Keywords: Elite Leadership; Global Complex Optimization Problem; Quadratic Interpolation Optimization; Self-Feedback Memory Refresh; Sine Cosine Algorithm

1 Introduction

Optimization is a very active and extensive research direction in the current era, which aims to solve some large-scale and global complex optimization problems in real life. The traditional methods for solving optimization problems, such as the Newton method, conjugate gradient method, and branch and bound method, have the advantages of high accuracy and complete theoretical basis. However, they also have disadvantages, such as high computational complexity and slow convergence speed. Therefore, for many large-scale and high complexity optimization problems, traditional optimization methods have been difficult to find an effective solution in a reasonable time. Nevertheless, various meta-heuristic optimization algorithms inspired by physical or biological behaviors in nature, have the characteristics of simple operation, flexible mechanism, and easy implementation. They can quickly solve these large-scale and complex problems and obtain satisfactory solutions. Classical meta-heuristic algorithms include Particle Swarm Optimization (PSO), which is motivated by the birds' foraging activities; Genetic Algorithm (GA), which draw on the natural evolutionary process of living organisms, etc. At the same time, many new algorithms with different mechanisms and superior performance have emerged in recent years. For example, Chimp Optimization Algorithm (ChOA) [26], which simulates the hunting process of chimpanzee groups; Gradient-Based Optimizer (GBO) [2], inspired by the gradient-based Newton's method; Arithmetic Optimization Algorithm (AOA) [3], which is based on the four mixed operations in arithmetic; Snake Optimizer (SO) [19], which imitates the special mating behavior of snakes; Equilibrium Optimizer (EO) [11], which is motivated by the dynamic mass balance of the control volume, etc. These algorithms have fast convergence speed and high optimization-seeking accuracy, which provide new ideas and design solutions for solving large-scale global complex problems. They have been successfully ap-

plied to many fields such as path planning [32], traveler problems [14, 31], cryptanalysis [20, 36], intrusion detection [38], and blockchain [6, 18].

The need to seek optimal solutions is prevalent in information security technologies, and these needs are computationally intensive and computationally complex. Metaheuristic optimization algorithms are one of the effective methods to solve complex optimization problems in network security. At present, a number of optimization algorithms have been applied in this area, such as Alzaqebah [4] et al. proposed an improved gray wolf optimization algorithm to enhance the identification and detection ability of intrusion detection system on network attacks. Rizk-Allah [36] et al. proposed a hybrid method incorporating the particle swarm algorithm and equilibrium optimizer and applied it to cryptanalysis problems, and the test results demonstrated that the proposed method improved the accuracy of cryptanalysis. Kan [23] et al. proposed an intrusion detection method for the Internet of Things based on adaptive particle swarm optimization convolutional neural network, and verified the effectiveness of the proposed algorithm through simulation experiments. To enhance the performance of blockchain networks, Cai [6] et al. proposed a multi-objective optimization algorithm based on a dynamic reward and punishment mechanism. The experimental results showed that the proposed algorithm significantly improved the throughput and effectiveness of the sharding. In response to the problem of high false alarm rate of intrusion detection system, Dwivedi [10] et al. proposed a hybrid method combining the feature selection algorithm and grasshopper optimization algorithm. The proposed method was experimentally proven to be effective in improving the detection accuracy of intrusion detection systems. Ponmalar [34] et al. applied the chaotic game optimization algorithm combined with ensemble support vector machine to the data processing of intrusion detection system. The experimental results showed that compared with several existing algorithms, the proposed algorithm improved the accuracy of intrusion classification significantly. These algorithms have achieved good results in solving network security-related problems, but there is still room for further improvement in the solution quality of the algorithms. Therefore, there is still a need to explore algorithms with stronger search capability, higher solution accuracy, and better problem adaptability to solve network information security optimization problems.

Sine Cosine Algorithm (SCA) [33] is a newmetaheuristic intelligent algorithm proposed by Australian scholar Mirjalili in 2016. By creating multiple random candidate solutions, SCA uses a mathematical model based on sine and cosine to make the candidate solutions oscillate toward the optimal solution. The SCA has the characteristics of superior mechanism, simple structure, and excellent solution performance, so it has become one of the important algorithms in the field of evolutionary computing in recent years. At present, it has been successfully applied to many fields, such as real-time task scheduling in multiprocessor systems [1], image copyright protection [9], photovoltaic pumping system [24], hydropower system [13], economic power generation dispatching [25], and so on.

However, based on the NFL (no-free lunch) [5] theorem, we can learn that there does not exist one algorithm for solving various optimization problems. Thus, similar to other swarm intelligence algorithms, SCA also has problems such as unstable solutions, sometimes low optimization accuracy, and low applicability. Therefore, many scholars have conducted in-depth research and improvement on the shortcomings of the SCA algorithm. For example, Zhou [41] et al. integrated three improved strategies into the sine cosine algorithm, and proved the superiority of the improved algorithm through simulation experiments. Khokhar [27] et al. proposed a chaotic sine cosine algorithm based on two-dimensional sine logic mapping, which effectively improved the algorithm's solving precision and convergence rate. Guo [15] et al. proposed a sine cosine algorithm integrating an elite chaotic search mechanism, which better coordinated the exploration and exploitation of the algorithm and improved the solution stability. Chen [8] et al. presented a modified sine-cosine algorithm combining three mechanisms, and demonstrated through simulation experiments that this algorithm can significantly improve the exploration and usability of SCA. Gupta [16] et al. proposed a memory-guided sine cosine algorithm. Through function testing and evaluation of constrained engineering problems, this algorithm was demonstrated to have good search efficiency and solution accuracy. Aiming at the problem that SCA is prone to prematurity, Saha [37] et al. proposed an adaptive sine cosine algorithm based on multi-population, which enhanced the SCA's ability to avoid local optima and effectively improved the solution quality of the algorithm. To alleviate the problem of low accuracy of the SCA, Wei [39] et al. proposed an improved sine cosine algorithm based on a dynamic classification strategy, which effectively improved the convergence speed and stability of the SCA. Li [28] et al. proposed a dimension-by-dimension dynamic sine cosine algorithm, and the test of high-dimensional functions verified that the algorithm has good robustness. Feng [12] et al. proposed an improved sine cosine algorithm combining multiple strategies, such as contrastive learning, adaptive evolution, etc. And through the evaluation of numerical optimization problems, it was proved that the algorithm has good solution efficiency and convergence speed. Kale [22] et al. proposed several modified sine cosine algorithms that effectively enhance the SCA's optimization capability. Raut [35] et al. proposed an improved sine cosine algorithm incorporating Lvy flight, which nicely balanced the exploration and exploitation in the evolutionary process and enhanced the SCA algorithm's ability to escape from local extrema. Hamad [17] et al. proposed a Q-learning embedded sine cosine algorithm and demonstrated that it has a fast convergence rate using several functions and three engineering constraint problems.

These improvements have enhanced the optimization and application capabilities of SCA in their respective fields. However, the capability of SCA to get rid of local extremes, and its solution effectiveness and adaptability for global complex problems still need to be enhanced to a greater extent. To solve the optimization problem more efficiently, further improve the optimization performance and solution quality of SCA, and broaden its application area in network security, this paper proposes a sine cosine algorithm (EQSSCA) based on elite leadership, quadratic interpolation optimization, and self-feedback memory refresh. Firstly, the elite leadership mechanism is designed to enrich the population diversity, accelerate the convergence speed, and enhance the algorithm's capability to adjust global exploration and local mining. Secondly, the quadratic interpolation optimization mechanism is added to enhance the algorithm's exploitation ability, and increase the diversity of the population. Then, the self-feedback memory refresh mechanism is used to avoid the algorithm from being trapped in local extremes and speed up the convergence speed of the algorithm to the optimal solution. Through theoretical analysis, this paper proves that the time complexity of EQSSCA is the same as that of SCA, and the algorithm's execution efficiency does not decrease with the addition of three improvement mechanisms. To validate the proposed algorithm's effectiveness, EQSSCA is tested and analyzed with four superior performance comparison algorithms on CEC2017 test suite from different dimensions, respectively. The comparison results of numerical results, difference significance and convergence curves show that the convergence capability, optimization accuracy, as well as solution stability of EQSSCA significantly outperform the other four excellent comparative algorithms.

The remaining part of this paper is organized as follows. Section 2 mainly reviews the basic sine cosine algorithm. Section 3 gives a detailed description of the improved algorithm EQSSCA. Section 4 analyzes the time complexity of the SCA and EQSSCA. Section 5 selects CEC2017 test functions to verify the superior global optimization performance of EQSSCA. Finally, the work of this paper is summarized and presented in Section 6.

2 The Sine Cosine Algorithm

The basic sine cosine algorithm flow is as follows.

Step1: Initialize the population. According to the upper and lower bounds of each dimension of the solution space, the initial position $X_{i,j}(i = 1, 2, ..., N, j = 1, 2, ..., D)$ of the individual is randomly generated in the space,

$$X_{i,j} = rand() \times (ub(j) - lb(j)) + lb(j), \tag{1}$$

where, N is total number of individuals; D is the space dimension; $ub(j)$ and $lb(j)$ denote the maximum and minimum values of the j_th dimension, respectively.

Step2: Calculate the fitness value of each individual according to the objective function, and find out the current optimal solution.

Step3: Update individual position,

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} + r_1 \times \sin(r_2) \times \left| r_3 P_j^t - X_{i,j}^t \right|, r_4 < 0.5 \\ X_{i,j}^{t} + r_1 \times \cos(r_2) \times \left| r_3 P_j^t - X_{i,j}^t \right|, r_4 \geq 0.5 \end{cases}, \tag{2}$$

among them, t is the current iteration number; r_2, r_3, r_4 are uniformly distributed stochastic numbers, $r_2 \in (0, 2\pi)$, $r_3 \in [0, 2]$, $r_4 \in (0, 1)$; and $X_{i,j}^t$ is the j_th dimensional position of current individual i at the t_th iteration; P_j^t is the global optimal individual position in the j_th dimension for the t_th iteration. The parameter r_1 decreases linearly with the increasing number of iterations, and its expression is as follows:

$$r_1 = a - a \cdot \frac{t}{Max\_iteration}, \tag{3}$$

where, $Max\_iteration$ is the maximum number of iterations and $a = 2$.

Step4: Perform boundary processing on the updated individual. Then, the fitness value of the individual is calculated according to the objective function, and the current global optimal solution is updated.

Step5: Judge if the iteration count reaches $Max\_iteration$. If yes, output the result; otherwise, go to Step3 to continue the iteration.

3 The Proposed Algorithm EQSSCA

3.1 Elite Leadership

When SCA explores the problem space, the population will usually gather in a small area at the late iteration, leading to a tendency for SCA to be trapped in local extremes when solving complex optimization problems with multiple extremums. In response to these shortcomings, we propose an elite leadership strategy, which will update the candidate solutions together with the position update strategy of the basic SCA. This strategy not only updates the candidate solution based on the current optimal solution, but also randomly selects an individual from the population for differential search. Therefore, the activity of the population is improved, and the local search ability of the algorithm is enhanced. The specific equation is as follows:

$$\bar{X}_{i,j} = \begin{cases} P_j^t + r_1 \times \sin(r_2) \times \left(X_{m,j}^t - X_{i,j}^t \right), r_4 < 0.5 \\ P_j^t + r_1 \times \cos(r_2) \times \left(X_{m,j}^t - X_{i,j}^t \right), r_4 \geq 0.5 \end{cases}, \tag{4}$$

where, $\bar{X}_{i,j}$ is the j_th dimension position of the current individual i after this update; P_j^t is the position of the current optimal solution in the j_th dimension; $X_{i,j}^t$ is the j_th dimensional position of current individual i at the t_th iteration; X_m^t is an individual chosen randomly in the current population, and $m \neq i$.

To coordinate the exploration and exploitation of SCA, a dynamic transition probability A with a decreasing trend is set in EQSSCA. The transition probability A is used to determine whether the algorithm adopts the basic update equation or the proposed improved equation to update the current individual. The expression of transition probability A is as follows:

$$A = A_{\min} + (A_{\max} - A_{\min}) \cdot \left(1 - \frac{t}{Max\_iteration}\right) \cdot rand(), \quad (5)$$

where, $Max\_iteration$ is the maximum number of iterations, and t is the current iteration number. $A_{\max}$ and $A_{\min}$ are the maximum and minimum of the transition probability A respectively. After repeated tests, when $A_{\max} = 1$, $A_{\min} = 0.4$, the optimization effect is the best. $rand()$ is utilized to produce a random number evenly spread between (0,1), so that the transition probability A shows a decreasing trend, but there is also some randomness. This randomness reduces the risk of the algorithm converging prematurely.

The improved individual position update process is shown in Code 1. Where, the switching variable $Q \in (0, 1)$ is a random number with uniform distribution.

Code 1 The elite leadership strategy

```
if Q < A
  for j = 1 : D
X_{i,j}^{t+1} = { X_{i,j}^t + r1 × sin(r2) × |r3 P_j^t − X_{i,j}^t| , r4 < 0.5
               { X_{i,j}^t + r1 × cos(r2) × |r3 P_j^t − X_{i,j}^t| , r4 ≥ 0.5
  end for
else
  for j = 1 : D
X_{i,j}^{t+1} = { P_j^t + r1 × sin(r2) × |X_{m,j}^t − X_{i,j}^t| , r4 < 0.5
               { P_j^t + r1 × cos(r2) × |X_{m,j} − X_{i,j}^t| , r4 ≥ 0.5
  end for
end if
```

3.2 Quadratic Interpolation Optimization

Quadratic interpolation, as a local exploration operation, is based on the fundamental concept of continuously fitting a quadratic curve with three known points in the search space. And gradually use the extreme points of the quadratic curve to approach the minimum of the research problem.

Assume that $\bar{X}_i=(\bar{X}_{i,1}, \bar{X}_{i,2}, ..., \bar{X}_{i,D})$, $\bar{X}_y=(\bar{X}_{y,1}, \bar{X}_{y,2}, ..., \bar{X}_{y,D})$, $\bar{X}_z=(\bar{X}_{z,1}, \bar{X}_{z,2}, ..., \bar{X}_{z,D})$ and are three known individuals in the population, and their fitness values are $f(\bar{X}_i)$, $f(\bar{X}_y)$ and $f(\bar{X}_z)$, respectively. These three known individuals are used to generate a new individual X_i' through quadratic interpolation:

$$\begin{aligned} \varphi &= [\bar{X}_{y,j}^2 - \bar{X}_{i,j}^2] \times f(\bar{X}_z) + \\ &\quad [\bar{X}_{z,j}^2 - \bar{X}_{y,j}^2] \times f(\bar{X}_i) + [\bar{X}_{i,j}^2 - \bar{X}_{z,j}^2] \times f(\bar{X}_y), \\ \omega &= 2[(\bar{X}_{y,j} - \bar{X}_{i,j}) \times f(\bar{X}_z) + \\ &\quad (\bar{X}_{z,j} - \bar{X}_{y,j}) \times f(\bar{X}_i) + (\bar{X}_{i,j} - \bar{X}_{z,j}) \times f(\bar{X}_y)] \\ X'_{i,j} &= \frac{\varphi}{\omega}, \end{aligned} \quad (6)$$

where $\bar{X}_{i,j}$, $\bar{X}_{y,j}$, and $\bar{X}_{z,j}$ respectively stand for the components of the known points $\bar{X}_i$, $\bar{X}_y$, and $\bar{X}_z$ in the j_th dimension, and $j = 1, 2, ...D$, $i = 1, 2, ..., N$. y and z are the cyclic successors of i, that is, for $i = 1, 2, ..., N-2$, $y = i+1$, and $z = i+2$; while for i=$N-1$, $y = N$, and $z = 1$; and when $i = N$, $y = 1$, and $z = 2$.

In this paper, the algorithm EQSSCA introduces the quadratic interpolation strategy after the population is updated by the elite leadership strategy in Section 3.1. The individual $\bar{X}_i(i = 1, 2, ..., N)$ obtained through the elite leadership strategy is sorted according to the fitness value from small to large. Three individuals $\bar{X}_i$, $\bar{X}_y$ and $\bar{X}_z$ are successively selected from the ranked population for quadratic interpolation, and generate a new individual X_i' according to Equation (6). Then, the fitness values of $\bar{X}_i$ before quadratic interpolation and X_i' after quadratic interpolation are calculated. Compare their fitness values with that of X_i^t, and keep the best individual to X_i^{t+1}.The introduction of the improved strategy further improves the exploitation ability and solution accuracy of the algorithm, and increases the diversity of the population to a certain extent.

3.3 Self-Feedback Memory Refresh Mechanism

The algorithm EQSSCA in this paper improves the quality of the solution and accelerates the convergence speed through the quadratic interpolation optimization strategy. However, as the iterations proceed, the algorithm tends to converge prematurely and stagnate. To address this problem, this paper designs a self-feedback memory refresh strategy, and sets a stagnation check counter $C(i)$. If an individual is improved through the quadratic interpolation optimization strategy, the counter is set to 0, otherwise, it is added to 1. For individuals whose fitness value has not been improved for L consecutive times, the Cauchy disturbance mutation based on double random will be performed according to Equation (7),

$$M_{i,j} = X_{i,j}^{t+1} + r_1 \times Cauchy(0, 1) \times (X_{p,j}^{t+1} - X_{q,j}^{t+1}), \quad (7)$$

among them, $Cauchy(0, 1)$ is the standard Cauchy distribution; r_1 is calculated by Equation (3), which is used to control the individual exploration behavior; $X_{p,j}^{t+1}$, $X_{q,j}^{t+1}$ are the randomly selected individuals in the population $X_i^{t+1}(i = 1, 2, ..., N)$, and $p \neq q \neq i$.

Then, calculate and compare the individual's fitness values before and after mutation. If the individual is

improved after mutation, reset the counter to 0, otherwise, continue to add 1. That is, if the individual has not been improved after the disturbance mutation, it will try to refresh again in the next iteration. The specific code segment is shown in Code 2. Among them, $f(M_i)$ and $f(X_i^{t+1})$ respectively denote the fitness values of M_i and X_i^{t+1}. L is the refresh limit. After repeated tests, when $L = 10$, the optimization effect is better.

Code 2 Self-feedback memory refresh mechanism

```
if C(i) ≥ L
  for j = 1 : D
M_{i,j} = X_{i,j}^{t+1} + r_1 × Cauchy(0,1) × (X_{p,j}^{t+1} − X_{q,j}^{t+1})
  end for
  if f(M_i) < f(X_i^{t+1})
    X_i^{t+1} = M_i
    C(i) = 0
  else
    C(i) = C(i) + 1
  end if
end if
```

3.4 Pseudo-code of EQSSCA

The pseudo-code for EQSSCA is presented by Algorithm 1.

Algorithm 1 Pseudo-code of EQSSCA

```
Initialize the parameters
Generate the initial population by Equation (1)
Evaluate the fitness value for each individual
Select the individual P with the best fitness value Destination_fitness from the current population
t = 1
while (t ≤ Max_iteration) do
  Calculate the coefficien r_1 with Equation (3)
  Update the transition probability A by Equation (5)
  for i = 1 : N do
    update coefficients r_2, r_3, r_4
    update the switch variable Q
    if Q < A then
      use Equation (2) to generate individual X̄_i
    else
      the individual X̄_i is generated by Equation (4)
    end if
  end for
  Calculate the fitness value of the individuals in the population X̄_i(i = 1, 2, ...N), and sort the individuals according to the fitness value
  for i = 1 : N do
    Perform quadratic interpolation by Equation (6) to obtain a new individual X'_i
    if f(X̄_i) < f(X'_i) then
      temp = X̄_i
    else
      temp = X'_i
    end if
    if f(temp) < f(X_i^t) then
      X_i^{t+1} = temp
      C(i)=0
    else
      X_i^{t+1} = X_i^t
      C(i) = C(i) + 1
    end if
    Execute the self-feedback memory refresh mechanism according to Code 2
    if f(X_i^{t+1}) < Destination_fitness then
      P = X_i^{t+1}
      Destination_fitness = f(X_i^{t+1})
    end if
  end for
  t=t+1
end while
Output result
```

4 Time Complexity Analysis

Time complexity is a function that qualitatively describes the running time scale of an algorithm. It can reflect the algorithm's operational efficiency and is an indispensable tool for evaluating algorithms' performance. Literatures [30] and [29] respectively analyze the time complexity of the firefly algorithm and equilibrium optimizer. The time complexity of SCA and EQSSCA is examined in this paper by employing the same approach.

The meta-heuristic algorithm won't become more effective at optimization by adding more iterations or expanding the population scale. Therefore, when solving problems, intelligent optimization algorithms usually use a fixed population size and number of iterations. And the basic variable that determines the algorithm's time complexity is the individual spatial dimension, which represents the size of a problem.

4.1 Time Complexity Analysis of SCA

For the SCA, the total population scale is assumed to be N and the dimensionality of individuals is set to n.

In the initialization stage, suppose the time to set the initial parameter is t_1; $f(n)$ represents the time to compute an individual fitness value; the time to initialize each dimension of an individual by Equation (1) is t_2; the time to compare and replace each individual with the current best solution is t_3. Thus, the time complexity of this stage is :

$$T_1 = O(t_1 + N \cdot (n \cdot t_2 + f(n) + t_3)) = O(n + f(n)). \quad (8)$$

Once the iteration is started, the maximum iteration count is assumed to be $Max\_iteration$.

At the position update stage, let the time to calculate r_1 by Equation (3) is t_4; The time for producing evenly

distributed stochastic numbers r_2, r_3 and r_4 is t_5 , respectively. And time for updating every dimension of an individual by Equation (2) is t_6. This stage has the following time complexity:

$$T_2 = O(t_4 + N \cdot n \cdot (3 \cdot t_5 + t_6))=O(n). \quad (9)$$

In the stage of boundary processing and updating the optimal individual, suppose: the boundary processing time for every dimension of an individual is t_7 ; $f(n)$ represents the time to compute an individual fitness value; the time to compare and replace each individual with the current best solution is t_3. Then, the time complexity of the stage is:

$$T_3=O\left(N \cdot (n \cdot t_7 + f\left(n\right) + t_3)\right) = O\left(n + f\left(n\right)\right). \quad (10)$$

To sum up, the time complexity of SCA is as follows:

$$\begin{aligned} T &= T_1 + Max\_iteration \cdot (T_2 + T_3) \\ &= O\left(n + f\left(n\right)\right). \end{aligned} \quad (11)$$

4.2 Time Complexity Analysis of EQSSCA

For EQSSCA, the total population number, the dimensionality of individuals, the time to set the initial parameter, the time for calculating individual fitness value, and the time for comparing and replacing each individual with the current optimal individual are consistent with the SCA. Therefore, the initialization phase of EQSSCA has the same time complexity as SCA. That is:

$$\begin{aligned} T'_1 &= T_1 = O(t_1 + N \cdot (n \cdot t_2 + f(n) + t_3)) \\ &= O(n + f(n)). \end{aligned} \quad (12)$$

Once the iteration is started, the maximum iteration count is assumed to be $Max\_iteration$.

At the individual position update stage, suppose: the time of the dynamic transition probability A calculated by Equation (5) is η_1; the time of generating a random number Q and judging which updating strategy to use is η_2, the time for calculating r_1 by Equation (3) is t_4; and the time for producing evenly distributed stochastic numbers r_2, r_3 and r_4 is t_5, respectively. Let: there are $m(0 \le m \le N)$ individuals in the population using the basic SCA position update strategy. There are $N-m$ individuals using the elite leadership strategy for position updating. The time for the $m(0 \le m \le N)$ individuals to update their positions in each dimension using Equation (2) is t_6, and the time for the remaining $N-m$ individuals to update their positions in each dimension according to Equation (4) is η_3.The time complexity of this stage is:

$$\begin{aligned} T'_2 =& O(\eta_1 + t_4 + N \cdot \eta_2 + m \cdot n \cdot (3 \cdot t_5 + t_6) \\ &+ (N - m) \cdot n \cdot (2 \cdot t_5 + \eta_3)) = O\left(n\right). \end{aligned} \quad (13)$$

At the quadratic interpolation optimization stage, the boundary processing time for every dimension of an individual is t_7; $f(n)$ still stands for the time to compute an individual fitness value, and the time for sorting the individuals in the population is η_4. Suppose: the time for position update for each individual dimension by Equation (6) is η_5; the time for comparing the fitness values of two individuals twice is $2 \cdot \eta_6$; the time of saving the better one from two individuals twice is $2 \cdot \eta_7$. The time to update the stagnation check counter $C(i)$ is η_8. Then the time complexity of this stage is:

$$\begin{aligned} T'_3 =& O(N \cdot (n \cdot t_7 + f(n)) + \eta_4 + \\ & N \cdot (n \cdot \eta_5 + n \cdot t_7 + f(n) + 2 \cdot \eta_6 + 2 \cdot \eta_7 + \eta_8)) \\ =& O(n + f(n)). \end{aligned} \quad (14)$$

During the self-feedback memory refresh stage, suppose that the time to compare and judge whether the individual is stagnant and need to mutate is η_9; s denotes the amount of individuals that need to be mutated and $0 \le s \le N$; the time to mutate each dimension of the individual by Equation (7) is η_{10}. The time for computing individual fitness values, the time for comparing two individuals' fitness values, the time for updating the stagnation check counter $C(i)$, and the time for comparing and replacing each individual with the current optimal individual are all kept the same as above. Time complexity of this period is shown as follows:

$$\begin{aligned} T'_4 = & O(N \cdot \eta_9 + s \cdot (n \cdot \eta_{10} + n \cdot t_7 + f(n) + \eta_6 + \eta_8) \\ & + N \cdot t_3) = O(n + f(n)). \end{aligned} \quad (15)$$

As a result, EQSSCA's time complexity is given below:

$$\begin{aligned} T' &= T'_1 + Max\_iteration \cdot (T'_2 + T'_3 + T'_4) \\ &= O(n + f(n)). \end{aligned} \quad (16)$$

Obviously, the time complexity of the improved algorithm EQSSCA in this paper is the same as that of SCA, and the improvement mechanisms do not weaken the algorithm's performance efficiency.

5 Simulation Experiments

To verify the optimization ability of the proposed algorithm, the CEC2017 [40] test suite, which is challenging and difficult to solve, is selected to compare and test EQSSCA and four comparative algorithms with superior performance. These four comparison algorithms include the SCA [33], a novel and efficient representative SCA improvement algorithm COSCA [15], CPWOA [21] with excellent convergence and stability in function optimization tests, and a new algorithm AOA [3] with superior performance has been proposed in recent two years. The five algorithms are compared and tested on 50 and 100 dimensions.

In order to ensure the objectivity and fairness of the algorithm comparison, the experimental environment for running each algorithm is Windows 10 and Matlab R2019b. The five algorithms were independently run 50 times on each test function. The population size is 30,

Table 1: Comparison results of five algorithms with dim=50/100

Function	Algorithm	dim=50			dim=100		
		Best	Mean	Std	Best	Mean	Std
F_{21}	EQSSCA	2.368×10^{3}	2.410×10^{3}	2.379×10^{1}	2.622×10^{3}	2.700×10^{3}	6.152×10^{1}
	SCA	2.863×10^{3}	2.949×10^{3}	4.476×10^{1}	3.954×10^{3}	4.115×10^{3}	7.724×10^{1}
	COSCA	2.855×10^{3}	2.926×10^{3}	3.857×10^{1}	3.970×10^{3}	4.171×10^{3}	9.985×10^{1}
	CPWOA	2.815×10^{3}	3.041×10^{3}	1.099×10^{2}	3.974×10^{3}	4.404×10^{3}	2.280×10^{2}
	AOA	2.956×10^{3}	3.108×10^{3}	8.664×10^{1}	4.299×10^{3}	4.705×10^{3}	1.910×10^{2}
F_{22}	EQSSCA	2.333×10^{3}	1.145×10^{4}	1.972×10^{3}	2.278×10^{4}	3.098×10^{4}	3.327×10^{3}
	SCA	1.557×10^{4}	1.682×10^{4}	4.756×10^{2}	3.171×10^{4}	3.508×10^{4}	7.068×10^{2}
	COSCA	8.989×10^{3}	1.303×10^{4}	1.429×10^{3}	2.861×10^{4}	3.149×10^{4}	1.113×10^{3}
	CPWOA	1.192×10^{4}	1.450×10^{4}	1.037×10^{3}	2.921×10^{4}	3.176×10^{4}	1.612×10^{3}
	AOA	1.406×10^{4}	1.599×10^{4}	7.355×10^{2}	3.133×10^{4}	3.358×10^{4}	8.744×10^{2}
F_{23}	EQSSCA	2.807×10^{3}	2.873×10^{3}	3.728×10^{1}	3.173×10^{3}	3.305×10^{3}	6.294×10^{1}
	SCA	3.484×10^{3}	3.671×10^{3}	7.581×10^{1}	4.862×10^{3}	5.179×10^{3}	1.288×10^{2}
	COSCA	3.422×10^{3}	3.607×10^{3}	7.050×10^{1}	4.982×10^{3}	5.432×10^{3}	1.916×10^{2}
	CPWOA	3.456×10^{3}	3.760×10^{3}	1.633×10^{2}	4.421×10^{3}	5.126×10^{3}	2.411×10^{2}
	AOA	4.032×10^{3}	4.541×10^{3}	2.444×10^{2}	5.950×10^{3}	7.250×10^{3}	5.850×10^{2}
F_{24}	EQSSCA	2.959×10^{3}	3.024×10^{3}	2.980×10^{1}	3.687×10^{3}	3.896×10^{3}	9.632×10^{1}
	SCA	3.697×10^{3}	3.841×10^{3}	7.160×10^{1}	6.437×10^{3}	7.143×10^{3}	2.930×10^{2}
	COSCA	3.764×10^{3}	3.954×10^{3}	1.045×10^{2}	6.673×10^{3}	7.678×10^{3}	3.905×10^{2}
	CPWOA	3.478×10^{3}	3.773×10^{3}	1.460×10^{2}	5.840×10^{3}	6.525×10^{3}	3.183×10^{2}
	AOA	4.375×10^{3}	4.981×10^{3}	3.144×10^{2}	9.735×10^{3}	1.171×10^{4}	1.115×10^{3}
F_{25}	EQSSCA	2.990×10^{3}	3.064×10^{3}	2.703×10^{1}	3.504×10^{3}	3.660×10^{3}	7.357×10^{1}
	SCA	6.619×10^{3}	8.298×10^{3}	1.038×10^{3}	1.518×10^{4}	2.022×10^{4}	2.621×10^{3}
	COSCA	7.244×10^{3}	9.088×10^{3}	1.036×10^{3}	1.712×10^{4}	2.017×10^{4}	1.573×10^{3}
	CPWOA	3.720×10^{3}	4.435×10^{3}	3.869×10^{2}	6.765×10^{3}	8.509×10^{3}	9.333×10^{2}
	AOA	1.243×10^{4}	1.636×10^{4}	1.565×10^{3}	2.275×10^{4}	2.907×10^{4}	2.744×10^{3}
F_{26}	EQSSCA	4.479×10^{3}	5.188×10^{3}	2.777×10^{2}	9.425×10^{3}	1.103×10^{4}	8.369×10^{2}
	SCA	1.172×10^{4}	1.341×10^{4}	8.532×10^{2}	3.178×10^{4}	3.962×10^{4}	2.540×10^{3}
	COSCA	1.108×10^{4}	1.242×10^{4}	7.429×10^{2}	3.027×10^{4}	3.512×10^{4}	2.152×10^{3}
	CPWOA	1.229×10^{4}	1.483×10^{4}	1.132×10^{3}	2.870×10^{4}	3.748×10^{4}	3.836×10^{3}
	AOA	1.462×10^{4}	1.712×10^{4}	1.207×10^{3}	4.259×10^{4}	5.250×10^{4}	4.266×10^{3}
F_{27}	EQSSCA	3.303×10^{3}	3.501×10^{3}	1.185×10^{2}	3.477×10^{3}	3.655×10^{3}	9.190×10^{1}
	SCA	4.372×10^{3}	4.835×10^{3}	2.342×10^{2}	7.120×10^{3}	8.231×10^{3}	6.171×10^{2}
	COSCA	4.537×10^{3}	5.052×10^{3}	2.372×10^{2}	7.177×10^{3}	8.277×10^{3}	4.518×10^{2}
	CPWOA	3.770×10^{3}	4.643×10^{3}	6.025×10^{2}	4.568×10^{3}	6.069×10^{3}	1.204×10^{3}
	AOA	5.716×10^{3}	6.903×10^{3}	6.882×10^{2}	1.069×10^{4}	1.281×10^{4}	1.214×10^{3}
F_{28}	EQSSCA	3.271×10^{3}	3.323×10^{3}	2.816×10^{1}	3.583×10^{3}	3.742×10^{3}	8.342×10^{1}
	SCA	6.776×10^{3}	8.237×10^{3}	7.636×10^{2}	2.159×10^{4}	2.528×10^{4}	2.140×10^{3}
	COSCA	6.694×10^{3}	7.696×10^{3}	5.567×10^{2}	1.870×10^{4}	2.152×10^{4}	1.605×10^{3}
	CPWOA	4.122×10^{3}	5.513×10^{3}	8.586×10^{2}	1.011×10^{4}	1.247×10^{4}	1.014×10^{3}
	AOA	8.910×10^{3}	1.270×10^{4}	1.541×10^{3}	2.812×10^{4}	3.478×10^{4}	2.887×10^{3}
F_{29}	EQSSCA	3.432×10^{3}	4.058×10^{3}	3.157×10^{2}	5.267×10^{3}	6.863×10^{3}	5.679×10^{2}
	SCA	6.698×10^{3}	8.334×10^{3}	6.581×10^{2}	1.655×10^{4}	2.760×10^{4}	8.228×10^{3}
	COSCA	7.160×10^{3}	8.704×10^{3}	9.693×10^{2}	1.826×10^{4}	2.728×10^{4}	9.594×10^{3}
	CPWOA	6.772×10^{3}	9.143×10^{3}	1.306×10^{3}	1.330×10^{4}	1.957×10^{4}	3.094×10^{3}
	AOA	9.827×10^{3}	4.878×10^{4}	3.954×10^{4}	7.042×10^{4}	6.762×10^{5}	5.088×10^{5}
F_{30}	EQSSCA	1.116×10^{6}	1.804×10^{6}	6.518×10^{5}	5.486×10^{5}	2.260×10^{6}	1.657×10^{6}
	SCA	5.500×10^{8}	1.031×10^{9}	3.256×10^{8}	7.123×10^{9}	1.129×10^{10}	2.227×10^{9}
	COSCA	4.482×10^{8}	8.684×10^{8}	1.945×10^{8}	6.532×10^{9}	1.238×10^{10}	2.506×10^{9}
	CPWOA	8.306×10^{7}	2.582×10^{8}	1.049×10^{8}	7.705×10^{8}	1.664×10^{9}	5.717×10^{8}
	AOA	1.963×10^{9}	6.827×10^{9}	2.398×10^{9}	2.549×10^{10}	4.063×10^{10}	7.151×10^{9}

and the maximum evolutionary generation is 1000. In terms of algorithm parameter setting, the control parameter $a = 2$ in EQSSCA and SCA algorithms. In EQSSCA algorithm, the initial and final values of the transition probability are $A_{\max} = 1$ and $A_{\min} = 0.4$ respectively, the refresh limit $L = 10$. For the COSCA algorithm, the regulation coefficient η=1, the initial and final values of the control parameters are $a_{start} = 1$ and $a_{end} = 0$, respectively, and the proportion of elite individuals $pr = 0.1$. The constant $b = 1$ used in the CPWOA to define the shape of a logarithmic helix. For AOA algorithm, the control parameter $\mu = 0.5$ and sensitive parameter α=5. To ensure the fairness and credibility of the experimental results and comparative analysis, the parameters setting of the above five algorithms are the same as their original literature values, without any changes.

5.1 CEC2017 Test Suite

In the CEC2017 test suite, all 30 functions are rotation and shifted functions, which increases the difficulty of the algorithm finding the optimal solution. And the value range of each dimension of the independent variable is [-100, 100]. Among them, $F_1 - F_3$ are unimodal functions, $F_4 - F_{10}$ are multimodal functions, $F_{11} - F_{20}$ are hybrid functions, and $F_{21} - F_{30}$ are composition functions. In this paper, all 10 composition functions $F_{21} - F_{30}$, which are the most difficult and challenging to solve in the test suite, are selected to test the optimization performance of the EQSSCA. These composition functions consist of multiple hybrid functions or benchmark functions that have been rotated and shifted. Each sub-function adds an offset and then is assigned a weight. Therefore, the composition functions further increase the difficulty of algorithm optimization. In addition, the composition functions $F_{21} - F_{30}$ change the theoretical optimum to 2100-3000 by the offset property, avoiding the problem of convergence to zero in the algorithm test.

5.2 Experimental Results and Analysis

To evaluate EQSSCA's optimization capability, the dimensions of the CEC2017 test suite are set to dim = 50/100, respectively. The five algorithms including EQSSCA, SCA, COSCA, CPWOA, and AOA, are tested under different dimensions. Table 1 statistics the test results of the above five algorithms running independently for 50 times when the spatial dimension dim = 50/100.

As can be seen from Table 1, under the high-dimensional conditions of 50 and 100 dimensions, for the most challenging composition functions $F_{21} - F_{30}$, the solution precision of EQSSCA in this paper outperforms the SCA, COSCA, CPWOA, and AOA, and EQSSCA has an obvious solution advantage. For composition functions F_{21}, $F_{23} - F_{30}$, EQSSCA's mean and optimal values are the best results among the five algorithms. Moreover, EQSSCA outperforms the four comparison algorithms in terms of standard deviation, which fully shows the effectiveness of the EQSSCA for improving the SCA optimization mechanism. As for F_{22}, the standard deviation obtained by EQSSCA is a little worse than the other comparison algorithms. However, EQSSCA's optimal and mean values are the best among the five algorithms, which indicates that EQSSCA fluctuates at an optimal solution level, while the other algorithms are stable at a poor solution level, and their solution ability is still inferior to EQSSCA.

Based on the above optimization results and analysis, it is indicated that under different dimensions, EQSSCA's optimization results significantly outperform the remaining four comparative algorithms, showing excellent solution ability and stability. These test results fully illustrate that EQSSCA effectively solves the problems of low optimization accuracy and solution instability of SCA during function optimization.

Figure 1: Wilcoxon rank-sum test results between EQSSCA and each comparison algorithm

5.3 Significance Analysis of CEC2017 Experimental Results

To verify that the experimental results of EQSSCA are significantly different from those of the SCA, COSCA, CPWOA, and AOA, this paper uses the Wilcoxon rank-sum test method for statistical analysis. The null hypothesis rejection value for its significance evaluation is 5% [7]. That is, when $p - value < 0.05$, it means that for the current function, there is a significant difference between the optimization results of the EQSSCA and the comparison algorithm. For the composition functions $F_{21} - F_{30}$ in the CEC2017 test suite, Figure 1 intuitively shows EQSSCA and four comparison algorithms' rank-sum test results in the 100-dimensional condition.

Observing Figure 1, it is clear that on F_{22}, the $p - values$ obtained by the Wilcoxon rank-sum test between EQSSCA and COSCA, CPWOA are slightly greater than 0.05. However, for the remaining 9 functions, the $p - values$ obtained by the Wilcoxon rank-sum test between EQSSCA and four comparison algorithms are less than

0.05, indicating that EQSSCA's optimization results are significantly different from those of the four compared algorithms. Moreover, according to the solution results in Section 5.2, the EQSSCA's solution results have significant advantages compared with those of other comparison algorithms.

5.4 Analysis of Convergence Curve

An algorithm's optimization performance can be directly shown by convergence curves. The convergence curves show the algorithm's variation in convergence speed and its ability to jump out of local extremes during the optimization process. To more clearly compare the optimization ability of the proposed algorithm EQSSCA with the four comparison algorithms, Figure 2-Figure 11 show the comparison results of the five algorithms on the above 10 functions $F_{21} - F_{30}$ when $Max\_iteration = 1000$ and $dim = 100$.

Figure 2-Figure 11 clearly show the trends of the fitness values of EQSSCA, SCA, COSCA, CPWOA, and AOA algorithms in the iteration process. As can be seen from these figures, the proposed algorithm EQSSCA outperforms the other four algorithms in terms of solving ability and convergence speed on the composition functions $F_{21} - F_{30}$, showing significant superiority.

Specifically, for Figure 2, Figure 10, and Figure 11, the SCA, COSCA, CPWOA, and AOA all converge slower than EQSSCA from the beginning to the end of the iteration. In addition, these four comparison algorithms are trapped in local optima at late iteration and cannot jump out. However, the proposed algorithm EQSSCA always maintains a fast convergence rate, and its precision is the best among the five algorithms. For Figure 3, EQSSCA sometimes trapped in local extrema during the iterative process, the convergence speed is a little slow. However, by the late iteration, the EQSSCA can get rid of the extremes and converge quickly, and its convergence precision is the best among the five algorithms. For Figure 4 and Figure 5, the convergence speed of AOA is always slower than that of EQSSCA. In contrast, although the SCA, COSCA, and CPWOA converge faster than EQSSCA at the initial stage of iteration, they quickly fall into the local extremum and their convergence speed decreases. When iterating to the 500th generation, the convergence accuracy of EQSSCA is significantly superior to that of the four compared algorithms. In Figure 6, Figure 7, and Figure 9, SCA and AOA have a slow convergence rate compared with EQSSCA. The convergence speed of COSCA and CPWOA is slightly faster than that of EQSSCA in the early stage, but in the middle of iteration, the convergence speed has been all slower than that of EQSSCA. As shown in Figure 8, compared with the EQSSCA, the convergence speed of SCA, COSCA and AOA is always slow. At the early stage of iteration, the EQSSCA sometimes falls into the local optimum and stagnates, and the convergence rate is slightly lower than CPWOA. However, after jumping out of the local optimum, EQSSCA maintains a fast convergence speed. At the later stage of iteration, the convergence precision of EQSSCA is better than the other four algorithms.

Based on the above analysis, we can see that under different dimensions of the composition functions in CEC2017 test suite, the EQSSCA algorithm outperforms the four comparison algorithms in optimization precision and convergence performance. This is mainly because an elite leadership strategy is inserted into the SCA algorithm, which accelerates the algorithm's convergence while maintaining the population diversity and better balances the global and local search of the algorithm. The quadratic interpolation optimization strategy is introduced to further improve the exploitation ability and solution accuracy of the algorithm. Finally, for individuals that have not been improved many times, the self-feedback memory refresh strategy is implemented to enrich the population diversity and strengthen the algorithm's capability to escape from local extreme values.

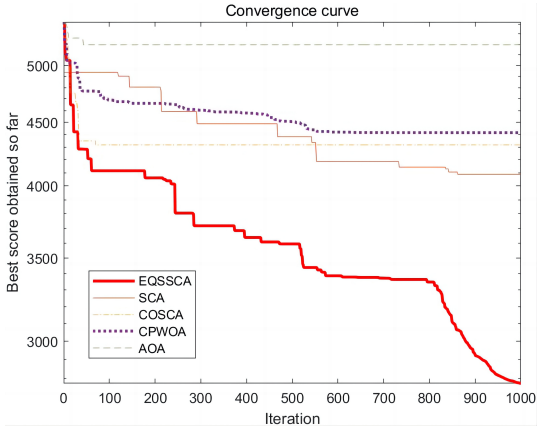


Figure 2: Convergence curve of 5 algorithms on F_{21}

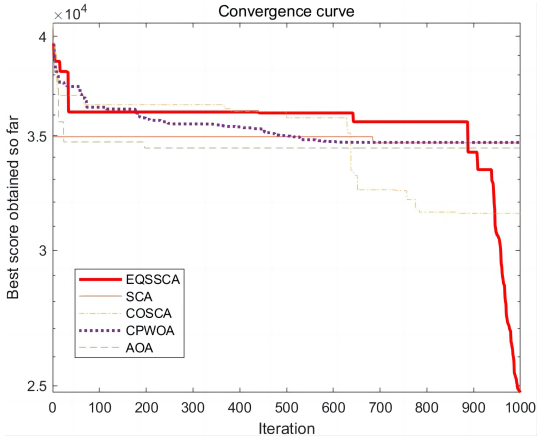


Figure 3: Convergence curve of 5 algorithms on F_{22}

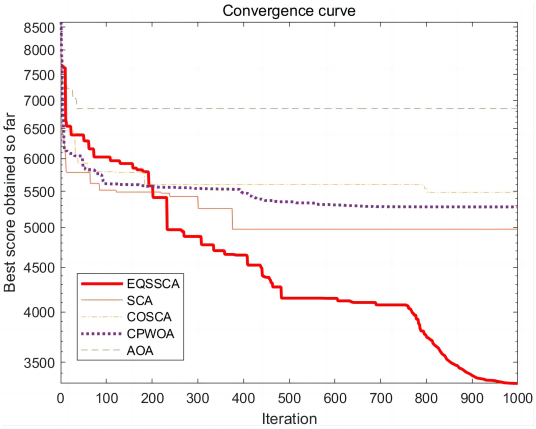


Figure 4: Convergence curve of 5 algorithms on F_{23}

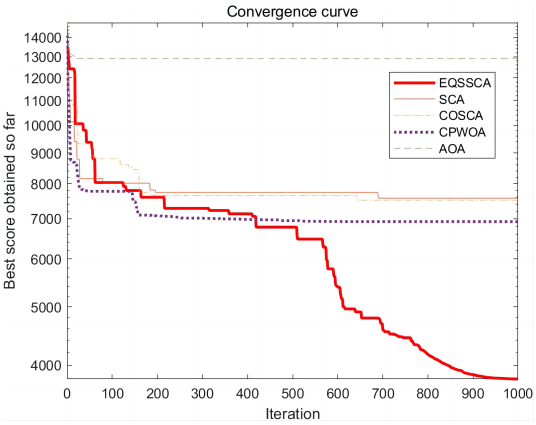


Figure 5: Convergence curve of 5 algorithms on F_{24}

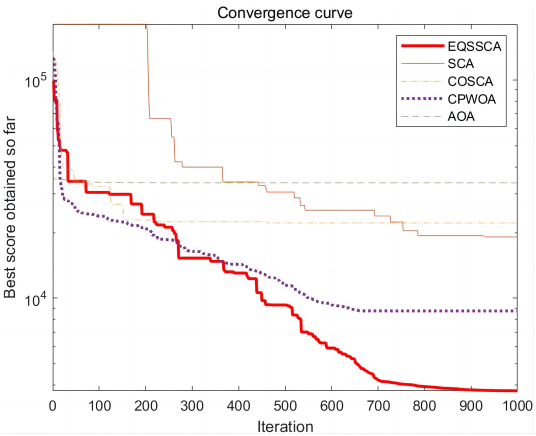


Figure 6: Convergence curve of 5 algorithms on F_{25}

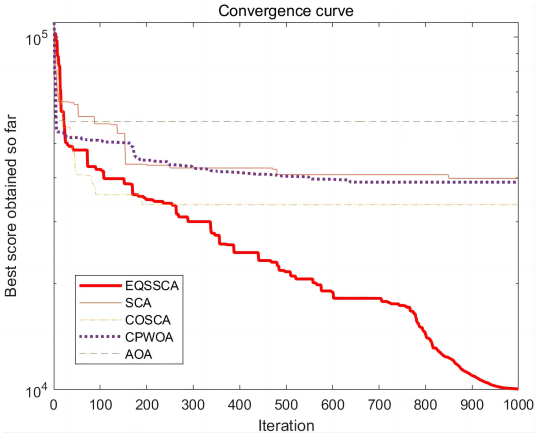


Figure 7: Convergence curve of 5 algorithms on F_{26}

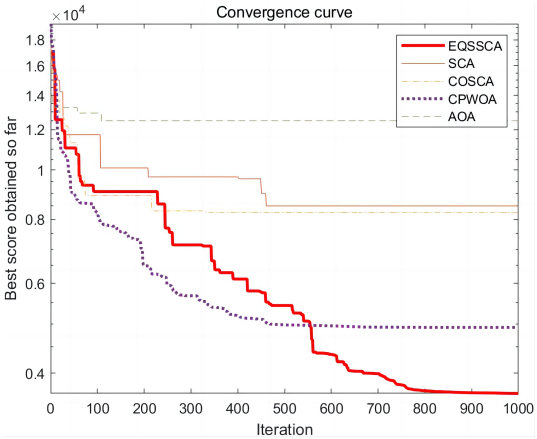


Figure 8: Convergence curve of 5 algorithms on F_{27}

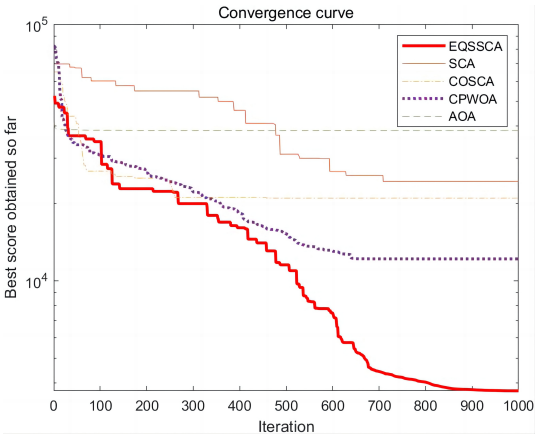


Figure 9: Convergence curve of 5 algorithms on F_{28}

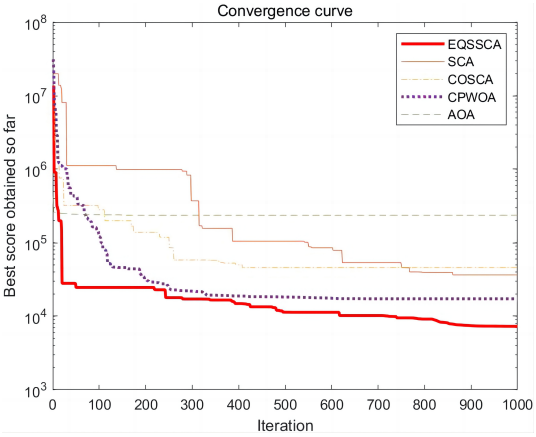


Figure 10: Convergence curve of 5 algorithms on F_{29}

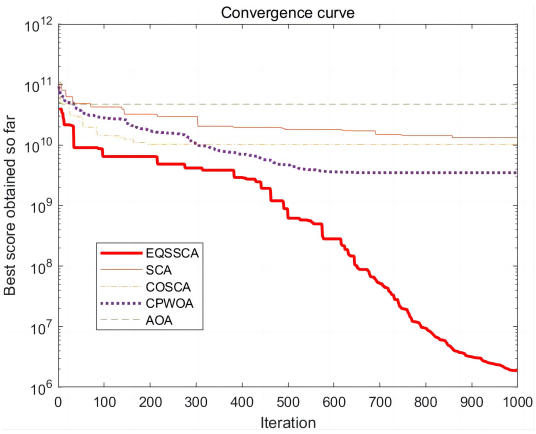


Figure 11: Convergence curve of 5 algorithms on F_{30}

6 Conclusions

To better solve the optimization problems in network security, improve the optimization performance of the basic sine-cosine algorithm (SCA), and strengthen the algorithm's application ability in practical complex problems, this paper proposes a modified sine-cosine algorithm (EQSSCA) combining three mechanisms. In terms of algorithm improvement, the elite leadership mechanism is adopted, which effectively regulates the balance between the algorithm's global search ability and local mining capacity. Then, the quadratic interpolation optimization mechanism is introduced to further improve the algorithm's convergence speed and optimization accuracy. Finally, a self-feedback memory refresh mechanism is employed to enhance the algorithm's ability to escape from local extremes and avoid premature convergence. Besides, the results of the time complexity analysis prove that the time complexity of EQSSCA and SCA algorithms are the same, and these three improvements do not weaken the algorithm's efficiency.

To evaluate the performance of EQSSCA, CEC2017 test suite are used to test EQSSCA and four representative comparison algorithms. The comparison results, which include numerical results, statistical tests, and convergence curves, show that the improved algorithm in this paper outperforms the other four comparison algorithms in terms of solution accuracy, convergence speed, and solution stability. Next, we will continue to enhance the optimization mechanism of EQSSCA and sine cosine algorithm, strengthen the problem adaptability and optimization stability of the algorithms, and apply it to solve complex optimization problems in network security.

Acknowledgments

This work is jointly supported by the Key R&D and Promotion Projects of Henan Province, China (grant number 222102210065), and the Major Science and Technology Project of Henan Province, China (grant number 201300210400), and the Action Plan for Postgraduate Training Innovation and Quality Improvement of Henan University (grant number SYLYC2022150). The authors would like to thank the anonymous reviewers and the editors for their helpful comments.

References

[1] M. Abdel-Basset, R. Mohamed, M. Abouhawwash, R. K. Chakrabortty, and M. J. Ryan, "Ea-msca: An effective energy-aware multi-objective modified sine-cosine algorithm for real-time task scheduling in multiprocessor systems: Methods and analysis," *Expert systems with applications*, vol. 173, p. 114699, 2021.

[2] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new metaheuristic optimization algorithm," *Information Sciences*, vol. 540, pp. 131–159, 2020.

[3] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer methods in applied mechanics and engineering*, vol. 376, p. 113609, 2021.

[4] A. Alzaqebah, I. Aljarah, O. Al-Kadi, and R. Damaševičius, "A modified grey wolf optimization algorithm for an intrusion detection system," *Mathematics*, vol. 10, no. 6, p. 999, 2022.

[5] E. Barnard, "Determination and the no-free-lunch paradox," *Neural computation*, vol. 23, no. 7, pp. 1899–1909, 2011.

[6] X. Cai, S. Geng, J. Zhang, D. Wu, Z. Cui, W. Zhang, and J. Chen, "A sharding scheme-based many-objective optimization algorithm for enhancing security in blockchain-enabled industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7650–7658, 2021.

[7] J. Carrasco, S. García, M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for

comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm and Evolutionary Computation*, vol. 54, p. 100665, 2020.

[8] H. Chen, A. A. Heidari, X. Zhao, L. Zhang, and H. Chen, "Advanced orthogonal learning-driven multi-swarm sine cosine optimization: Framework and case studies," *Expert Systems with Applications*, vol. 144, p. 113113, 2020.

[9] A. Daoui, H. Karmouni, M. Sayyouri, H. Qjidaa, M. Maaroufi, and B. Alami, "New robust method for image copyright protection using histogram features and sine cosine algorithm," *Expert Systems with Applications*, vol. 177, p. 114978, 2021.

[10] S. Dwivedi, M. Vardhan, and S. Tripathi, "Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection," *Cluster Computing*, pp. 1–20, 2021.

[11] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, p. 105190, 2020.

[12] Z.-k. Feng, J.-f. Duan, W.-j. Niu, Z.-q. Jiang, and Y. Liu, "Enhanced sine cosine algorithm using opposition learning, adaptive evolution and neighborhood search strategies for multivariable parameter optimization problems," *Applied Soft Computing*, vol. 119, p. 108562, 2022.

[13] Z.-k. Feng, S. Liu, W.-j. Niu, B.-j. Li, W.-c. Wang, B. Luo, and S.-m. Miao, "A modified sine cosine algorithm for accurate global optimization of numerical functions and multiple hydropower reservoirs operation," *Knowledge-Based Systems*, vol. 208, p. 106461, 2020.

[14] M. Gunduz and M. Aslan, "Djaya: A discrete jaya algorithm for solving traveling salesman problem," *Applied Soft Computing*, vol. 105, p. 107275, 2021.

[15] W. Guo, Y. Wang, F. Dai, and T. Liu, "Alternating sine cosine algorithm based on elite chaotic search strategy," *Control and decision (in Chinese)*, vol. 34, no. 8, pp. 1654–1662, 2019.

[16] S. Gupta, K. Deep, and A. P. Engelbrecht, "A memory guided sine cosine algorithm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 93, p. 103718, 2020.

[17] Q. S. Hamad, H. Samma, S. A. Suandi, and J. Mohamad-Saleh, "Q-learning embedded sine cosine algorithm (qlesca)," *Expert Systems with Applications*, vol. 193, p. 116417, 2022.

[18] N. Hamian, M. Bayat, M. R. Alaghband, Z. Hatefi, and S. M. Pournaghi, "Blockchain-based user reenrollment for biometric authentication systems," *IJ of Electronics and Information Engineering*, vol. 14, no. 1, pp. 18–38, 2022.

[19] F. A. Hashim and A. G. Hussien, "Snake optimizer: A novel meta-heuristic optimization algorithm," *Knowledge-Based Systems*, vol. 242, p. 108320, 2022.

[20] M.-S. Hwang, E. F. Cahyadi, Y.-C. Chou, and C.-Y. Yang, "Cryptanalysis of kumar's remote user authentication scheme with smart card," in *2018 14th International Conference on Computational Intelligence and Security (CIS)*. IEEE, 2018, pp. 416–420.

[21] Q. Huang, J. Li, C. Song, C. Xu, and X. Lin, "Whale optimization algorithm based on cosine control factor and polynomial mutation," *Control and Decision (in Chinese)*, vol. 35, no. 3, pp. 559–568, 2020.

[22] G. A. Kale and U. Yüzgeç, "Advanced strategies on update mechanism of sine cosine optimization algorithm for feature selection in classification problems," *Engineering Applications of Artificial Intelligence*, vol. 107, p. 104506, 2022.

[23] X. Kan, Y. Fan, Z. Fang, L. Cao, N. N. Xiong, D. Yang, and X. Li, "A novel iot network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network," *Information Sciences*, vol. 568, pp. 147–162, 2021.

[24] H. Karmouni, M. Chouiekh, S. Motahhir, H. Qjidaa, M. O. Jamil, and M. Sayyouri, "Optimization and implementation of a photovoltaic pumping system using the sine–cosine algorithm," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105104, 2022.

[25] G. Kaur and J. Dhillon, "Economic power generation scheduling exploiting hill-climbed sine–cosine algorithm," *Applied Soft Computing*, vol. 111, p. 107690, 2021.

[26] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Expert systems with applications*, vol. 149, p. 113338, 2020.

[27] B. Khokhar, S. Dahiya, and K. S. Parmar, "Load frequency control of a microgrid employing a 2d sine logistic map based chaotic sine cosine algorithm," *Applied Soft Computing*, vol. 109, p. 107564, 2021.

[28] Y. Li, Y. Zhao, and J. Liu, "Dimension by dimension dynamic sine cosine algorithm for global optimization problems," *Applied Soft Computing*, vol. 98, p. 106933, 2021.

[29] J. Liu, W. Li, and Y. Li, "Lwmeo: An efficient equilibrium optimizer for complex functions and engineering design problems," *Expert Systems with Applications*, vol. 198, p. 116828, 2022.

[30] J. Liu, Y. Mao, X. Liu, and Y. Li, "A dynamic adaptive firefly algorithm with globally orientation," *Mathematics and Computers in Simulation*, vol. 174, pp. 76–101, 2020.

[31] J. Liu, L. Teng, and S. Yin, "An improved cuckoo algorithm based on adaptive simulated annealing method used for traveling salesman problem," *INTERNATIONAL JOURNAL OF ELECTRONICS & INFORMATION ENGINEERING*, vol. 8, pp. 107–123, 2018.

[32] D. V. Lyridis, "An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints," *Ocean Engineering*, vol. 241, p. 109890, 2021.

[33] S. Mirjalili, "Sca: a sine cosine algorithm for solving optimization problems," *Knowledge-based systems*, vol. 96, pp. 120–133, 2016.

[34] A. Ponmalar and V. Dhanakoti, "An intrusion detection approach using ensemble support vector machine based chaos game optimization algorithm in big data platform," *Applied Soft Computing*, vol. 116, p. 108295, 2022.

[35] U. Raut and S. Mishra, "An improved sine–cosine algorithm for simultaneous network reconfiguration and dg allocation in power distribution systems," *Applied Soft Computing*, vol. 92, p. 106293, 2020.

[36] R. M. Rizk-Allah, H. Abdulkader, S. S. A. Elatif, W. S. Elkilani, E. Al Maghayreh, H. Dhahri, and A. Mahmood, "A novel binary hybrid pso-eo algorithm for cryptanalysis of internal state of rc4 cipher," *Sensors*, vol. 22, no. 10, p. 3844, 2022.

[37] A. K. Saha, "Multi-population-based adaptive sine cosine algorithm with modified mutualism strategy for global optimization," *Knowledge-Based Systems*, vol. 251, p. 109326, 2022.

[38] A. K. Shukla, "Detection of anomaly intrusion utilizing self-adaptive grasshopper optimization algorithm," *Neural Computing and Applications*, vol. 33, no. 13, pp. 7541–7561, 2021.

[39] F. Wei, Y. Zhang, J. Li, and Y. Shi, "Improved sine cosine algorithm based on dynamic classification strategy," *Systems Engineering and Electronic (in Chinese)*, vol. 43, no. 6, pp. 1596–1605, 2021.

[40] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 competition on constrained real-parameter optimization," *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*, 2017.

[41] W. Zhou, P. Wang, A. A. Heidari, M. Wang, X. Zhao, and H. Chen, "Multi-core sine cosine optimization: Methods and inclusive analysis," *Expert Systems with Applications*, vol. 164, p. 113974, 2021.

Biography

Jingsen Liu received his Ph.D. from Northwestern Polytechnical University, Xian, China. He is a professor of College of Software, Henan University, Kaifeng, China. His research interests include intelligence algorithm, optimal control and network information security, etc.

Fangyuan Zhao is a master of College of Software, Henan University. Her research interest is intelligence algorithm.

Ping Hu received her Master Degree from China University of Mining and Technology, Xuzhou, China. She is a lecturer of College of Software, Henan University. Her research interests include intelligent algorithm, multiple-robot path planning, etc.

Deep Learning Models: Vulnerability Mining Research for Network Security

Lili Ban[1,2] and Yan Li[1]
(Corresponding author: Lili Ban)
Hebei Academy of Fine Arts, Shijiazhuang 050000, China[1]
Maha Sarakham University, Thailand[2]
Email: ba56991822281@163.com
(Received July 22, 2022; Revised and Accepted June 18, 2023; First Online Aug. 25, 2023)

Abstract

Internet security is crucial to the security of network systems, and vulnerabilities in smart contracts can affect the security of blockchain networks. This paper used the long short-term memory (LSTM) algorithm, a deep learning approach, to identify and exploit vulnerabilities in smart contract code based on the similarity between smart contract code and text sequences. In addition, to strengthen the performance of the vulnerability detection algorithm, the word2vec word vector was combined with the mutual information values of keywords. The coordinate attention (CA) mechanism was used to highlight critical features and bidirectional LSTM was applied to incorporate contextual information. Finally, the vulnerability detection algorithm was compared with support vector machine (SVM) and traditional LSTM algorithms in simulated experiments. The results demonstrated that the vulnerability detection algorithm converged faster and more steadily during training, and the proposed algorithm achieved the highest performance in both binary classification and multi-classification of smart contract vulnerabilities.

Keywords: Bidirectional Long Short-Term Memory; Deep Learning; Smart Contracts; Vulnerability

1 Introduction

Network security concerns are getting greater attention as the Internet continues to evolve. Vulnerabilities are flaws or mistakes in software, systems, or networks that can be used by attackers to commit harmful acts in the context of network security [22]. Therefore, timely detection and remediation of vulnerabilities are crucial to maintaining the security of computer systems. Blockchain, as a product of Internet development, enables users to conduct more secure transactions and data storage [8]. Smart contracts are protocol codes in the blockchain that can automatically execute specified functions and are an important component in ensuring the security of blockchain networks [3–6, 11, 12]. The security and reliability of smart contracts directly affect the network security of the blockchain. Compared to traditional software and web vulnerabilities, smart contracts have similar vulnerability characteristics due to their difficulty to modify once deployed and their automatic execution features [10].

These vulnerabilities can not only be exploited maliciously, but can also lead to the malfunctioning of contracts. Moreover, since smart contracts are often associated with blockchain, serious vulnerabilities can result in economic losses. Identifying and remediating vulnerabilities in smart contracts is crucial to maintaining the security of blockchain networks. Traditional vulnerability detection relies on manual auditing, which is not only inefficient, but also lacks accuracy. Deep learning algorithms can automatically extract and learn feature data from smart contracts, enabling efficient vulnerability identification [18].

Shi [16] developed a machine learning-based vulnerability text classifier that standardized the unified description of vulnerability information, providing a solid foundation for vulnerability analysis. To identify reentrancy vulnerabilities in Ethereum smart contracts, Samreen [15] proposed a framework that integrates static and dynamic analysis and tested its effectiveness. Hu [7] introduced a new and efficient method for detecting memory-related vulnerabilities using vulnerability features. The experimental results demonstrated the feasibility and effectiveness of the proposed approach. In this paper, based on the similarity between smart contract code and textual sequences, we employ the long short-term memory (LSTM) algorithm in deep learning to identify and exploit vulnerabilities in smart contracts. Additionally, to enhance the performance of the vulnerability detection algorithm, the word2vec word vector was combined with the mutual information values of keywords. The coordinate attention (CA) mechanism was applied to highlight critical feature, and bidirectional LSTM was used to incorporate

contextual information. Finally, the proposed vulnerability detection algorithm was compared with SVM and traditional LSTM algorithms in simulated experiments.

2 Deep Learning Based Smart Contract Vulnerability Mining Identification

During the development of the Internet, its network security is crucial. Various protocols and software programs in the Internet will inevitably generate vulnerabilities in the process of writing, which may be caused by programmers' writing errors or may be caused by not fully considering other conditions in the writing process. Therefore, in order to ensure network security [14], timely investigation of vulnerabilities is required.

Traditional manual audits are time-consuming and inaccurate; intelligent algorithms can automatically extract vulnerability features and identify them more efficiently. The characteristics of vulnerabilities may differ in different kinds of protocols and software programs. This paper takes smart contracts in blockchain networks as the object of vulnerability mining and identification [2].

```
pragma solidity >=0.6.25;
contract Example {
   uint128 number;
   function store(uint128 num) public {
     number = num;
   }
}
```

Figure 1: Part of the code of the smart contract

Blockchain is a product of the Internet development process that uses distributed nodes to store data. A smart contract is a program in the blockchain that allows nodes in the blockchain network that do not trust each other to perform tasks together. Smart contracts can automatically execute code-compliant applications [13] without the need for third-party validation, but it is also make users participating in the contract to lose once the code has vulnerabilities. Given that it is difficult to change a smart contract once it is published on the blockchain, it is necessary to identify vulnerabilities before publication. Figure 1 shows the code of some smart contracts. From Figure 1, we can see that the code of smart contracts is similar to the structure of a text sequence, so we can consider the code of smart contracts as a text sequence, and then use deep learning algorithms to identify vulnerabilities in the "text sequence" [20].

Figure 2 illustrates the basic flow of vulnerability identification for smart contracts using deep learning algorithms:

1) The code text of smart contracts is collected.

2) The code text is pre-processed, including word separation, case conversion, word type reduction, removal of non-text symbols [1], etc.

3) To vectorize the code text, this paper uses word2vec to obtain the word vector of the code text. In order to highlight the keyword information in it, the term frequency-inverse document frequency (TF-IDF) values of the words in the text are used to construct the two-dimensional matrix of feature words [17], and the mutual information values between the feature words and the text labels are calculated by combining the text labels, i.e., vulnerability types, and the key feature words are selected. The mutual information values of the key feature words are combined with the corresponding word2vec word vectors in a weighted manner. The relevant formula is:

$$\begin{cases} MI(w_{i,j},c) = \sum_{w_{i,j}\in T}\sum_{c\in C} p(w_{i,j},c)\log_2\frac{p(w_{i,j},c)}{p(w_{i,j})p(c)} \\ V(w) = v(w)e^{MI(w)} \end{cases}$$

where $w_{i,j}$ is the j-th feature word in text d_i, T is the set of all feature words, c is the text label (vulnerability type), C is the set of text labels, $p(w_{i,j},c)$ is the total TF-IDF value of $w_{i,j}$ in the text labeled as c, $p(w_{i,j})$ is the total TF-IDF value of $w_{i,j}$, $p(c)$ is the total TF-IDF value of the feature words within the text labeled as c, $v(w)$ is the word2vec word vector of key feature word w, $MI(w)$ is the mutual information value of key feature word w, and $V(w)$ is the word vector of key feature word w after weighting [9].

4) The CA mechanism is used to obtain the key features of the code text vector. The formula is:

$$\begin{cases} A_{H\times W} = f(MLP(Maxpool_{1D}(Y_{H\times W})) \\ \qquad\qquad +MLP(Avgpool_{1D}(Y_{H\times W}))) \\ Z_{H\times W} = A_{H\times W}\otimes Y_{H\times W} \\ B_{H\times 1} = f(conv_{7\times 7}[Maxpool_y(Z_{H\times W}); \\ \qquad\qquad Avgpool_y(Z_{H\times W})]) \\ O_{H\times W} = Z_{H\times W}\otimes B_{H\times 1} \end{cases} \quad (1)$$

where $A_{H\times W}$ is the feature matrix output from CA module 1, $Y_{H\times W}$ is the word vector matrix of the code text, H is the number of words in the code text, W is the number of dimensions of the word vector [19], $Z_{H\times W}$ is the vector product of $A_{H\times W}$ and $Y_{H\times W}$, $B_{H\times 1}$ is the feature matrix output from CA module 2, $O_{H\times W}$ is the final output feature matrix of the CA mechanism, $Maxpool_{1D}(\cdot)$ is the one-dimensional global maximum pooling, $Avgpool_{1D}(\cdot)$ is the one-dimensional global mean pooling, $Maxpool_y(\cdot)$ is the global maximum pooling along the y axis, $Avgpool_y(\cdot)$ is the global mean pooling along the y axis, $MLP(\cdot)$ is the two-layer neural network, $conv_{7\times 7}(\cdot)$ is the convolution operation using the 7×7 specification filter, and $f(\cdot)$ is the sigmoid activation function.

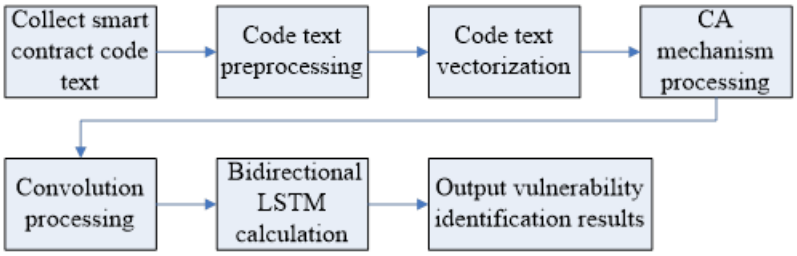


Figure 2: Vulnerability identification process for smart contracts based on deep learning

5) Longitudinal convolutional computation is performed on $O_{H\times W}$ using a convolutional kernel in a $3 \times W$ specification. The Relu activation function is used.

6) The extracted convolutional features are input into the bidirectional LSTM network for computation. Each row of the convolutional features obtained by the vertical convolution in the previous step is the partial feature vector of the word, and each column is the dimension of the feature vector. The input to the bi-directional LSTM network is in "row" sequence. Unlike the traditional LSTM where features are input in one-way sequence, the bidirectional LSTM in this paper inputs features in forward sequence and reverse sequence at the same time, which simply means that the bidirectional LSTM contains two groups of LSTM units, one group is responsible for calculating the forward input sequence and the other group is responsible for calculating the reverse input sequence [21], and then the output is calculated by combining the forward and reverse hidden layer states. formula is:

$$\begin{cases} y_t = g(\omega_1 \overrightarrow{h_t} + \omega_2 \overleftarrow{h_t} + b_y) \\ \overrightarrow{h_t} = g(\omega_3 x_t + \omega_4 \overrightarrow{h_{t-1}} + \overrightarrow{b}) \\ \overleftarrow{h_t} = g(\omega_5 x_t + \omega_6 \overleftarrow{h_{t-1}} + \overleftarrow{b}) \end{cases} \tag{2}$$

where y_t is the output, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ is the forward and reverse LSTM hidden states at the current moment, respectively, x_t is the input sequence at the current moment, ω_1 and ω_2 are the weights of $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, respectively, b_y is the corresponding bias, ω_3 and ω_4 are the weights of x_t and $\overrightarrow{h_{t-1}}$ in the forward LSTM hidden layer, $\overrightarrow{b}$ is the corresponding bias, ω_5 and ω_6 are the weights of x_t and $\overleftarrow{h_{t-1}}$ in the reverse LSTM hidden layer, and $\overleftarrow{b}$ is the corresponding bias.

If it is in the training phase of the algorithm, the cross-entropy loss between the predicted vulnerability label and the actual vulnerability label is calculated as the computational error of the algorithm. If the error converges to the preset range, the training is completed; if the error does not converge to the preset range, the weight parameters are adjusted according to the error in reverse.

3 Simulation Experiments

3.1 Experimental Data

A crawler was used to crawl smart contracts from the browsing website of Ether, a blockchain platform. The crawler started from the initial web page of Ethereum to traverse all the uniform resource locators (URLs) of the pages, and then collected the addresses, version numbers, source codes and bytecodes of smart contracts by the URLs' smart contract addresses. According to the statistics, the crawler crawled 35,890 smart contracts. Then, the tools and manual methods were combined to mark the vulnerabilities of smart contracts. A smart contract may contain more than one type of vulnerabilities, so there were 12,560 contracts containing integer overflow vulnerabilities, 13,480 contracts containing unchecked return value vulnerabilities, 10,115 contracts containing transaction order vulnerabilities, 10,115 contracts containing timestamp vulnerabilities, 5,890 contracts with timestamp vulnerabilities, and 860 contracts with re-entry vulnerabilities.

3.2 Experimental Setup

The relevant parameters of the deep learning-based smart contract vulnerability identification algorithm used in this paper are as follows. The number of dimensions of the word2vec word vector was set to 256 by the orthogonal experiment. The moving step length of the pooling frame in the CA mechanism module was set to 1. The moving step length of the convolution kernel in the convolution calculation in Step 5 was set to 1. Two hundred and fifty-six nodes were set in the LSTM hidden layer in Step 6, and the activation function was set to tahn. The Adam optimization algorithm was used to adjust the weight parameters during the training process, the learning rate was set to 0.001, the training batch size was set to 200, and the epoch was set to 100.

In order to verify the effectiveness and feasibility of the smart contract vulnerability identification algorithm in this paper, two other vulnerability identification algorithms were also simulated and tested. The other two algorithms were the SVM-based vulnerability identification algorithm and the traditional LSTM-based vulnerability identification algorithm, respectively.

The SVM-based vulnerability identification algorithm for vulnerability identification first converted the bytecode of a smart contract into an opcode and simplifies it, then extracted the features of the simplified opcode using an n-gram (n=2), and then constructed the n-gram features of the smart contract as a feature matrix. Each row in the feature matrix was a smart contract, and each column represented an n-gram feature in each contract. After that, the SVM algorithm was used to classify and identify vulnerabilities based on the feature matrix of smart contracts. The relevant parameters of the SVM algorithm were sigmoid kernel function and a penalty factor of 1.

The traditional LSTM-based vulnerability identification algorithm for identifying smart contract code followed similar steps. It first used word2vec to vectorize the pre-processed smart contract code and then directly input it into the LSTM in a "row" sequence. The traditional LSTM algorithm only computed the sequence in one direction. The number of dimensions of word2vec word vector was also set to 256, and the parameters in the LSTM were the same as the LSTM part of the algorithm proposed in this paper. The Adam algorithm was also used to adjust the weight parameters during training, the learning rate was set to 0.001, the training batch size was set to 200, and the epoch was set to 100.

3.3 Evaluation Criteria

The vulnerability identification algorithm for smart contracts is a classification algorithm, so the performance of the vulnerability identification algorithm can be assessed using the confusion matrix (Table 1). When judging whether vulnerabilities in smart contracts can be identified, the binary-classification evaluation criteria is used, and the formula is:

$$\begin{cases} P = \frac{TP}{TP+FP} \\ R = \frac{TP}{TP+FN} \\ F = \frac{2 \cdot P \cdot R}{P+R} \end{cases} \tag{3}$$

where P is the precision, R is the recall rate, and F is the harmonic mean of the precision and recall rate. However, there are various types of vulnerabilities in smart contracts, and the vulnerability identification algorithm also needs to identify the types of vulnerabilities, which requires the multi-classification evaluation criteria. The formula is:

$$\begin{cases} P_{macro} = \frac{\sum_{i=1}^{n} P_i}{n} \\ R_{macro} = \frac{\sum_{i=1}^{n} R_i}{n} \\ F_{macro} = \frac{2 \cdot P_{macro} \cdot R_{macro}}{P_{macro}+R_{macro}} \end{cases} \tag{4}$$

where P_{macro} is the macro precision rate, R_{macro} is the macro recall rate, F_{macro} is the macro harmonic mean, P_i is the recognition precision rate for the i category, R_i is the recognition recall rate for the i category, and n is the number of categories.

3.4 Experimental Results

Among the three vulnerability identification algorithms tested in the simulation experiment, the SVM-based algorithm used the training set to fit the "hyperplane". Figure 3 shows the convergence curves of the traditional LSTM-based algorithm and the vulnerability identification algorithm proposed in this paper during the training process. From Figure 3, it can be seen that the cross-entropy loss of both vulnerability identification algorithms in predicting vulnerabilities decreased to stability with the increase of training times. The traditional LSTM-based algorithm converged to stability after about 60 times of training, and the vulnerability identification algorithm proposed in this paper converged to stability after about 40 times of training. The cross-entropy loss of the vulnerability identification algorithm proposed in this paper was lower than that of the traditional LSTM-based algorithm after stabilization.

Figure 4 shows the binary performance of three vulnerability identification algorithms for smart contract vulnerabilities, i.e., the performance of the vulnerability identification algorithm to determine whether there is a vulnerability in a smart contract. From Figure 4, it can be seen that the vulnerability identification algorithm proposed in this paper had the best performance in identifying vulnerabilities, the traditional LSTM-based algorithm had the second best performance, and the SVM-based vulnerability identification algorithm had the worst performance.

In the actual identification process of vulnerability mining, the vulnerability identification algorithm should not only determine whether there is a vulnerability in the smart contract, but also determine the type of vulnerability, i.e., the multi-classification performance of the vulnerability identification algorithm, as shown in Table 2. As can be seen from Table 2, for each vulnerability type, the vulnerability identification algorithm proposed in this paper had the best identification performance, followed by the traditional LSTM algorithm, and the SVM algorithm was the worst. This is also true for the overall macro indicators. Compared to the performance indicators of binary classification, the identification performance of the SVM-based vulnerability identification algorithm was reduced, while the traditional LSTM-based vulnerability identification algorithm and the vulnerability identification algorithm proposed in this paper were basically unaffected.

4 Conclusion

Based on the similarity between smart contract code and text sequences, this paper applied the LSTM algorithm in deep learning for vulnerability mining and identification in smart contracts. To strengthen the performance of the vulnerability detection algorithm, the word2vec word vector was combined with the mutual information values of keywords. The critical features were highlighted using the CA mechanism, and the bidirectional LSTM algorithm was used to incorporate contextual information.

Table 1: Confusion matrix

	Predicted as positive examples	Predicted as negative examples
True positive examples	TP	FN
True negative examples	FP	TN

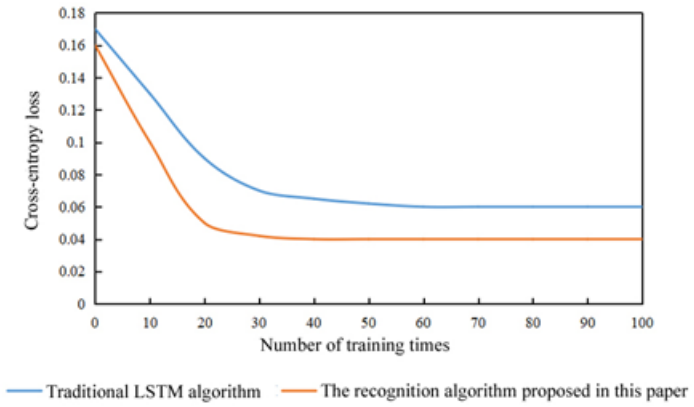


Figure 3: Training curves of the traditional LSTM vulnerability identification algorithm and the vulnerability identification algorithm proposed in this paper

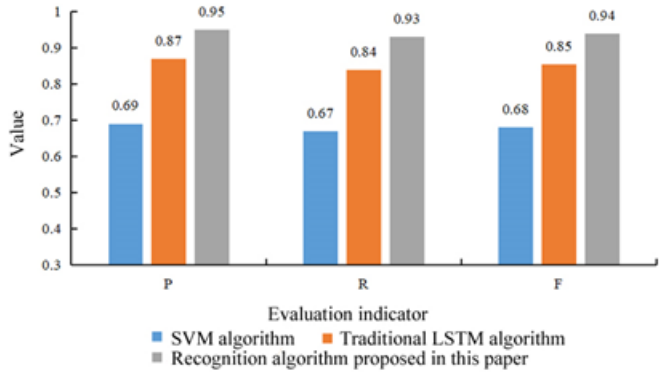


Figure 4: Binary classification performance of three vulnerability identification algorithms for smart contract vulnerabilities

Table 2: Multi-classification performance of three vulnerability identification algorithms for smart contract vulnerabilities

Vulnerability type	SVM			Traditional LSTM			Our Algorithm		
	P	R	F	P	R	F	P	R	F
Integer overflow	0.59	0.58	0.58	0.79	0.78	0.78	0.96	0.95	0.95
Unchecked return value	0.64	0.63	0.63	0.86	0.85	0.85	0.95	0.94	0.94
Trading order	0.63	0.61	0.62	0.89	0.88	0.88	0.94	0.92	0.93
Timestamp	0.61	0.6	0.60	0.87	0.86	0.86	0.95	0.91	0.93
Reentry	0.59	0.6	0.59	0.82	0.83	0.82	0.93	0.93	0.93
Overall	0.61	0.60	0.61	0.85	0.84	0.84	0.95	0.93	0.94

The algorithm was then compared with SVM and traditional LSTM algorithms through simulated experiments. Finally, the following results were obtained. The vulnerability detection algorithm based on traditional LSTM converged and stabilized after about 60 training iterations, while the proposed algorithm converged and stabilized after about 40 iterations. In addition, the proposed algorithm exhibited lower cross-entropy loss for vulnerability detection when stabilized. The proposed vulnerability detection algorithm performed best, followed by the traditional LSTM algorithm and the SVM algorithm in binary classification of smart contract vulnerabilities. (3) The proposed vulnerability detection algorithm also outperformed better than the traditional LSTM and SVM algorithms in multi-classification of smart contract vulnerabilities. In addition, the multi-classification performance of the SVM algorithm was slightly degraded compared to the binary classification performance, while the other two methods showed no significant impact.

References

[1] M. A. Albahar, "A modified maximal divergence sequential auto-encoder and time delay neural network models for vulnerable binary codes detection," *IEEE Access*, vol. 8, pp. 14999-15006, 2020.

[2] N. Antunes, M. Vieira, "Assessing and comparing vulnerability detection tools for web services: Benchmarking approach and examples," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 269-283, 2015.

[3] P. Y. Chang, M.S. Hwang, C.C. Yang, "A blockchain-based traceable certification system", in *Security with Intelligent Computing and Big-data Services*, pp. 363-369, 2018.

[4] L. Chen, Q. Fu, Y. Mu, L. Zeng, F. Rezaeibagha, M. S. Hwang, "Blockchain-based random auditor committee for integrity verification", *Future Generation Computer Systems*, vol. 131, pp. 183-193, 2022.

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

[6] Y. H. Chen, L. C. Huang, I. C. Lin, M. S. Hwang, "Research on blockchain technologies in bidding systems", *International Journal of Network Security*, vol. 22, no. 6, pp. 897-904, 2020.

[7] J. Hu, J. Chen, L. Zhang, L. Zhang, Y. Liu, Q. Bao, H. Ackah-Arthur, C. Zhang, "A memory-related vulnerability detection approach based on vulnerability features," *Journal of Tsinghua University: Natural Science Edition*, vol. 25, no. 5, pp. 604-613, 2020.

[8] S. H. Lee, S. M. Shin, J. S. Hwang, J. Park, "Operational vulnerability identification procedure for nuclear facilities using STAMP/STPA," *IEEE Access*, vol. 8, pp. 166034-166046, 2020.

[9] Z. Li, D. Zou, J. Tang, Z. Zhang, M. Sun, H. Jin, "A comparative study of deep learning-based vulnerability detection system," *IEEE Access*, vol. 7, pp. 103184-103197, 2019.

[10] Z. Li, D. Zou, S. Xu, X. Ou, H. Jin, S. Wang, Z. Deng, Y. Zhong, "VulDeePecker: A deep learning-based system for vulnerability detection," in *Network and Distributed System Security Symposium*, 2018.

[11] C. Y. Lin, L. C. Huang, Y. H. Chen, M. S. Hwang, "Research on security and performance of blockchain with innovation architecture technology", *International Journal of Network Security*, vol. 23, no. 1, pp. 1-8, 2021.

[12] Z. P. Peng, M. L. Chiang, I. C. Lin, C. C. Yang, M. S. Hwang, "CBP2P: Cooperative electronic bank payment systems based on blockchain technology", *Journal of Internet Technology*, vol. 23, no. 4, pp. 683-692, 2022.

[13] A. Qasem, P. Shirani, M. Debbabi, L. Wang, B. Lebel, B. L. Agba, "Automatic vulnerability detection in embedded devices and firmware: Survey and layered taxonomies," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1-42, 2022.

[14] L. Sampaio, A. Garcia, "Exploring context-sensitive data flow analysis for early vulnerability detection," *Journal of Systems & Software*, vol. 113, pp. 337-361, 2016.

[15] N. F. Samreen, M. H. Alalfi, "Reentrancy vulnerability identification in ethereum smart contracts," in *IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE'20)*, 2020.

[16] K. Shi, Y. Dai, J. Xu, "Construction of a security vulnerability identification system based on machine learning," *Journal of Sensors*, vol. 2020, no. 2, pp. 1 -9, 2020.

[17] G. Tang, L. Yang, S. Ren, L. Meng, F. Yang, H. Wang, "An automatic source code vulnerability detection approach based on KELM," *Security and Communication Networks*, vol. 2021, no. 1, pp. 1-12, 2021.

[18] J. X. Tong, H. Li, and S. L. Yin, "Research on face recognition method based on deep neural network," *International Journal of Electronics and Information Engineering*, vol. 12, no. 4, pp. 182–188, 2020.

[19] X. Wang, Q. Zheng, K. Zheng, K. Zheng, Y. Sui, J. Zhang, "Semi-GSGCN: Social robot detection research with graph neural network," *Computers, Materials, and Continua*, no. 10, pp. 617-638, 2020.

[20] Y. Wang, Z. Wu, Q. Wei, Q. Wang, "NeuFuzz: Efficient fuzzing with deep neural network," *IEEE Access*, vol. 7, pp. 36340-36352, 2019.

[21] L. Xie, D. Pi, X. Zhang, J. Chen, Y. Luo, W. Yu, "Graph neural network approach for anomaly detection," *Measurement*, vol. 180, no. 1, pp. 1-11, 2021.

[22] X. Zhan, L. Fan, S. Chen, F. Wu, T. Liu, X. Luo, Y. Liu, "ATVHunter: Reliable version detection of third-party libraries for vulnerability identification in

android applications," in *Proceedings of the 43rd International Conference on Software Engineering*, pp. 1695-1707, 2021.

Biography

Ban Lili graduated from Hebei Normal University in 2015, worked at Hebei Academy of Fine Arts, studied at Mahashalakan University in Thailand, Engaged in computer science and educational informatization research.

Li Yan graduated from Tianjin Normal University in 2012 and worked at Hebei Academy of Fine Arts, specializing in computer information management, computer network technology, and educational technology.

An Image Denoising Fractional-Order Model with Coupling of Fidelity Terms

Donghong Zhao, Xinyao Yu, and Haoyu Liu
(Corresponding author: Donghong Zhao)

Department of Applied Mathematics, School of Mathematics and Physics, University of Science and Technology Beijing
30, Xueyuan Road, Haidian District, Beijing 100083, China
Email: zdh751111@ustb.edu.cn

(Received Nov. 21, 2022; Revised and Accepted July 11, 2023; First Online Aug. 25, 2023)

Abstract

Network security is becoming increasingly important due to the development of networks. In the network, digital image is an important channel for information transmission, where noise will be generated. Image denoising is the most fundamental and core problem in image processing. The classical Total Variation (TV) denoising model has advantages and shortcomings. Based on this, many improved models have been developed to overcome the staircase effect and show advantages on TV models, such as high-order and adaptive TV models. This paper aims to propose a model that can remove noise well, maintain image details, and overcome the staircase effect. It is also hoped to find a better and more efficient numerical simulation algorithm to process the model. Based on the Fractional Total Variation (FTV) model, an improved fractional-order TV model is proposed with the coupling of fidelity terms. The isotropic and anisotropic denoising models are studied, and the new model is compared with the classical FTV model for image denoising.

Keywords: Coupling of Fidelity Terms; Euler-Lagrange Equation; Fractional-Order Total Variation; Image Denoising; Split Bregman Iteration

1 Introduction

Network security is becoming more and more important. From the daily transmission of individual information to national security, network security is inseparable. Network security is a well-known focus. Digital image technology is an important medium in the network, and image processing technology is closely related to network security. Zhang *et al.* (1999) proposed a block based digital watermarks for copy protection of images [6]. This method can resist various attacks, such as blurring, loss compression, cropped and scaling. This watermarked image is called a stego-image. Lu *et al.* (2003) developed a novel fragile watermarking scheme for image authentication to against the quantization attack. Unfortunately, their scheme is not secure enough. Liao *et al.* (2006) proposed two types of attack based on Lu *et al.*, and pointed out that the scheme was still impractical. To achieve the copyright protection and tamper detection of stego-image, a robust-fragile watermarking algorithm is proposed by Wu *et al.* (2008) [24]. Yang *et al.* (2007) showed that the Lin-Tsai scheme has three weaknesses. They not only improved the authentication ability and image quality of stego-image, but also introduced a lossless image sharing scheme for secret images. Unfortunately, the embedding algorithm was not optimal. Furthermore, a novel secret image sharing scheme based on the simple LSB substitution and an optimal pixel adjustment process was proposed [23]. Then, a lossless reversible secret image sharing scheme was proposed by Wu *et al.* (2009). Consequently, lossless restoration of original images, secret images, and cover images has been achieved [25]. In addition, Huang *et al.* studied data hiding in medical images [11]. All of these require image transmission and processing, and noise will be generated.

Due to the influence of equipment, environmental and human factors, images often contain motion blur and Gaussian white noise, which affect the acquisition of image information by machines and humans [12]. Optimal denoising techniques need to preserve important image features, such as edges and textures, while removing noise. Mathematically, image denoising can be expressed as an estimation problem, i.e. finding an approximate of the original image u from the noise image u_0. The process of image degradation is the process of noise generation. This can be expressed in the form of a raw image function and an additive noise term, denoted as follows:

$$u_0 = u + n \tag{1}$$

where n is the Gaussian noise.

In the past decades, various denoising methods have been proposed, such as Wavelet Transforms, Partial Differential Equations (PDE), Fourier Transforms, and TV methods. Rudin *et al.* proposed one of the most famous television models in 1992, the ROF model, which treats

the denoising problem as a minimization problem [19]:

$$\min_u = \int_\Omega |\nabla u| \, \mathrm{d}\Omega + \frac{\lambda}{2} \|u - u_0\|_2^2 \tag{2}$$

In Equation (2), the first term is often referred to the regularization term, which is responsible for the sharpness of edges. The second term is the fidelity term that contributes to control the similarity between image u and u_0. The ROF model can achieve a good trade-off between edge-preserving and noise removal, but it produces piecewise constant solutionsthat tends to lead to blocky effects, and tend to filter out small details during denoising.

The image denoising model of partial differential equation has a good mathematical theoretical foundation. Itislocallyadaptable and high flexibility [30]. This method can effectively remove noise from the image, resulting in better performance of evaluation measures, such as Peak Signal-to-Noise Ratio (PSNR), Entropy and Mean Square Error(MSE) of the processed image, but the average running time of thealgorithm is long [10]. To reduce the detrimental effects of the denoising process, many new mathematical methods, such as high-order and fractional-order denoising models have introduced in the past few years [20]. Fractional-order differential operator is rapidly developing in many fields, and arbitrary-order differential equations have been extensivelystudied in physics, fluid mechanics, physiology and engineering [21, 22]. In image denoising, the integer-order differential operatorsareonly suitable for the high-frequency part of the image, and cannot retainthe discontinuous boundary points and low-frequency variation infeature details. However, texture details in the image belong to low and medium frequency components. The fractional-order differential can nonlinearly preserve low-frequency features in asmooth area of an image, and nonlinearly enhance the high-frequency edges and textural details with large or insignificant gray-scale variations [22, 27, 29]. Aclass of fractional-order anisotropic diffusion models and a fractional-order TV-L2 model were introduced for removing the noise [7]. The discrete Fourier transform and optimization-minimization (MM) algorithm [8] are used and solved by the conjugate gradient method.Combined the FTV model with the TV model, a fractional-order TV regularization function was proposedfor image processing [18].

Taking into account the long-term memory and non-locality of fractional order equations, the idea of complementary gradient fidelity terms and the advantages of the TV model are considered [17]. Here, a combined model with coupling of fractional-order fidelity termand global fidelity term is proposed to make the image bright in noiseremoving. Experimental results show that compared with the previously mentioned classical FTV model, the proposed model is robust in terms of visual improvement and PSNR.

The rest of the article is organized as:In Section 2, the fractional-order model and the Split Bregman Iteration algorithm are briefly introduced; in Section 3,the model and its detailed analysis are presented. In Section 4, comparative experiments are carried out to verify that the proposedmodel and method have better denoising effects. Section 5 gives the conclusion.

2 Preliminary

2.1 Models: From Integer-Order to Fractional-Order

In this section, the TV model and the fractional-order model are briefly reviewedfor the additive noise removal problem.

The TV based image denoising model is the most well-knownmodelproposed by Rudin *et al.* [19].

$$\min_u = \int_\Omega |\nabla u| \, \mathrm{d}\Omega + \frac{\lambda}{2} \|u - u_0\|_2^2 \tag{3}$$

$\Omega \subset R^2$denotes the image domain. More precisely, the penalty function is as follows:

$$\int_\Omega |\nabla u| \, \mathrm{d}\Omega = \sup \left\{ \int_\Omega u \, div\varphi \, \mathrm{d}\Omega | \varphi \in C_c^1 \left(\Omega, R^2\right), |\varphi| \leq 1 \right\} \tag{4}$$

Formally,

$$\hat{u} = \arg\min \left\{ \int_\Omega |\nabla u| \, \mathrm{d}\Omega + \frac{\lambda}{2} \|u - u_0\|_2^2 \right\} \tag{5}$$

is used toapproximately replace Equation (3).

However, there is also staircase effect in Equation (5). To deal with this problem, the FTV model [2] is proposed as follows:

$$\min_u = \int_\Omega |\nabla^\alpha u| \, \mathrm{d}\Omega + \frac{\lambda}{2} \|u - u_0\|_2^2 \tag{6}$$

where $\nabla^\alpha u = \left(\nabla_x^\alpha u, \nabla_y^\alpha u\right)^T$.

Fractional-order differentiation has many different definitions. According tothe Grünwald-Letnikov fractional derivative definition [16], the fractional-order differentiation can be obtained by

$$\begin{aligned} \nabla_x^\alpha u_{i,j} &= \sum_{k=0}^{i-1} (-1)^k C_k^\alpha u_{i-k,j}, \nabla_y^\alpha u_{i,j} \\ &= \sum_{k=0}^{j-1} (-1)^k C_k^\alpha u_{i,j-k} \end{aligned} \tag{7}$$

where $C_k^\alpha = \frac{\Gamma(\alpha-1)}{\Gamma(k+1)\Gamma(\alpha-k+1)}$ denotes the generalized binomial coefficient, and $\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-1} \mathrm{d}t \, (x > 0)$ is the Gamma function.

The fractional-order gradient $\nabla^\alpha u_{i,j} \, (i, j = 1, 2, \ldots, n)$

can be expressed as follows:

$$\nabla^{\alpha}u_{i,j}=\begin{cases}\left(\sum_{k=0}^{i-1}(-1)^{k}C_{k}^{\alpha}u_{i-k,j},\sum_{k=0}^{j-1}(-1)^{k}C_{k}^{\alpha}u_{i,j-k}\right)^{T}\\ if\ \ i>1\ \ and\ \ j>1\\ \left(0,\sum_{k=0}^{j-1}(-1)^{k}C_{k}^{\alpha}u_{i,j-k}\right)^{T}\\ if\ \ i=1\ \ and\ \ j>1\\ \left(\sum_{k=0}^{i-1}(-1)^{k}C_{k}^{\alpha}u_{i-k,j},0\right)^{T}\\ if\ \ i>1\ \ and\ \ j=1\\ (0,0)^{T}\\ if\ \ i=1\ \ and\ \ j=1\end{cases}\tag{8}$$

2.2 The Split Bregman Iteration

Some definitions of the Split Bregman algorithm are reviewed.

First, the Bregman distance [3] has to be introduced, which is associated with a convex function J given by

$$D_{J}^{p}(u,v)=J(u)-J(v)-\langle p,u-v\rangle\tag{9}$$

where $u,v\in J$ and p is the sub-gradient of J at v. Clearly, it is not a distance in the general sense, since it is not symmetric in thegeneral sense, but it does measure the proximity of u and v. The conception is mainly used to solve the constrained optimization problem:

$$\min_{u}J(u)+H(u)\tag{10}$$

where J and H are convex functions defined on R^{n}, and H is differentiable. As shown in [8,18], Bregman iterative formula is as follows:

$$\begin{aligned}u^{k+1}&=\min_{u}D_{J}^{p^{k}}\left(u,u^{k}\right)+H(u)\\&=\min_{u}J(u)-J\left(u^{k}\right)-\left\langle p^{k},u-u^{k}\right\rangle+H(u)\\p^{k+1}&=p^{k}-\nabla H\left(u^{k+1}\right)\end{aligned}\tag{11}$$

where ∇H represents the gradient of $H(u)$.

Goldstein and Osher [9] improved the Bregman algorithms. They pointed out that the difficulty in solving the ROF model lies in the non-differentiability of TV subnorms. The technique 'de-coupling' in the split Bregman algorithm is to split the L1 and L2 parts from the objective energy function, which is the key to this method. The Split Bregman algorithm is applied to the following constrained minimization problem:

$$\min_{u}|d|+H(u),such\ \ that\ \ d=J(u)\tag{12}$$

Using the quadratic penalty function, Equation (12) is changed into an unconstrained problem:

$$\min_{u}|d|+H(u)+\frac{\gamma}{2}\|d-J(u)\|_{2}^{2}\tag{13}$$

Then,

$$\begin{aligned}\left(u^{k+1},d^{k+1}\right)=&\min_{u,d}D_{J}^{p}\left(u,u^{k},d,d^{k}\right)+\frac{\gamma}{2}\|d-J(u)\|_{2}^{2}\\=&J(u)+J(d)-J\left(u^{k}\right)-J\left(d^{k}\right)\\&-\left\langle p_{u}^{k},u-u^{k}\right\rangle-\left\langle p_{d}^{k},d-d^{k}\right\rangle\end{aligned}\tag{14}$$

$$p_{u}^{k+1}=p_{u}^{k}-\gamma\left(J\left(u^{k+1}\right)-d^{k+1}\right)\tag{15}$$

$$p_{d}^{k+1}=p_{d}^{k}-\gamma\left(d^{k+1}-J\left(u^{k+1}\right)\right)\tag{16}$$

Equation (14)-Equation (16) can be reduced to the following two-stage algorithm:

$$\left(u^{k+1},d^{k+1}\right)=\min_{u,d}|d|+H(u)+\frac{\gamma}{2}\left\|d-J(u)-b^{k}\right\|_{2}^{2}\tag{17}$$

$$b^{k+1}=b^{k}+\left(J\left(u^{k+1}\right)-d^{k+1}\right)\tag{18}$$

Using the Split Bregman algorithm, Equation (17) can bedivided into two sub-problems:

Sub-problem 1:

$$u^{k+1}=\min_{u}H(u)+\frac{\gamma}{2}\left\|d^{k}-J(u)-b^{k}\right\|_{2}^{2}\tag{19}$$

Sub-problem 2:

$$d^{k+1}=\min_{d}|d|+\frac{\gamma}{2}\left\|d-J\left(u^{k+1}\right)-b^{k}\right\|_{2}^{2}\tag{20}$$

For Sub-problem 1, a variety of optimization techniques can be used to solve this problem. The exact method used to solve this optimization problem depends on the nature of H. The Gauss-Seidel or Fourier transform can be used for many common problems. The problem of minimizing J can be approximated by several steps of the conjugate gradient method. In Sub-problem 2, there is no coupling between the elements of d, and the optimal value of d can be calculated explicitly using the contraction operator. Here, it can be simply calculated as follows:

$$d^{k+1}=shrink\left(J(u)+b^{k},\frac{1}{\gamma}\right)\tag{21}$$

where $shrink(x,\beta)=\frac{x}{|x|}*\max(|x|-\beta,0)$.

The results are obtained when the iterative minimization scheme is put into the process described in Equation (17).

Then the above ideas will be used to solve the proposed model in Section 3.

3 The Proposed Model

Classical models often have regularization term and fidelity terms to control image denoising. Many novel mathematical methods have been introduced to further overcome those disadvantages.

The selective smoothing method is proposed [4], and Alvarez-Lions-Morel improved the PM model [1]. You and Kaveh [28] proposed a fourth-order partial differential equation denoising model by minimizing the energy function of the second-order derivative. Chan *et al.* [5] introduced a modified high-order TV model and added a nonlinear fourth-order diffusive term to the Euler–Lagrange

equations ofthe TV model. Then, Lysaker *et al.* proposed a new approach based on a fourth-order PDE model (LLT) for image denoising, and this approach was tested on a series of medical magnetic resonance images [13]. Lysaker and Tai [14] combined TV minimization with the second-order functional, and Zhu and Xia [31] introduced the gradient fidelity term:

$$E(u) = \int_{\Omega} \alpha (u-u_0)^2 \,\mathrm{d}x\mathrm{d}y + \|\nabla u - \nabla (G_\sigma \otimes u_0)\|_2^2 \tag{22}$$

However, in this model, the authors discarded some regular terms that were critical to traditional denoising models, resulting in poor processing results.The image is preprocessed by Gaussian filtering, so that the gradient of is close to the estimated gradient value, which improves the proximity of image. This approach reduces the staircase effect to a certain extent. Combined with gradient fidelity term model (TVGF), Xiao *et al.* [26] corrected the erroneous analysis and proposed a new theoretical analysis of the TV model, which has a significant effect onslowing down the staircase effect.

$$\min_u \int_{\Omega} |\nabla u| + \frac{\lambda}{2}\|u-u_0\| + \frac{\mu}{2}\|\nabla u - \nabla (G_\sigma \otimes u_0)\|_2^2 \tag{23}$$

The model does have significant improvements and its structure is theoretically more reasonable. However, this model generates more blurring around the edges.As such, some scholars recommend using an improved fractional-order gradient fidelity [15] to replace the gradient fidelity in Equation (23) such as

$$\min_u \int_{\Omega} |\nabla u| + \frac{\lambda}{2}\|u-u_0\|_2^2 + \frac{\mu}{2}\|\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)\|_2^2$$

This model combines a stepwise fidelity term and a global fidelity term. The coupled gradient fidelity term is used to obtain the nonlinear diffusion method, which is based on the long-term memory and non-locality of fractional differential equations. It can prevent the effect of staircase and enhances the intricate artistic details of smooth areas. The image becomes sharper and brighter.

Fractional-order calculus has been successfully applied to various fields of image processing, with optimalresults. In image denoising, the integer-order differential operatorsare only suitable for the high-frequency part of the image, and lack the ability to retain the discontinuous boundary points and low-frequency variation details. Image texture details are medium and low frequency componentsandthe integer-order difference operators cannot handle it well. Contraryto the nature of integer-order calculus, the fractional-order calculus has a non-zero differentiation at a constant. It can nonlinearly maintain low-frequency features in smooth regions of the image and enhance high-frequency edges and texture details in the regions where grayscale varies greatly or insignificantly. Considering the advantages of fractional-order models, the classical fractional-order model and the fractional-order gradient fidelity term are fitted. A fractional-order model with coupling of fidelity terms is proposed,

$$\begin{aligned} E(u) =& E_r(u) + \lambda E_{f_1} + \mu E_{f_2} \\ =& \int_{\Omega} \|u\|_{FTV} + \frac{\lambda}{2}\|u-u_0\|_2^2 \\ &+ \frac{\mu}{2}\|\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)\|_2^2 \,\mathrm{d}\Omega \\ =& \int_{\Omega} |\nabla^\alpha u| + \frac{\lambda}{2}\|u-u_0\|_2^2 \\ &+ \frac{\mu}{2}\|\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)\|_2^2 \,\mathrm{d}\Omega \end{aligned} \tag{24}$$

where λ and μ are two positive parameters that control the balance between global fidelity and gradient fidelity. E_r, E_{f_1}, and E_{f_2} are the fractional-order TV regularization term, global fidelity term and fractional order gradient fidelity term, respectively. The energy function would be discussed according todifferent definitions of the regular term.

3.1 Isotropic TV Denoising

3.1.1 Fractional-Order Isotropic TV Regularization

The fractional TV regular termis defined as:

$$E_r(u) = \int_{\Omega} \|u\|_{FTV}\,\mathrm{d}\Omega = \int_{\Omega} |\nabla^\alpha u|\,\mathrm{d}\Omega \tag{25}$$

There are two possible definitions of the fractional-order TV $\|u\|_{FTV}$. The first one is the isotropic fractional-order TV, defined as

$$|\nabla^\alpha u| = \|\nabla^\alpha u\|_2 = \sqrt{(\nabla_x^\alpha u)^2 + (\nabla_y^\alpha u)^2} \tag{26}$$

The second definition of the fractional-order TV is called the anisotropic fractional-order TV, which will be introduced in Section 3.2.

According to the definition of Grünwald-Letnikov derivative, the gradient descent flow of the fractional-order TV regularization term E_r is deduced.

$$\begin{aligned} E_r'(u)\,\eta &= \lim_{\varepsilon\to 0} \frac{E_r(u+\varepsilon\eta) - E_r(u)}{\varepsilon} \\ &= \lim_{\varepsilon\to 0} \frac{\int_\Omega |\nabla^\alpha u + \varepsilon\nabla^\alpha \eta|\,\mathrm{d}\Omega - \int_\Omega |\nabla^\alpha u|\,\mathrm{d}\Omega}{\varepsilon} \\ &= \int_\Omega \left(\frac{\nabla_x^\alpha u}{|\nabla^\alpha u|}\nabla_x^\alpha \eta + \frac{\nabla_y^\alpha u}{|\nabla^\alpha u|}\nabla_y^\alpha \eta \right) \mathrm{d}\Omega \\ &= \int_\Omega \left(\nabla_x^{\alpha^*}\left(\frac{\nabla_x^\alpha u}{|\nabla^\alpha u|}\right) + \nabla_y^{\alpha^*}\left(\frac{\nabla_y^\alpha u}{|\nabla^\alpha u|}\right)\right)\eta \mathrm{d}\Omega \end{aligned} \tag{27}$$

where $\nabla_x^{\alpha^*}$ and $\nabla_y^{\alpha^*}$ are the adjoint operators of ∇_x^α and ∇_y^α. The gradient descent flow of E_r is as follows:

$$\frac{\partial u}{\partial t} = -\left(\nabla_x^{\alpha^*}\left(\frac{\nabla_x^\alpha u}{|\nabla^\alpha u|}\right) + \nabla_y^{\alpha^*}\left(\frac{\nabla_y^\alpha u}{|\nabla^\alpha u|}\right)\right) \tag{28}$$

There are two kinds of fidelity terms, which are global fidelity and fractional-order gradient fidelity.

The global fidelity is defined as:

$$E_{f_1} = \int_\Omega \frac{1}{2} \|u - u_0\|_2^2 \, \mathrm{d}\Omega \tag{29}$$

It contributes to control the degree of the approximation between the noisy images u_0 and u.

The gradient descent flow of Equation (29) is as follows:

$$\frac{\partial u}{\partial t} = -(u - u_0) \tag{30}$$

Moreover, the gradient fidelity term is expressed as follows:

$$E_{f_2} = \int_\Omega \frac{1}{2} \|\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)\|_2^2 \, \mathrm{d}\Omega \tag{31}$$

This fidelity term contributes to measuring the similarity in the gradient of images. In other words, it can make the gradient of the recovered image close to that of the estimated image $G_\sigma \otimes u_0$.

The gradient descent flow of E_{f_2} is calculated as follows:

$$\frac{\partial u}{\partial t} = -\nabla^{\alpha^*} (\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)) \tag{32}$$

where $\nabla^{\alpha^*} u$ is the adjoint of $\nabla^\alpha u$.

3.1.2 The Corresponding Iterative Approach

The IFTV-FDF (the Isotropic Fractional-order Total Variation model coupled with the Fractional Differential Fidelity term) model is derived as follows:

$$\begin{aligned} \min_u & \sqrt{\left(\nabla_x^\alpha u\right)^2 + \left(\nabla_y^\alpha u\right)^2} + \frac{\lambda}{2} \|u - u_0\|_2^2 \\ & + \frac{\mu}{2} \|\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)\|_2^2 \end{aligned} \tag{33}$$

According to Equation (29), Equation (30) and Equation (32), the corresponding evolution equation of the Euler-Lagrange equation of the energy function Equation (33) is as follows:

$$\begin{aligned} \frac{\partial u}{\partial t} = & -\left(\nabla_x^{\alpha^*}\left(\frac{\nabla_x^\alpha u}{|\nabla^\alpha u|}\right) + \nabla_y^{\alpha^*}\left(\frac{\nabla_y^\alpha u}{|\nabla^\alpha u|}\right)\right) \\ & - \lambda (u - u_0) - \mu (\nabla^\alpha)^* (\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)) \end{aligned} \tag{34}$$

To numerically approximate Equation (34), $u_{i,j}$ is defined as the value of the image u at pixel (ih, jh), Δt as the time step and $h = 1$ as the space step. The temporal partial derivative $\frac{\partial u}{\partial t}$ can be represented by $\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t}$.

Equation (34) can be further derived by discretization, and expressed as:

$$\begin{aligned} u_{i,j}^{n+1} = & u_{i,j}^n (1 - \lambda \Delta t) + \Delta t[\lambda u_{i,j}^0 \\ & - \left(A_{i,j}^n + \mu\right) \left(\sum_{k=0}^{i-1} (-1)^k C_k^\alpha u_{i-k,j} \right. \\ & \left. + \sum_{k=0}^{j-1} (-1)^k C_k^\alpha u_{i,j-k}\right) + \mu B_{i,j}^n] \end{aligned} \tag{35}$$

where $A_{i,j}^n = \frac{1}{\sqrt{\left(\nabla_x^\alpha u_{i,j}^n\right)^2 + \left(\nabla_y^\alpha u_{i,j}^n\right)^2}}$ and $B_{i,j}^n = \nabla_x^\alpha \left(G_\sigma \otimes u_{i,j}^0\right) + \nabla_y^\alpha \left(G_\sigma \otimes u_{i,j}^0\right)$.

The process of the algorithm is shown as follows:

Algorithm 1 Algorithm of the model IFTV-FDF

1: Initialization. Pick u^0 and choose feasible λ and μ, set $n = 0$.
2: Iteration. For every $i, j = 1, 2, \ldots$ compute $u_{i,j}^{n+1} = u_{i,j}^n (1 - \lambda \Delta t) + \Delta t[\lambda u_{i,j}^0 - \left(A_{i,j}^n + \mu\right) \left(\sum_{k=0}^{i-1} (-1)^k C_k^\alpha u_{i-k,j} + \sum_{k=0}^{j-1} (-1)^k C_k^\alpha u_{i,j-k}\right) + \mu B_{i,j}^n]$, let $\Delta t = 1$, $A_{i,j}^n = \frac{1}{\sqrt{\left(\nabla_x^\alpha u_{i,j}^n\right)^2 + \left(\nabla_y^\alpha u_{i,j}^n\right)^2}}$, $B_{i,j}^n = \nabla_x^\alpha \left(G_\sigma \otimes u_{i,j}^0\right) + \nabla_y^\alpha \left(G_\sigma \otimes u_{i,j}^0\right)$
3: If u^n satisfies the stopping criterion, terminate the iteration and output; otherwise, go to step 2.

3.2 Anisotropic TV Denoising

3.2.1 The Fractional-Order Anisotropic TV Regularization

In this subsection, the regular terms in the proposed model Equation (25) with the anisotropic fractional-order TV is considered. The definition of the fractional-order TV $\|u\|_{FTV}$ in Equation (26) is expressed as:

$$|\nabla^\alpha u| = \|\nabla^\alpha u\|_1 = |\nabla_x^\alpha u| + \left|\nabla_y^\alpha u\right| \tag{36}$$

The anisotropic fractional-order TV regularization is denoted as:

$$E_r(u) = \int_\Omega |\nabla_x^\alpha u| + \left|\nabla_y^\alpha u\right| \mathrm{d}\Omega \tag{37}$$

3.2.2 The Corresponding Iterative Approach

The AFTV-FDF (the Anisotropic Fractional-order Total Variation modelcoupled with the Fractional Differential Fidelity term) model is derived as:

$$\begin{aligned} \min_u & |\nabla_x^\alpha u| + \left|\nabla_y^\alpha u\right| + \frac{\lambda}{2} \|u - u_0\|_2^2 \\ & + \frac{\mu}{2} \|\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)\|_2^2 \end{aligned} \tag{38}$$

Then the Split Bregman algorithm will be used to solve the model. For the last term of Equation (38), the following derivation is given:

$$\begin{aligned} \nabla^\alpha u &= \left(\nabla_x^\alpha u, \nabla_y^\alpha u\right) \\ \nabla^\alpha (G_\sigma \otimes u_0) &= \left(\nabla_x^\alpha (G_\sigma \otimes u_0), \nabla_y^\alpha (G_\sigma \otimes u_0)\right) \end{aligned}$$

$$
\begin{aligned}
&\left\|\nabla^{\alpha}u-\nabla^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2\\
&=\left\|\left(\nabla_x^{\alpha}u,\nabla_y^{\alpha}u\right)-\left(\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right),\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right)\right\|_2^2\\
&=\left\|\left(\nabla_x^{\alpha}u-\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right),\nabla_y^{\alpha}u-\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right)\right\|_2^2\\
&=\left[\sqrt{\left(\nabla_x^{\alpha}u-\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right)^2+\left(\nabla_y^{\alpha}u-\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right)^2}\right]^2\\
&=\left(\nabla_x^{\alpha}u-\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right)^2+\left(\nabla_y^{\alpha}u-\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right)^2\\
&=\left\|\nabla_x^{\alpha}u-\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2+\left\|\nabla_y^{\alpha}u-\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2
\end{aligned}
$$

Equation (38) is equivalent to the following constrained problem:

$$
\begin{aligned}
\min_u |d_1|+|d_2|+\frac{\lambda}{2}\left\|u-u_0\right\|_2^2+\frac{\mu}{2}\left\|d_1-\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2\\
+\frac{\mu}{2}\left\|d_2-\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2\\
s.t.\ d_1=\nabla_x^{\alpha}u, d_2=\nabla_y^{\alpha}u
\end{aligned}
$$

We can change it into the following unconstrained problem:

$$
\begin{aligned}
\min_u |d_1|+|d_2|+\tfrac{\lambda}{2}\left\|u-u_0\right\|_2^2\\
+\tfrac{\mu}{2}\left\|d_1-\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2+\tfrac{\mu}{2}\left\|d_2-\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2\\
+\tfrac{\gamma}{2}\left\|d_1-\nabla_x^{\alpha}u\right\|_2^2+\tfrac{\gamma}{2}\left\|d_2-\nabla_y^{\alpha}u\right\|_2^2
\end{aligned}
\tag{39}
$$

Then the Split Bregman Iteration Equation (17) and Equation (18) are applied to obtain:

$$
\begin{aligned}
\left(u^k,d^k\right) &= \arg\min_{u,d}\{|d_1|+|d_2|+\frac{\lambda}{2}\left\|u-u_0\right\|_2^2\\
&\quad+\frac{\mu}{2}\left\|d_1-\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2\\
&\quad+\frac{\mu}{2}\left\|d_2-\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2\\
&\quad+\frac{\gamma}{2}\left\|d_1-\nabla_x^{\alpha}u-b_1^k\right\|_2^2\\
&\quad+\frac{\gamma}{2}\left\|d_2-\nabla_y^{\alpha}u-b_2^k\right\|_2^2\}
\end{aligned}
\tag{40}
$$

$$
b^{k+1} = b^k+\left(\nabla^{\alpha}u^{k+1}-d^{k+1}\right) \tag{41}
$$

The above problem Equation (40) can be separated into the following sub-problems:

$$
\begin{aligned}
u^{k+1} &= \arg\min_u\{\frac{\lambda}{2}\|u-u_0\|_2^2+\frac{\gamma}{2}\left\|d_1^k-\nabla_x^{\alpha}u-b_1^k\right\|_2^2\\
&\quad+\frac{\gamma}{2}\left\|d_2^k-\nabla_y^{\alpha}u-b_2^k\right\|_2^2\}^{\alpha}u-b_2^k\|_2^2\}
\end{aligned}
\tag{42}
$$

$$
\begin{aligned}
d^{k+1} &= \arg\min_{d_1,d_2}\{|d_1|+|d_2|+\frac{\mu}{2}\left\|d_1^k-\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2\\
&\quad+\frac{\mu}{2}\left\|d_2^k-\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)\right\|_2^2\\
&\quad+\frac{\gamma}{2}\left\|d_1^k-\nabla_x^{\alpha}u-b_1^k\right\|_2^2\\
&\quad+\frac{\gamma}{2}\left\|d_2^k-\nabla_y^{\alpha}u-b_2^k\right\|_2^2\}
\end{aligned}
\tag{43}
$$

Since the minimization of u in Equation (42) is differentiable, Equation (42) has the optimality condition:

$$
\begin{aligned}
0=&\lambda\left(u-u_0\right)-\gamma\nabla_x^{\alpha*}\left(d_1^k-\nabla_x^{\alpha*}u-b_1^k\right)\\
&-\gamma\nabla_y^{\alpha*}\left(d_2^k-\nabla_y^{\alpha*}u-b_2^k\right)
\end{aligned}
\tag{44}
$$

Equation (44) further can be derived by discretization, and be expressed by the following iterative formula:

$$
\begin{aligned}
u_{ij}^{k+1}=\frac{1}{\lambda+2\gamma}[\lambda u_{ij}^0+\gamma(d_{ij1}^k-\sum_{k=0}^{i-1}(-1)^{\alpha}C_k^{\alpha}u_{i-k,j}\\
-b_{ij1}^k+d_{ij2}^k-\sum_{k=0}^{j-1}(-1)^{\alpha}C_k^{\alpha}u_{i,j-k}-b_{ij2}^k)]
\end{aligned}
\tag{45}
$$

For d^{k+1},

$$
d_{ij}^{k+1}=shrink\left(\frac{\mu s+\gamma\left(\nabla^{\alpha}u^k+b^k\right)}{\mu+\gamma},\frac{1}{\mu+\gamma}\right), \tag{46}
$$

where $s=\nabla^{\alpha}\left(G_{\sigma}\otimes u_0\right)$,and

$$
shrink\left(x,\beta\right)=\frac{x}{|x|}*max\left(|x|-\beta,0\right)
$$

That is,

$$
d_{ij1}^{k+1}=shrink\left(\frac{\mu s_1+\gamma\left(\sum_{k=0}^{i-1}(-1)^{\alpha}C_k^{\alpha}u_{i-k,j}^{k+1}+b_{ij1}^k\right)}{\mu+\gamma},\frac{1}{\mu+\gamma}\right),
$$

where $s_1=\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right)$,

$$
d_{ij2}^{k+1}=shrink\left(\frac{\mu s_2+\gamma\left(\sum_{k=0}^{j-1}(-1)^{\alpha}C_k^{\alpha}u_{i,j-k}^{k+1}+b_{ij2}^k\right)}{\mu+\gamma},\frac{1}{\mu+\gamma}\right),
$$

where $s_2=\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)$.

The Split Bregman iteration algorithm can be presented for the AFTV-FDF model.

Algorithm 2 Algorithm of the model AFTV-FDF

1: Initialization. $b_1^0=0$, $b_2^0=0$, $d_1^0=0$, $d_2^0=0$ choose feasible λ, μ and γ, let $k=0$.
2: Iteration.
3: **while** $\frac{\left\|u^{n+1}-u^n\right\|^2}{\|u^{n+1}\|^2}\le\varepsilon$ **do**
4: $u_{ij}^{k+1}=\frac{1}{\lambda+2\gamma}[\lambda u_{ij}^0+\gamma(d_{ij1}^k-\sum_{k=0}^{i-1}(-1)^{\alpha}C_k^{\alpha}u_{i-k,j}$ $-b_{ij1}^k+d_{ij2}^k-\sum_{k=0}^{j-1}(-1)^{\alpha}C_k^{\alpha}u_{i,j-k}-b_{ij2}^k)]$
5: $d_{ij1}^{k+1}=shrink\left(\frac{\mu s_1+\gamma\left(\sum_{k=0}^{i-1}(-1)^{\alpha}C_k^{\alpha}u_{i-k,j}^{k+1}+b_{ij1}^k\right)}{\mu+\gamma},\frac{1}{\mu+\gamma}\right)$, where $s_1=\nabla_x^{\alpha}\left(G_{\sigma}\otimes u_0\right)$
6: $d_{ij2}^{k+1}=shrink\left(\frac{\mu s_2+\gamma\left(\sum_{k=0}^{j-1}(-1)^{\alpha}C_k^{\alpha}u_{i,j-k}^{k+1}+b_{ij2}^k\right)}{\mu+\gamma},\frac{1}{\mu+\gamma}\right)$, where $s_2=\nabla_y^{\alpha}\left(G_{\sigma}\otimes u_0\right)$
7: $b_{ij1}^{k+1}=b_{ij1}^k+\left(\nabla_x^{\alpha}u_{ij}^{k+1}-d_{ij1}^{k+1}\right)$
8: $b_{ij2}^{k+1}=b_{ij2}^k+\left(\nabla_y^{\alpha}u_{ij}^{k+1}-d_{ij2}^{k+1}\right)$
9: **end while**

3.2.3 Analysis of Coupled Gradient Fidelity Term Model Based on the Coordinate Descent Method

Image restoration using coordinate descent optimization method can optimally decomposeeach pixel by updating

the coordinate variables with some suitable patterns, and has an optimal convergence speed. For anisotropic forms, the coordinate descent method can be directly used to decompose each pixel.For isotropic models, the coordinate descent method cannot be used directly for decomposition, becausein this case, the horizontal and vertical gradients interact quadratic ally. The following section applies this idea to the anisotropic case of theproposed model:

$$\begin{aligned}&\min_u \int_\Omega |\nabla^\alpha u| + \frac{\lambda}{2}\|u-u_0\|_2^2 + \frac{\mu}{2}\|\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)\|_2^2 \\ =&\min_u \int_\Omega |\nabla_x^\alpha u| + |\nabla_y^\alpha u| + \frac{\lambda}{2}\|u-u_0\|_2^2 \\ &+ \frac{\mu}{2}\|\nabla^\alpha u - \nabla^\alpha (G_\sigma \otimes u_0)\|_2^2\end{aligned}$$

Then the above equation is the classical fractional order model when μ is 0 and the double fidelity term model when μ is not 0. The following study will focus on the case when μ is not 0.

Assuming that the image size is $n \times n$, the above model can be discretized as follows:

$$\begin{aligned}min_u \sum_{1\le i,j\le n} &|\nabla^\alpha u_{ij}| + \frac{\lambda}{2}\left(u_{ij}-u_{ij}^0\right)^2 \\ &+ \frac{\mu}{2}\left(\nabla^\alpha u_{ij} - \nabla^\alpha\left(G_\sigma \otimes u_{ij}^0\right)\right)^2\end{aligned}$$

Based on the idea of coordinate descent, the above denoising problem can be transformed into a series of scalar sub-problems for each pixel $u_{i,j}$

$$\begin{aligned}&\min_{u_{ij}} |\nabla^\alpha u_{ij}| + \frac{\lambda}{2}\left(u_{ij}-u_{ij}^0\right)^2 \\ &+\frac{\mu}{2}\left(\nabla^\alpha u_{ij} - \nabla^\alpha\left(G_\sigma \otimes u_{ij}^0\right)\right)^2, (i,j=1,2,\dots n)\end{aligned}$$

i.e.

$$\begin{aligned}&\min_{u_{ij}} |\nabla_x^\alpha u_{ij}| + |\nabla_y^\alpha u_{ij}| + \frac{\lambda}{2}\left(u_{ij}-u_{ij}^0\right)^2 \\ &+\frac{\mu}{2}\left(\nabla^\alpha u_{ij} - \nabla^\alpha\left(G_\sigma \otimes u_{ij}^0\right)\right)^2, (i,j=1,2,\dots n)\end{aligned}$$

Using the same splitting algorithm for the above optimization problem, the above problem can be transformed into the constrained optimization problem expressed as follows:

$$\begin{aligned}&\min_{u_{ij}} |d_{ij1}| + |d_{ij2}| + \frac{\lambda}{2}\left(u_{ij}-u_{ij}^0\right)^2 \\ &+\frac{\mu}{2}\left(d_{ij1} - \nabla_x^\alpha\left(G_\sigma \otimes u^0\right)_{ij}\right)^2 \\ &+\frac{\mu}{2}\left(d_{ij2} - \nabla_x^\alpha\left(G_\sigma \otimes u^0\right)_{ij}\right)^2\end{aligned}$$

$$s.t.\ d_{ij1} = \nabla_x^\alpha u_{ij}, d_{ij2} = \nabla_y^\alpha u_{ij}$$

The constrained problem is then transformed into an unconstrained problem:

$$\begin{aligned}&\min_{u_{ij}} |d_{ij1}| + |d_{ij2}| + \frac{\lambda}{2}\left(u_{ij}-u_{ij}^0\right)^2 \\ &+\frac{\mu}{2}\left(d_{ij1} - \nabla_x^\alpha\left(G_\sigma \otimes u^0\right)_{ij}\right)^2 \\ &+\frac{\mu}{2}\left(d_{ij2} - \nabla_x^\alpha\left(G_\sigma \otimes u^0\right)_{ij}\right)^2 \\ &+\frac{\gamma}{2}\left(d_{ij1} - \nabla_x^\alpha u_{ij}\right)^2 + \frac{\gamma}{2}\left(d_{ij2} - \nabla_y^\alpha u_{ij}\right)^2\end{aligned}$$

With the introduction of variable b, the above problem can be further transformed into the following problem:

$$\begin{aligned}&\left(u_{ij}^{k+1}, d_{ij1}^{k+1}, d_{ij2}^{k+1}\right) = \arg\min_{u_{ij},d_{ij1},d_{ij2}} |d_{ij1}| + |d_{ij2}| \\ &+\frac{\lambda}{2}\left(u_{ij}-u_{ij}^0\right)^2 + \frac{\mu}{2}\left(d_{ij1} - \nabla_x^\alpha\left(G_\sigma \otimes u^0\right)_{ij}\right)^2 \\ &+\frac{\mu}{2}\left(d_{ij2} - \nabla_x^\alpha\left(G_\sigma \otimes u^0\right)_{ij}\right)^2 + \frac{\gamma}{2}\left(d_{ij1} - \nabla_x^\alpha u_{ij} - b_{ij1}^k\right)^2 \\ &+\frac{\gamma}{2}\left(d_{ij2} - \nabla_y^\alpha u_{ij} - b_{ij2}^k\right)^2\end{aligned} \tag{47}$$

$$b_{ij1}^{k+1} = b_{ij1}^k + \left(\nabla_x^\alpha u_{ij}^{k+1} - d_{ij1}^{k+1}\right) \tag{48}$$

$$b_{ij2}^{k+1} = b_{ij2}^k + \left(\nabla_y^\alpha u_{ij}^{k+1} - d_{ij2}^{k+1}\right) \tag{49}$$

Decomposing the above problem Equation (47),

$$\begin{aligned}u_{ij}^{k+1} =& \arg\min_{u_{ij}} \frac{\lambda}{2}\left(u_{ij}-u_{ij}^0\right)^2 + \frac{\gamma}{2}\left(d_{ij1} - \nabla_x^\alpha u_{ij} - b_{ij1}^k\right)^2 \\ &+ \frac{\gamma}{2}\left(d_{ij2} - \nabla_y^\alpha u_{ij} - b_{ij2}^k\right)^2\end{aligned} \tag{50}$$

$$\begin{aligned}d_{ij1}^{k+1} =& \arg\min_{d_{ij1}} |d_{ij1}| + \frac{\mu}{2}\left(d_{ij1} - \nabla_x^\alpha\left(G_\sigma \otimes u^0\right)_{ij}\right)^2 \\ &+ \frac{\gamma}{2}\left(d_{ij1} - \nabla_x^\alpha u_{ij} - b_{ij1}^k\right)^2\end{aligned} \tag{51}$$

$$\begin{aligned}d_{ij2}^{k+1} =& \arg\min_{d_{ij2}} |d_{ij2}| + \frac{\mu}{2}\left(d_{ij2} - \nabla_x^\alpha\left(G_\sigma \otimes u^0\right)_{ij}\right)^2 \\ &+ \frac{\gamma}{2}\left(d_{ij2} - \nabla_y^\alpha u_{ij} - b_{ij2}^k\right)^2\end{aligned} \tag{52}$$

The following is to discretize Equation (50), Equation (51) and Equation (52) to obtain the appropriate iterative update of the relational equations.

$$\begin{aligned}u_{ij}^{k+1} =& \frac{1}{\lambda+2\gamma}[\lambda u_{ij}^0 + \gamma(d_{ij1}^k - \sum_{k=0}^{i-1}(-1)^\alpha C_k^\alpha u_{i-k,j} - b_{ij1}^k \\ &+ d_{ij2}^k - \sum_{k=0}^{j-1}(-1)^\alpha C_k^\alpha u_{i,j-k} - b_{ij2}^k)]\end{aligned} \tag{53}$$

$$d_{ij1}^{k+1} = shrink(\frac{\mu\nabla_x^\alpha\left(G_\sigma \otimes u_0\right) + \gamma\left(\sum_{k=0}^{i-1}(-1)^\alpha C_k^\alpha u_{i-k,j}^{k+1} + b_{ij1}^k\right)}{\mu+\gamma}, \frac{1}{\mu+\gamma})$$

$$d_{ij2}^{k+1} = shrink\left(\frac{\mu\nabla_y^{\alpha}\left(G_\sigma \otimes u_0\right) + \gamma\left(\sum_{k=0}^{j-1}(-1)^{\alpha} C_k^{\alpha} u_{i,j-k}^{k+1} + b_{ij2}^{k}\right)}{\mu+\gamma}, \frac{1}{\mu+\gamma}\right)$$

The analysis shows that, the previous use of coordinate descent method for the dual-fidelity model simplifies the problem by choosing the direction of the iteration from the beginning. This makes the model in the following analysis easier and the difficulty of discretization,ultimately reducing the computational effort in computer processing. Since the Split Bregman algorithm is used in the previous analysis for the anisotropic case of the dual-fidelity model, the discretization processdescribed-aboveis still using the splitting algorithm.Thereby, the final updated iterative equation is the same as the previous one, and the experiment will not be repeated in Section 4.

4 Numerical Experiments

In this section, numerical evidence demonstrates that the model has better texture enhancement and image denoising capabilities.

4.1 Preparation

First, some preparations for the experiment are described.

To evaluate the quality of denoised images, the values of PSNR (Peak Signal to Noise Ratio) can be calculated:

$$\begin{aligned} PSNR &= 10\log_{10}\frac{MaxValue^2}{MSE}\,(dB) \\ &= 10\log_{10}\frac{255^2}{MSE}\,(dB)\,, \end{aligned} \tag{54}$$

$$MSE = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[u\left(i,j\right)-u_0\left(i,j\right)\right]^2. \tag{55}$$

The above MSE is the mean-variance measure; $M\times N$ is the size of the image; u is the denoised image, and u_0 denotes the original image. The larger PSNR, the better the effect of image denoising.Although PSNR is an important index to measure the denoising effect, the denoised images with high PSNR do not always have better visual effects than those with low PSNR. Both PSNR and visual impressions are considered.

The iterative process of the algorithms used here is given in Section 3.1.2 and Section 3.2.2, respectively. For the iterations of these algorithms, the stopping conditions are given by the iteration stops when the following conditions are satisfied:

$$\frac{\left\|u^{n+1}-u^{n}\right\|^2}{\left\|u^{n+1}\right\|^2} \le \varepsilon, \tag{56}$$

where ε is the maximum permissible error. In general, the range of ε is $5.0*10^{-4}$-$1.0*10^{-2}$, and here $\varepsilon = 5.0*10^{-3}$. The images in Figure 1 will be used for numerical experiments.

The image sizeused in the experiment is 128×128. In addition, all the numerical calculations are accomplished by using MATLAB R2017b.

4.2 Model Parameters

The denoisingmainly depends on the parameters λ and Equation (24). Usually, the key of problem is to find a parameter that strikes a good balance between filtering enough noise and not losing too much information. Then experiments with different parameters are carried out for comparison. To find the relationship between λ and μ, the following experiments are carried out. The proposed IFTV-FDF model and the AFTV-FDF model are used to handle the image 'Dog' of 128×128, which is corrupted by additive white Gaussian noise with a standard deviation of 15 and 20, respectively. The PSNR values are recorded at different parameters. The results are shown in Figure 2 and Figure 3.

Let $\alpha = 0.9$ and $\gamma = 5$, the figures show the processing effect of the two models under different λ and μ.

From Figure 2 and Figure 3, it can be seen that the variation of PSNR with parameters is similar indifferent noises. For IFTV-FDF model, the optimal value of λ is around 0.4 and the corresponding μ has the optimal value of around 0.26. For AFTV-FDF model, the optimal value of λ is around 1.4 and the corresponding μ has the optimal value of around 1.

The denoising effect of the AFTV-FDF model by Split Bregman iteration algorithm is also related to the parameter γ in Section 3.2.2. In the following experiment, the values of λ and μ are fixed, $\lambda = 1.4$ and $\mu = 1$. The experiments are carried out bythe AFTV-FDF model, with the variation of γ at several different fractional order values ($\alpha = 0.5, 0.9, 1.5$). The results are shown in Figure 4.

In Figure 4, a small change of γ makes a small effect on the experimental results, vice versa. The results are better when $\alpha = 0.9$.

It can be seen that atdifferent noises, the models proposed in this paper have optimal denoising effect, and the effect is similar.

After a series of comparisons, $\lambda = 0.4$, $\mu = 0.26$ and $\alpha = 0.9$ in the IFTV-FDF model and $\lambda = 1.4$, $\mu = 1$, $\gamma = 3$and $\alpha = 0.9$ in the AFTV-FDF model.

4.3 Numerical Implementations

The parameters in Section4.2 are applied to Fig,1, with the noise $\sigma = 15$. Equation (6) is a FTV model. As a control model, FTV model is compared with the model proposed in this paper. The method in Section3.2 is used for FTV model. The method in Section 4.2 is used to determine the parameters. The parameters with the best effect of FTV model is $\lambda = 0.07$, $\mu = 0.0011$ and $\alpha = 1.1$.

In Figure 5, the first line is the images are corrupted by additive white Gaussian noise with a standard devi-

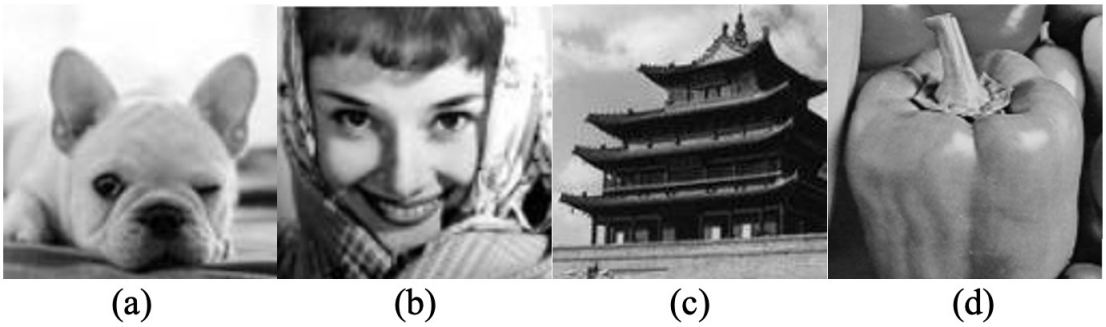


Figure 1: The original image: (a) Dog (b) Actress (c) Building (d) Pepper

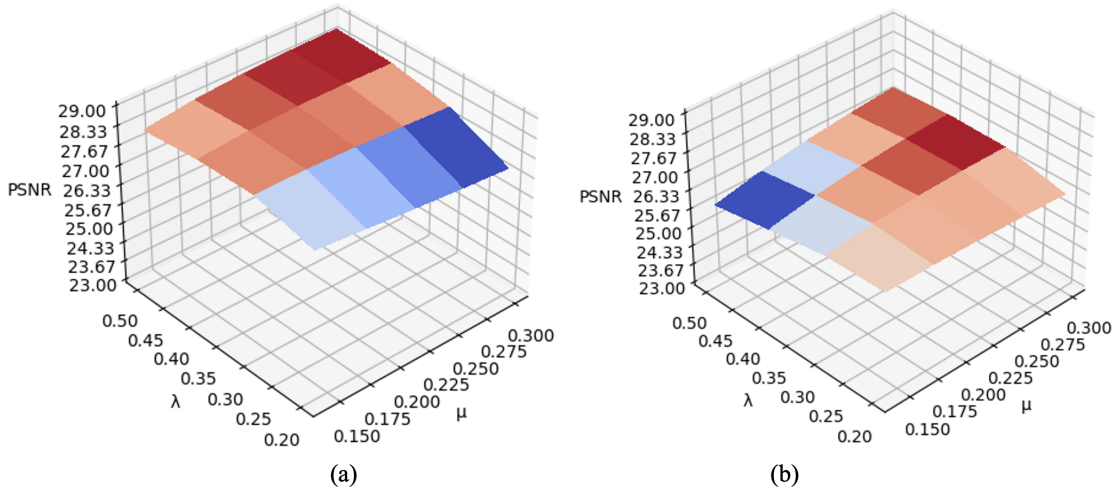


Figure 2: Comparison of parameter values in IFTV-FDF model (a)$\sigma = 15$ (b)$\sigma = 20$

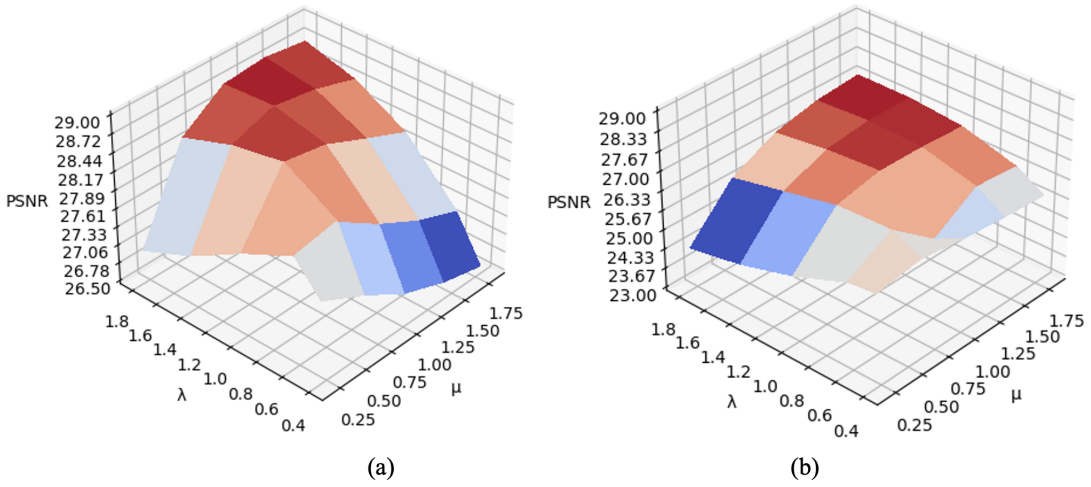


Figure 3: Comparison of parameter values in AFTV-FDF model (a)$\sigma = 15$ (b)$\sigma = 20$

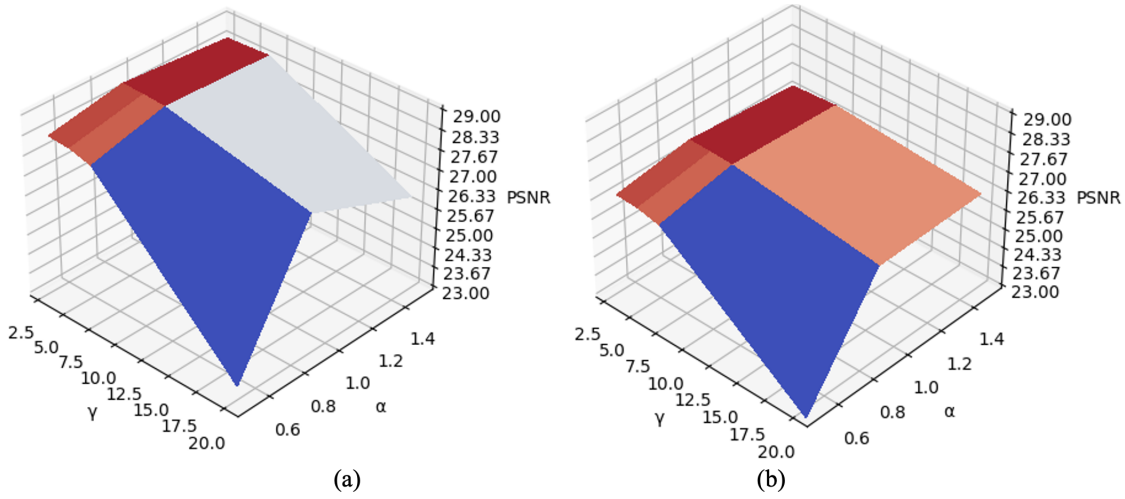


Figure 4: Comparison of γ at different α (a)$\sigma = 15$ (b)$\sigma = 20$

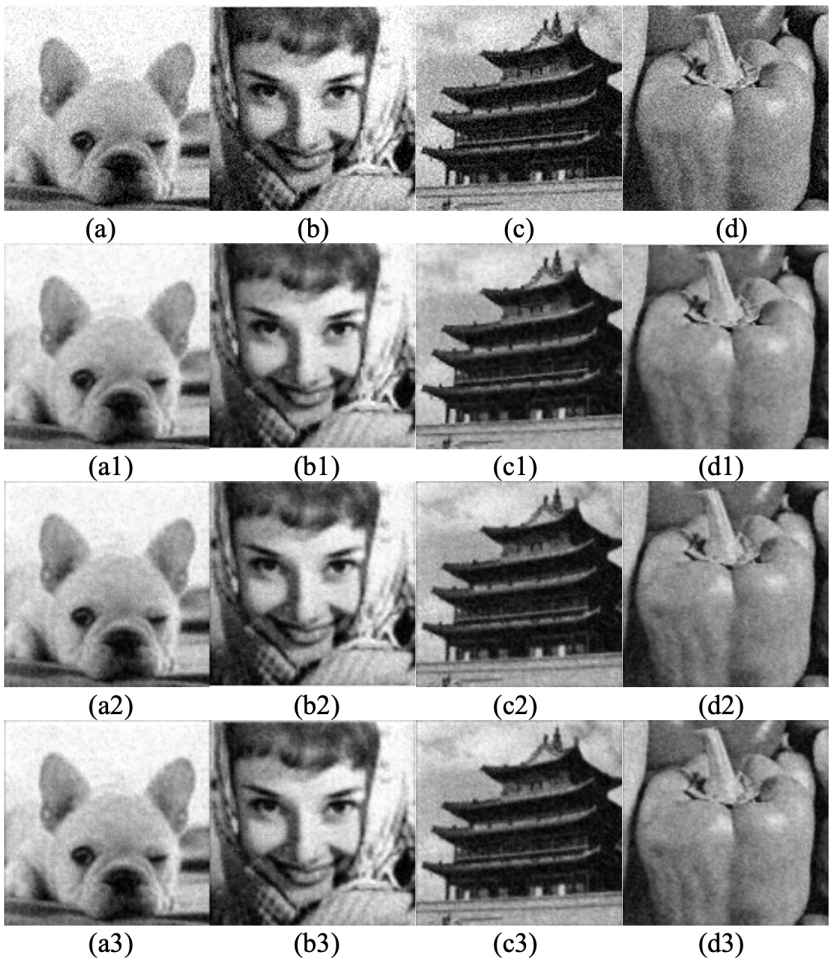


Figure 5: (a),(b),(c), and (d) are noise images. Denoising images: (a1), (b1), (c1), and (d1) are denoised by FTV model. (a2), (b2), (c2), and (d2) are denoised by IFTV-FDF model. (a3), (b3), (c3), and (d3) are denoised by AFTV-FDF model

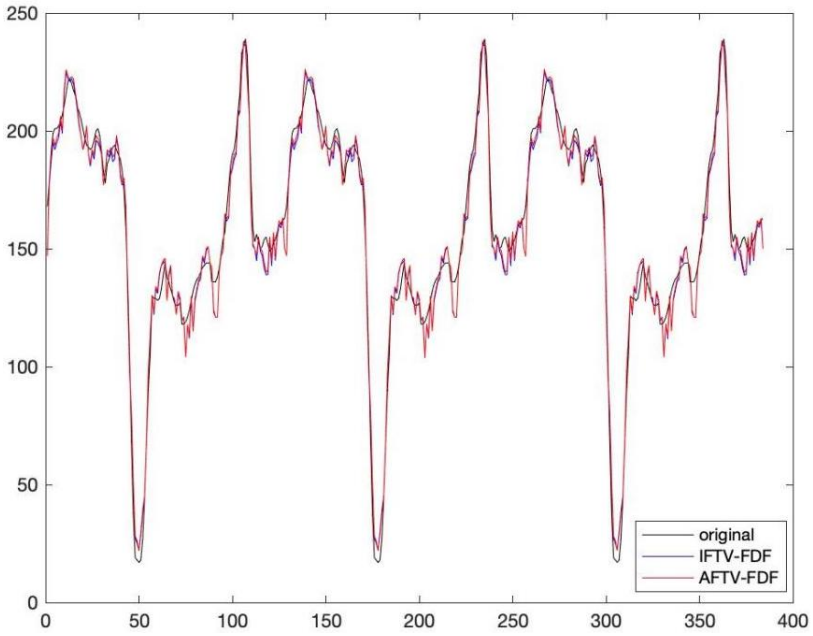


Figure 6: The 80th line of the denoising 'Dog' image

Table 1: PSNR of denoising images

model	Dog	Actress	Building	Pepper
FTV	28.4858	26.3492	26.9380	28.2756
IFTV-FDF	28.6027	26.9322	27.5090	28.6004
AFTV-FDF	28.7774	26.9748	27.5103	28.5755

ation of 15 respectively. The second line is FTV model; the third line is IFTV-FDF model, and the fourth line is AFTV-FDF model. Data of PSNR can be found in Table 1. From Figure 5, the model proposed in this paper is better than the FTV model in terms of denoising effect and clarity. From Table 1, it can be seen that the model has a larger PSNR.

The IFTV-FDF model and AFTV-FDF model are compared with the corresponding algorithms. To efficiently show the denoised abilities, the 80th line of the original image of 'Dog' and the denoising image are selected, as shown in Figure 6.

It can be seen that there is little difference in the effect of the two models. The IFTV-FDF model has slightly better denoising results with the corresponding Euler-Lagrange equations, closer to the original image.

To observe the denoising effect in more detail, a part of the 'Building' is taken to make 3D images for noise image and denoising image, as shown in Figure 7. It can be seen that the image is smooth after denoising, and the results are similar.

The model is checked to overcome the staircase effect. During the image denoising process, at some time, there is a 'staircase effect' or 'block effect'. This means that as the number of iterations increases, the image transitions to a chunked, homogeneous grayscale image. The edge of an image is the part of the image where the brightness of alocal region changes significantly. The gray profile of this region can be regarded as a step, i.e., a sharp change from one gray value in a small buffer region to another gray value with a large gray difference. The edge part of the image concentrates most of the information of the image. The determination and extraction of image edges is important for the recognition and understanding of the entire image scene.It is also an important feature on which the image segmentation relies, while edge detection is mainly the metric, detection and localization of image grayscale changes. Here Canny edge detection is applied.

In Figure 8, the first column is the original image; the second column, the third column, the fourth column, and the fifth column are the edge detection images of the original image, the noisy image, IFTV-FDF, and AFTV-FDF, respectively. Denoising images have far fewer edge lines than noisy images, and there is no reduction compared with the original image. It indicates that the image details are retained while removing the noise. In addition, the edge lines of the denoised images are smooth, indicating that this model can overcome the staircase effect. Visual observation methods show the characteristics of the model to overcome the staircase effect and preserve the edges. The models and algorithms proposed in this paper are effective.

5 Conclusions

This paper focuses on an improved fractional-order model with fidelity term coupling. It consists of a fractional-

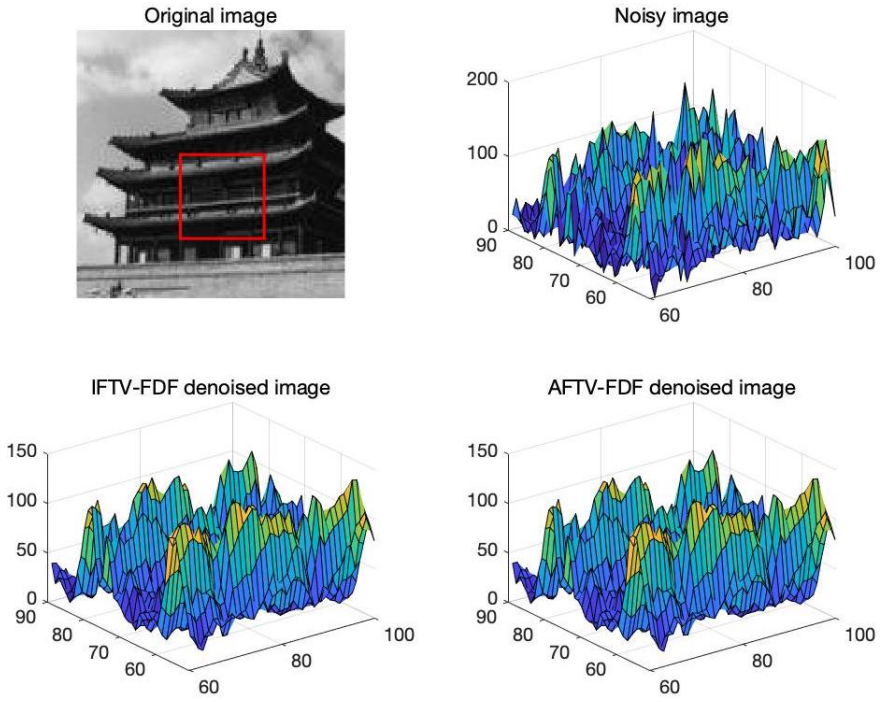


Figure 7: 3D images for noise image and denoising image

Figure 8: Canny edge detection images

order regular term, an integer-order fidelity term, and a fractional-order gradient fidelity term. According to different definitions of canonical terms, this model is analyzed. The corresponding Euler-Lagrange equation of isotropic case is introduced, and then processed by the artificial time evolution method. In terms ofthe anisotropic case, the Split Bregman Iteration algorithm is applied. In the experiments, the method of controlling the parameter variables is used for denoising the noisy images, anda more suitable parameter value is found. After that, the experiments are then compared to the Split Bregman method to denoise images with FTV models. The results show that the improved model is effective for image denoising. Finally, the results of the IFTV-FDF model and the AFTV-FDF model are compared through linear and 3D images. The final experimental results indicate that the improved fractional-order model has advantages in image denoising and has better performance in edge protection. In the future, the model will be used for network security and further research.

Acknowledgments

The authors thank reviewers for their helpful comments and valuable suggestions. This work is supported by the China Scholarship Council Fund (No. 201606465066) and the University of Science and Technology Beijing 2016 Youth Talents Project.

References

[1] L. Alvarez, P.L. Lions, and J.M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion ii," *SIAM Journal on Numerical Analysis*, vol. 29, no. 3, pp. 845–866, 1992.

[2] J. Bai and X. C. Feng, "Fractional-order anisotropic diffusion for image denoising," *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2492–2502, 2007.

[3] L.M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200–217, 1967.

[4] F. Catté, P.L. Lions, J.M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM Journal on Numerical Analysis*, vol. 29, no. 1, pp. 182–193, 1992.

[5] T. Chan, A. Marquina, and P. Mulet, "High-order total variation-based image restoration," *SIAM Journal on Scientific Computing*, vol. 22, no. 2, pp. 503–516, 2000.

[6] C. C. Chang, K. F. Hwang, and M. S. Hwang, "A block based digital watermarks for copy protection of images," in *Fifth Asia-Pacific Conference on ... and Fourth Optoelectronics and Communications Conference on Communications (IEEE 1999)*, pp. 977–980, Beijing, China, October 1999.

[7] D. Chen, S. Sun, C. Zhang, Y. Chen, and D. Xue, "Fractional-order tv-l2 model for image denoising," *Central European Journal of Physics*, vol. 11, no. 10, pp. 1414–1422, 2013.

[8] M.A. Figueiredo, J.M. Bioucas-Dias, and R.D. Nowak, "Majorization minimization algorithms for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2980–2991, 2007.

[9] T. Goldstein and S. Osher, "The split bregman method for l1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[10] Y.P. Hu, W.W. Kong, M. Li, and C.L. Huang, "Improved panoramic image mosaic algorithm for tv image denoising model," *Computer Engineering and Applications*, vol. 57, no. 17, pp. 203–209, 2021.

[11] L.C. Huang, L.Y. Tseng, and M.S. Hwang, "The study on data hiding in medical images," *International Journal of Network Security*, vol. 14, no. 6, pp. 301–309, 2012.

[12] G.Y. Liu, Z.Y. Zeng, Y. Cao, E.M. Zhao, and C.X. Xing, "Image denoising based on partial differential equation and wiener filter model," *Journal of Sichuan Vocational and Technical College*, vol. 32, no. 2, pp. 163–168, 2022.

[13] M. Lysaker, A. Lundervold, and X.C. Tai, "Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1579–1590, 2003.

[14] M. Lysaker and X.C. Tai, "Iterative image restoration combining total variation minimization and a second-order functional," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 5–18, 2006.

[15] Q. Ma, F. Dong, and D. Kong, "A fractional differential fidelity-based pde model for image denoising," *Machine Vision and Applications*, vol. 28, no. 5, pp. 635–647, 2017.

[16] I. Podlubny, *Fractional Differential Equations*. New York: Academic Press, 1999.

[17] Y.F. Pu and J.L. Zhou, "A novel approach for multiscale texture segmentation based on fractional differential," *International Journal of Computer Mathematics*, vol. 88, no. 1, pp. 58–78, 2011.

[18] Z. Ren, C. He, and Q. Zhang, "Fractional order total variation regularization for image super-resolution," *Signal Processing*, vol. 93, no. 9, pp. 2408–2421, 2013.

[19] L.I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physical D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[20] J.Z. Wang, Z.K. Pan, W.B. Wei, and J. Xu, "High-order variational model for color image multiplicative noise removal," *Computer Simulation*, vol. 37, no. 2, pp. 443–448+485, 2020.

[21] W. Wang, C. Wu, and J. Deng, "Piecewise constant signal and image denoising using a selective averaging method with multiple neighbors," *Inverse Problems & Imaging*, vol. 13, no. 5, pp. 903–930, 2019.

[22] X. Wang, W. Chen, J. Gao, and C. Wang, "Hybrid image denoising method based on non-subsampled contourlet transform and bandelet transform," *IET Image Processing*, vol. 12, no. 5, pp. 778–784, 2018.

[23] C. C. Wu, S. J. Kao, W. C. Kuo, and M. S. Hwang, "Enhance the image sharing with steganography and authentication," in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IEEE 2008)*, pp. 1177–1181, Harbin, China, August 2008.

[24] C. C. Wu, S. J. Kao, W. C. Kuo, and M. S. Hwang, "A robust-fragilewatermarking scheme for image authentication," in *Proceedings of The Third International Conference on Innovative Computing, Information and Control (ICICIC 2008)*, pp. 176–180, Dalian, China, June 2008.

[25] C. C. Wu, S. J. Kao, W. C. Kuo, and M. S. Hwang, "Reversible secret image sharing based on shamir's scheme," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IEEE 2009)*, pp. 1014–1017, Kyoto, Japan, September 2009.

[26] L. Xiao, L. Huang, and Z. Wei, "Comments on "staircase effect alleviation by coupling gradient fidelity term," *Image and Vision Computing*, vol. 28, no. 11, pp. 1569–1574, 2010.

[27] Y.H. Xu, "Image denoising algorithm based on fractional calculus," *Electronic Design Engineering*, vol. 29, no. 24, pp. 48–51+58, 2021.

[28] Y.L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal," *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1723–1730, 2000.

[29] F. Zhang, N. Cai, J. Wu, G. Cen, H. Wang, and X. Chen, "Image denoising method based on a deep convolution neural network," *IET Image Processing*, vol. 12, no. 4, pp. 485–493, 2018.

[30] X.J. Zhang, "Adaptive partial differential equation and image denoising," *Shanghai University*, 2019.

[31] L.X. Zhu and D.S. Xia, "Staircase effect alleviation by coupling gradient fidelity term," *Image and Vision Computing*, vol. 26, no. 8, pp. 1163–1170, 2008.

Biography

Donghong zhao is an associate professor of the University of Science and Technology Beijing .She obtained a doctorate of Science.Her research interests include image processing and so on. She have published more than twenty articles about image processing.

Xinyao Yu is a graduate student in the School of Mathematics and Physics, University of Science and Technology Beijing. Her research interests include image denoising.

Haoyu Liu received a Master's degree from the University of Science and Technology Beijing in 2019. Her research interests include image denoising.

Freeze-Phish: An ANN Based Phishing Detection System

Cheng-Ying Yang[1], Chun-Yi Shih[2], Chou-Chen Yang[2], and Min-Shiang Hwang[3,4]
(Corresponding author: Min-Shiang Hwang)

Department of Computer Science, University of Taipei, Taipei, Taiwan (R.O.C.)[1]
Department of Management Information Systems, National Chung Hsing University, Taichung, Taiwan (R.O.C.)[2]
Department of Computer Science and Information Engineering, Asia University[3]
Fintech and Blockchain Research Center, Asia University[4]
Taichung 41354, Taiwan
Email: mshwang@asia.edu.tw

(Received Dec. 11, 2022; Revised and Accepted July 21, 2023; First Online Aug. 31, 2023)

Abstract

Phishing (also known as online phishing) is an online criminal. Attackers use a fake webpage that imitates trusted websites to steal sensitive personal information such as passwords and credit card details. In this paper, we propose an application named Freeze-Phish, which uses Python to build a web crawler to collect information such as hyperlinks from the website. In addition, we build a brand word and suspicious word database by editing distance algorithms like Levenshtein distance and Hamming distance to compare the difference between the words in the website URL and the suspicious one. Then, we use a neural network to train our model and export our code as executable code(.exe) so that our users can use our code more easily to detect suspicious websites. Compared to other methods, our model's accuracy is about 97% true positive rate, and the average execution time is 21.3 seconds.

Keywords: Hamming Distance; Levenshtein Distance; Machine Learning Model; Phishing Detection System

1 Introduction

Over the past years, phishing attack has been a severe issue for online users. At-tackers design a similar website either by copying or slightly changing the legitimate web so that the online user can't differentiate between phishing and legitimate web. Anti-Phishing Working Group (APWG) 2019 1Q report [1] shows the percentage of different kinds of web affected by phishing attacks. The highest percentage is SaaS/webmail and payment. Moreover, it's the first time phishing of SaaS and web-mail webs surpass payment webs. APWG 2019 2Q [2] report shows that SaaS/webmail web's percentage is still higher than payment's, which means the at-tackers change their target from payment to SaaS/webmail.

In addition, the percentage of phishing attacks hosted on HTTPS has been increas-ing rapidly in these few years, which has caused a severe problem [3, 4]. More and more webs use SSL because browser only warns users when SSL is unused. This will con-fuse users when they enter a phishing website hosted on HTTPS.

The ways that attackers use to build a phishing web is to copy legitimate web ele-ments as far as possible to fool users. Some replace entire legitimate text with a single image, some replace legitimate hyperlinks with NULL or their links, and some host on HTTPS to imitate highly legitimate behavior. According to these problems, this paper aims to help users distinguish between phishing and legitimate webs, especially those hosted on HTTPS.

2 Related Work

In 2016, Moghimi and Varjani [5] used a decision tree plus a support vector ma-chine (SVM) as a training model. They used a rule-based method to make a browser extension named PhishDetector. It was rapid for users to determine whether the web-site was phishing or legitimate. However, it's only designed for Chrome users. In addition, since they wanted to make a browser extension, they needed to make a white-box model, so they used a decision tree, which usually had a problem. That problem was that decision trees can be unstable because slight variations in the data might result in a completely different tree being generated.

In 2018, Jain and Gupta [6] proposed a model using html source code and URL text as features, using the random forest. Some of those features were HTTP Status Codes, such as 404 302. However, when a website has hyperlinks up to 50 or even more, it takes a long time to collect HTTP Status Codes; sometimes, it takes minutes

to collect those features, which is not rapid for users.

Likewise, in 2019, Rao and Pais [7] also used HTTP Status Codes as features, just like Jain and Gupta [6]. Unlike the former, they classified hyperlinks by position, such as in Footer, in the HTML body. The same problem is that these feature collections take a long time. Furthermore, a hyperlink may cause redirection, and when it happens, the model will keep asking for HTTP Status Code, and the website will keep redirecting. In the end, it becomes an endless loop.

In 2019, Rao and Pais [8] proposed a model called JailPhish, which was a browser extension using Search Engine Results Pages (SERP) to detect phishing websites. One of their disadvantages is that it is only for Chrome users. Another is that it's not suitable for detecting non-English websites.

In 2019, Sahingoz, Buber, Demir, and Diri [9] proposed a Machine learning-based method to detect phishing websites. They used text-based features and Markov Chain and Levenshtein distance to determine whether the word was random. This data preprocessing differs significantly from other papers. However, they only used one kind of feature, which is a text-based feature. It means some attacks can't fake on their URL, so the model wouldn't find out.

3 The Proposed Model

The Freeze-Phish method is based on two kinds of feature sets. The entire system architecture of the proposed method is shown in Figure 1.

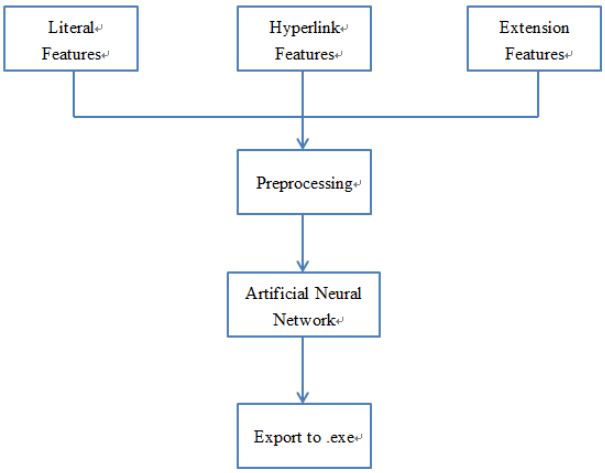


Figure 1: Entire system architecture of freeze-phish.

We propose three feature sets to improve the performance and the speed of detecting phishing attacks. The three kinds of feature sets are literal, hyperlink, and extension features. The literal feature set is about the website's URL feature, such as word number, dot number, and symbol number. It contains 26 features and will be explained in section A. The hyperlink feature set is about the website's source code, such like the hyperlink of each tag contains. The percentage of each tag has HTTPS links. It contains eight features and will be explained in section B. The extension fea-ture set is about the website's extensions, such as HTML, HTML, PHP, etc. We will collect these websites' extensions and use OneHotEncoder for preprocessing. It will be explained in section C.

3.1 Literal Features

We build a few word lists containing band and suspicious words. Band word list contains famous company names, such as Google, Apple, PayPal, and so on. A suspicious word list contains security-related words, like login, password, and register. In addition, we use brown college and Reu-ters word lists to find the count of meaningful words in the URL. Also, we use Levenshtein distance [10] and Hamming distance [11] to check whether the URL contains a similar word string.

Levenshtein distance is a string metric used to measure the difference between two sequences. The edit distance between two words is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into the other. For example, the Levenshtein distance between "soanden" and "standing" is 3.

1) soanden → standen (substitution of "t" for "o");
2) standen → standin (substitution of "i" for "e");
3) standin → standing (inserting "g" at the end).

The Hamming distance between two strings of equal length is the number of positions where the corresponding symbols differ. In other words, it measures the minimum number of substitutions required to change one string into another or possibly the minimum number of errors to convert one string into another.

1) "karolin" and "kathrin" is 3.
2) "karolin" and "kerstin" is 3.

3.2 Hyperlink Features

We build a top-level domain word list that contains some commonly used top-level domains, like .com, .edu, .gov, and .net, and we will check whether the website's URL and its hyperlink contain this domain or not.

In addition, we will call the hyperlink not hosted on the same domain as the web-site a foreign Link and check the ratio of the foreign Link among all the hyperlinks in the source code. Also, we will classify the hyperlink into four kinds by their tag, which are $< Script >, < Link >, < A >, < Img >$, and count the ratio of the hyperlink that is hosted on HTTPS (SSL).

At last, we will check whether the website has an icon. If it has an icon, we will check whether the icon's Link is

hosted in a foreign domain or not, so we will this feature into three kinds, the one with no icon, the one the icon link is hosted in a for-eign domain, and the one that icon link is hosted on the same domain.

3.3 Extension Features

The extension feature module is to obtain the file extension of the webpage and then understand the language in which the website is written. HTML is a web page used in the early days because the early DOS system was a 16-bit operating system and only accepted a file extension of up to three words, so it is simplified to him. Most HTML files are phishing websites because this web page was used early, and most legitimate websites are no longer used today. In addition, most phishing web-sites will also use PHP to write.

We will obtain the file extension of the webpage and convert it into an encoding using OneHotEncoder. We chose to use OneHotEncoder instead of LabelEncoder because most algorithms are calculated based on the metrics in the vector space. For example, There are three color features: yellow, green, and blue. You might want to make yellow = 1, green = 2, and blue = 3, that is, label different categories. However, this may allow the machine to learn "yellow <green < blue," but this is not the inten-tion of the data features we want the machine to learn.

3.4 Machine Learning Model

We use artificial neural networks to build our Freeze-Phish model. Artificial neural networks (ANNs) [12,13] are computing systems inspired by, but not identical to, the biological neural networks that make up animal brains. Artificial neural networks are based on collections of connected units, or nodes, called artificial neurons, which loosely model the neurons in biological brains. Like synapses in a biological brain, each connection can transmit a signal to other neurons. An artificial neuron that receives a signal pro-cesses it and can signal neurons connected to it.

When building an ANN, we can decide how many layers and how many neurons we want and can fit our dataset. In this paper, we build two hidden layers in our ANN model, one with 150 and one with 100 neurons. In addition, we add two dropouts and set the dropout as 0.7, so we can prevent overfitting happens. Overfitting produces an analysis that corresponds too closely or exactly to a particular data set and may fail to fit other data or reliably predict future observations. The activation we use is that the input layer uses the objective function, and two hidden layer uses the sigmoid function.

4 Experiment Results

We collect 5323 websites, 2919 legitimate websites, and 2404 phishing websites. The detail of data collection will be discussed in Section 4.1, and performance metrics will be discussed in Section 4.2.

4.1 Dataset Collection

The data we collect is from three sources. The phishing dataset is extracted from PhishTank [8] and Openphish [8], and the legitimate dataset is collected from Alexa [8] and other websites. Since the samples in the dataset can influence the performance of each model, and different researchers use different datasets and models, the accuracy is each research may be different. Therefore, we present our model accuracy as one of the performance measurements but not the primary measurement. We emphasize the execution time we test and different types of data features as our model's breakthrough.

PhishTank was launched in October 2006 by entrepreneur David Ulevitch as an offshoot of OpenDNS. The company offers a community-based phish verification system where users submit suspected phishes and other users "vote" if it is a phish.

OpenPhish launched in June 2014 due to a three-year research project on phishing detection. The research yielded a set of autonomous algorithms for detecting zero-day phishing sites. These algorithms form a self-contained kernel that can tell whether a URL is a phish.

We use Python to build a crawler to collect legitimate and phishing website data. Since the lifetime of a phishing website is very short, we must web crawling immedi-ately before the web goes offline. Many researchers using web crawlers to collect datasets must have different data since e former phishing webs are already offline, so we can't web crawl them.

4.2 Performance Metric

We calculate the actual positive rate (TPR), false positive rate (FPR), actual nega-tive rate (TNR), false negative rate (FNR), accuracy, and F1 score to evaluate the performance of Freeze-Phish. We use using 10-fold cross-validation to train and test our model. The confusion matrix and performance measures algorithm is shown in Figure 2. and the performance of Freeze-Phish is shown in Tables 1 and 2.

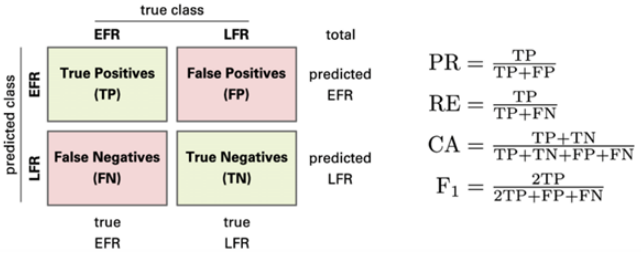


Figure 2: Confusion Matrix

Table 1: Performance of Freeze-Phish

Approach	ACC (%)	TPR (%)	FNR (%)	F1 score(%)
Freeze-Phish	97.0%	96.9%	2.8%	97.1%

Table 2: The Speed Performance of Freeze-Phish collecting one data feature Model Accu-racy Comparison

	NOT USING HTTP Status Code	USING HTTP Status Code
Accuracy	97.0%	96.9%
TPR	96.9%	96.9%
F1 Score	97.1%	96.8%
FNR	2.8%	3.0%

4.3 Black Box Model vs. White Box Model

Our model is based on the artificial neural network because it has a self-adaptability feature and a black box model, suitable for preventing zero-day attacks. Zero-day attack is the term used to describe the threat of an unknown security vulnerability in a computer application for which either the patch has not been released, or the application developers were unaware of or did not have time to fix. Besides, researchers like Mahmood Moghimi et al. [5], which use a decision tree, and Outhu Srinivasa Rao et al. [8], which use a rule and white box models, have a disadvantage. White box models are usually simple to understand the rule of classification and can be easily explained. However, these models' predicted performances are usually weaker than black box models. White box models focus on introspection, causal effects, and finding the correct model.

On the other hand, black box models focus on high computational prediction. The phishing attack is a kind of zero-day attack; attackers can change how they design the fake website over time, so the prediction model must be self-adaptive. For example, if the attackers find out the rule in the white box model, such as Outhu Srinivasa Rao et al. [8], they can change the design of the fake website and won't be detected by the machine.

4.4 Execution Time

In addition, we test our model's execution time. We show the difference between using HTTP Status Code and not HTTP Status Code to see the performance of HTTP Status Code features used in Rao et al. [7] and Jain et al. [6]. We decided not to toe HTTP Status Code because feature collection takes a long time. Sometimes a website with 100 hyperlinks may take up to 5 minutes to collect the feature. Besides this, it has another disadvantage. The disadvantage is that some of the hyperlinks will cause redirection, and when it happens, the model will keep asking for HTTP Status Code, and the website will keep redirecting. In the end, it becomes an endless loop. In addi-tion, when not using the HTTP Status Code feature, the execution time can improve, so it doesn't have to wait long for the result. The performance is shown in Table 3.

As a result, we can see that without using the HTTP Status Code feature, we can promote the execution time and feature collecting time. In addition, the accuracy of our model still stays high.

4.5 Different Types of Data Feature

In our research, we use three types of data features: literal, hyperlink, and exten-sion. Compared to Ozgur Koray Sahingoz et al. [9], we use more features than theirs. In addition, they use Levenshtein distance as their data features; our research uses not only Levenshtein distance but also Hamming distance and Levenshtein ratio as our feature collection. We add these edit distance algorithms to find more Suspicious words in the URL. In our hyperlink feature set, we cancel the HTTP Status Code feature, which can save a lot of time than Routhu Srinivasa Rao et al. [7] and Ankit Kumar Jain et al. [6] in executing and collecting data. Besides, we add features like the ratio of foreign links and firstdotlink to check whether the hyperlinks in the website are under the same domain.

5 Conclusion

In this paper, we proposed a method using machine learning to prevent phishing attacks, named Freeze-Phish, which can be used easily. Our potential users won't be limited to only Google browser users. We evaluated our application with our collected dataset and observed an accuracy of 97.0

Acknowledgments

The Ministry of Science and Technology partially supported this research, Taiwan (ROC), under contract no.: MOST 109-2221-E-468-011-MY3, MOST 108-2410-H-468-023, and MOST 111-2622-8-468-001 -TM1.

Table 3: The Speed Performance of Freeze-Phish collecting one data feature Model Execu-tion Time Comparison

	NOT USING HTTP Status Code	USING HTTP Status Code
Average collect time	22.68s	50.48s
Longest difference in time	23.9s	541s
shortest difference in time	24s	25.5s

References

[1] APWG, *Phishing Activity Trends Report 1st Quarter 2019*, May. 15 2019. (`https://docs.apwg.org/reports/apwg_trends_report_q1_2019.pdf`)

[2] APWG, *Phishing Activity Trends Report 2nd Quarter 2019*, September 12, 2019. (`https://docs.apwg.org/reports/apwg_trends_report_q2_2019.pdf`)

[3] M. J. Werner, K. Njenga, "Phishing Attack Victims and the Effect on Work Engagement," *Communications in Computer and Information Science*, vol. 1774, pp. 203–217, 2023. (`https://doi.org/10.1007/978-3-031-28472-4_13`)

[4] M. F. Alghenaim, N. A. A. Bakar, F. Abdul Rahim, V. Z. Vanduhe, G. Alkawsi, "Phishing Attack Types and Mitigation: A Survey," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 165, 2023. https://doi.org/10.1007/978-981-99-0741-0_10

[5] M. Moghimi, A. Y. Varjani, "New rule-based phishing detection method," *Expert Systems with Applications*, vol. 53, pp. 231-242, 2016.

[6] A. K. Jain, B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," *Telecommunication Systems*, vol. 68, no. 4, pp. 687–700, 2018.

[7] R. S. Rao, A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Computing and Applications*, vol. 31, no. 8, pp. 3851–3873, 2019.

[8] R. S. Rao, A. R. Pais, "Jail-Phish: An improved search engine based phishing detection system," *Computers & Security*, vol. 83, pp. 246-267, 2019.

[9] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, Banu Diri, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345-357, 2019.

[10] M. Baake, U. Grimm, R. Giegerich, "Surprises in approximating Levenshtein distances," *Journal of Theoretical Biology*, vol. 243, pp. 279-282, 2006.

[11] K. A. S. Abdel-Ghaffar, "Sets of binary sequences with small total Hamming distances," *Information Processing Letters*, vol. 142, pp. 27-29, 2019.

[12] F. Feng, X. Liu, B. Yong, R. Zhou, Q. Zhou, "Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device," *Ad Hoc Networks*, vol. 84, pp. 82-89, 2019.

[13] M. AL-Hawawreh, N. Moustafa, E. Sitnikov, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp.1-11,2018.

Biography

Cheng-Ying Yang received the M.S. degree in Electronic Engineering from Monmouth University, New Jersey, in 1991, and Ph.D. degree from the University of Toledo, Ohio, in 1999. He is a member of IEEE Satellite & Space Communication Society. Currently, he is employed as a Professor at Department of Computer Science, University of Taipei, Taiwan. His research interests are performance analysis of digital communication systems, error control coding, signal processing and computer security.

Chun-Yi Shih received his M.S. degree in Department of Management Information Systems from National Chung Hsing University, Taichung, Taiwan (ROC), in 2019. His research interests include network security, security and privacy of cloud computing, and applied cryptography.

Chou-Chen Yang received his B.S. in Industrial Education from the National Kaohsiung Normal University, in 1980, and his M.S. in Electronic Technology from the Pittsburg State University, in 1986, and his Ph.D. in Computer Science from the University of North Texas, in 1994. From 1994 to 2004, Dr. Yang was an associate professor in the Department of Computer Science and Information Engineering, Chaoyang University of Tech- nology. Currently, he is a professor in the Department of Management Information Systems, National Chung Hshing University. His research interests include network security, mobile computing, and distributed system.

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 distinguished professor and Chairman of the Department of Management Information Systems, National Chung Hsing University, during 2003-2011. He obtained 1997, 1998, 1999, 2000, and 2001 Excellent Research Awards from the 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, data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 300+ articles on the above research fields in international journals.

Trustworthy Delegation-authorization Model Based on Collaborative Access Control

Wei Sun

(Corresponding author: Wei Sun)

School of Computer and Information Technology, Xinyang Normal University
No. 237, Nanhu Road, Xinyang 464000, P. R. China
Email: sunny810715@xynu.edu.cn

(Received Feb. 28, 2023; Revised and Accepted Aug. 15, 2023; First Online Aug. 31, 2023)

Abstract

Role-based access control (RBAC) is one of the most popular models due to its various security constraint policies and flexible and convenient authorization mechanisms, such as the delegation-authorization technique. However, the security and reliability of the delegation process cannot be guaranteed using most existing approaches. Furthermore, delegation is not considered particularly in distributed and collaborative systems. To address these issues, this study proposes a novel delegation-authorization model based on the collaborative access control and trust degree and presents its framework and the delegation-authorization process. First, to ensure the consistency of the system status during the delegation process, the security and satisfiability of the system status are analyzed by implicit enforcement of the given collaborative division strategies. Second, to improve the reliability of the delegation process, the trust degrees of different candidate objects are comprehensively calculated, and the delegated objects with higher degrees are further selected. Last, the access decision to system resources is determined based on the participants' collaborative contributions. The experimental results show that compared to the existing studies, the proposed model not only satisfies the security requirements of the collaborative organization but also effectively improves the reliability of the delegation process.

Keywords: Collaborative Access Control; Collaborative Separation of Duties; Delegation Authorization; Role-based Access Control; Trust Degree

1 Introduction

The role-based access control (RBAC) model is very popular and is widely adopted by large-scale business organizations for the system deployment [6, 8, 24]. With the rapid development and wide application of the emerging information and network technologies such as the Internet of Things (IoT), online social networks, cloud computing, and blockchain, the confidentiality, privacy and integrity of the system resources become more and more important, particularly in the case of accessing sensitive data. In the computer-supported distributed and collaborative working systems based on RBAC, multiple users are required to cooperate and communicate with each other, in order to jointly make the decision to access system resources [1, 7, 12, 15, 25, 27]. This is because just trusting a single person may lead to the risk of the privileges abuse. To avoid the possibility of abusing the privileges due to too centralized privileges of individuals, multiple participants could obtain the access permission of some resource only if they together have enough contribution weights that reach or exceed the permission threshold. For this purpose, Alsulaiman *et al.* [5] proposed a threshold-based collaborative access control model, called TBCACM, which related any access privilege of the resource with a specified threshold, and then executed the role–permission assignments. Participants in the collaboration were assigned either different roles or the same role, in order to make common decision for accessing resources.

As an important manifestation of the RBAC system, the delegation-authorization technique based on RBAC chooses an appropriate delegated object as the substitute for the delegating subject or user, in order to decentralize the centralized authorization management to some ordinary object. The delegation authorization has been proved to be flexible and useful in many practical applications [3, 4, 18]. Yu *et al.* [26] proposed a locale-based access control model (LCACM) in the collaborative environment, and presented a hierarchical authorization mechanism within the collaborative group, which could meet the security and flexibility requirements of the delegation. Khan *et al.* [9] proposed a security delegation model by restricting the delegation of sensitive permission as well as the restriction from unauthorized access to resources, which could reduce the administrative burden and enable the automated delegation. Actually, the delegating subject should select competent and trustworthy delegated objects from various candidate objects with different trust degrees and execution capacities. However,

there exists the problem of arbitrariness using most of the existing approaches. The delegation process is unreliable once the delegated object with lower trustworthiness is selected. To realize the fine-grained, accurate and reliable delegation authorization, Liu *et al.* [14] combined conventional RBAC model and trust management, and proposed a trust-based access control model (TBACM), which comprehensively calculated the trust degrees of different users and evaluated their behavior ability.

According to the dynamic changes of user behavior, Zhu [28] divided the user trust into static trust level and dynamic trust level, and proposed a user trust-based dynamic multi-level access control model (UTDMACM), which could realize the hierarchical access control and fine-grained dynamic authorization. According to the dynamic variability of the network environment and user status, Liu and Chang [13] introduced the concept of trust measure and context constraint, and proposed a novel access control model based on the multi-dimension measurement and context, called MMCBACM, which could realize the dynamic and real-time control to the delegation authorization. To mitigate the malicious actions caused by the authenticated users, Abdul *et al.* [2] designed an access control mechanism by computing the trust degree based on the user's uncertain behavior, which could accurately detect and mitigate malicious users from the mobile cloud computing environment. These models or mechanisms can better reflect the reliability of the delegation process. However, they cannot meet the security requirements in a given collaborative working environment, and there may be potential security risk.

Similar to the RBAC mechanism, a critical characteristic of the collaborative access control system is that it allows the specification and enforcement of various constraint policies [10, 17, 20], including the collaborative separation of duties (CSOD) and the static mutually-exclusive-role constraint (SMER), which can ensure the security and satisfiability of the system status. Specifically, the collaborative strategy $k-n$ CSOD states that at least k users are required to cooperate with each other, in order to together execute a specific task with n different roles, which is simply expressed as $csod < \{r_1, r_2, \cdots, r_n\}, k >$. The static mutually-exclusive-role constraint $t-m$ SMER states that any user cannot have t or more roles out of the given m roles, which is simply expressed as $smer < \{r_1, r_2, \cdots, r_m\}, t >$. To meet the SOD constraints under the general model, Sarana *et al.* [21] proposed a novel role-optimization method that was represented as RMP_SOD, and developed three alternative approaches during or after the role mining. To further satisfy SOD constraints and ensure authorization security, Sun *et al.* [23] proposed a role-mining method, called role-mining optimization with separation-of-duty constraints and security detection for authorizations (RMO_SODSDA). Subsequently, in order to effectively enforce the given SOD constraints, Sun *et al.* [22] proposed another novel policy-engineering method, called policy-engineering optimization with visual representation and separation of duty constraints (PEO_VR&SOD), which utilized the method of SAT-based model counting to reduce the constraints and constructed mutually exclusive constraints. In the above-mentioned approaches for constructing RBAC systems with the SOD constraints, however, an inadequately addressed key challenge is that the delegation in the collaborative circumstance is not taken into consideration. As an alternative security-control strategy, the trust-collaboration mechanism [19] meets the confidentiality and privacy requirements of important or sensitive data, while reducing the possibility of abusing the privileges due to the randomness of the delegation authorization. Therefore, when some user is temporarily absent or is on leave, how to develop a secure and reliable delegation scheme for choosing a trustworthy substitute to participate in the collaboration is very challenging.

To resolve the above-mentioned problems, this study proposes a novel delegation-authorization model based on collaborative access control and trust degree, called TDAM_CAC. The main contributions of this work are as follows:

1) To realize the delegation authorization in the collaborative scenario, while ensuring the consistency of the system status, we present the structure of the proposal as well as the delegation-authorization process in detail, and construct the minimal set of mutually exclusive constraints to indirectly implement the collaborative strategies. We also demonstrate the effectiveness of the TDAM_CAC using real-world datasets.

2) To improve the reliability of the delegation process, we employ the trust incentive and trust penalty to comprehensively compute the trust degrees of different candidates, and then choose the objects with higher degrees to participate in the collaboration. We also demonstrate the effectiveness of the TDAM_CAC using a specific simulated system.

The rest of the article is structured as follows. The preliminaries used for our work are discussed in Section 2. Section 3 proposes a novel delegation model, and presents its delegation-authorization process and the relevant correctness verification. We implement experiments and comprehensively present the experimental analysis in Section 4, and conclude the article and discuss future work in Section 5.

2 Preliminaries

Before proposing our methodology, some necessary preliminaries are discussed, including the collaborative access control and trust computing.

2.1 Collaborative Access Control

To specify the collaborative access control based on RBAC, sets U, R, UA and RH are the basic elements of the RBAC model. Besides, we assume that any access permission of the system resources is associated with a particular weight τ, which is called permission threshold, and then present the definitions of set PT, relationship PCA and system status τ as follows.

Definition 1. *Set PT: If permission p_i associated with threshold τ_i is represented as a 2-tuple form (p_i, τ_i), then the set of all such tuples is referred to be permission set with thresholds, which is denoted as PT.*

Definition 2. *Relationship PCA: If quadruple (r_i, c_i, mc_i, p_i) is used to represent the contribution relation of role r_i to permission p_i, where c_i is the contribution weight of role r_i for permission p_i, mc_i is the maximum contributions allowed by r_i, then the set of all such quadruples is referred to be relationship of role–permission assignments with contributions, which is denoted as PCA.*

Definition 3. *System status γ: It is formalized as a triple $< UA, PCA, RH >$, denoted as γ, where UA, PCA and RH are the basic components of the RBAC model. $Roles_\gamma(u)$, $Perms_\gamma(u)$, and $w_\gamma(u)$ represent the roles, permissions with weights, and contribution weights associated to u under γ, respectively, which can be formalized as:*

$$\begin{aligned} Roles_\gamma(u) &= \{r \in R | \exists r_1 \in R, (u, r_1) \in UA \wedge (r_1, r) \\ &\quad \in RH\}; \\ Perms_\gamma(u) &= \{(p, w) | \exists (r_1, r_2) \in R, (u, r_1) \\ &\quad \in UA \wedge (r_1, r_2) \\ &\quad \in RH \wedge (r_2, w_2, mw_2, p) \in PCA\}; \\ w_\gamma(u) &= \{w | \exists (p, \tau) \in PT, (p, w) \in Perms_\gamma(u)\}. \end{aligned}$$

2.2 Trust Computing

The trust involves trust incentive and trust penalty, which are connected with the harmonization coefficient during the delegation process. For convenience in the following discussion, u_a and u_b are used to represent the delegating user, and delegated one, respectively.

Definition 4. *Harmonization coefficient for delegation $DH(u_a, u_b)$: The relationship between the successful execution of delegations and the total delegation requests is referred to be harmonization coefficient for delegation, which can be formally represented as $DH(u_a, u_b) = \frac{SD(u_a,u_b)}{DR(u_a,u_b)} \times \log_{10} SD(u_a, u_b)$, where $SD(u_a, u_b)$ is the number of successful execution of tasks by u_b instead of u_a, $DR(u_a, u_b)$ is the total number of the delegation requests that have been made by u_a, and $\log_{10} SD(u_a, u_b)$ is the stability of $SD(u_a, u_b)$ with respect to $DR(u_a, u_b)$. Obviously, when the value of $SD(u_a, u_b)/DR(u_a, u_b)$ is fixed, the greater the value of $SD(u_a, u_b)$ is, the greater the value of $DH(u_a, u_b)$ is.*

Definition 5. *Trust incentive $\alpha(u_a, u_b)$: The trust value of ua relative to u_b increases once u_b successfully completes a delegated task of u_a. The increasing value of trust is regarded as the trust incentive, which is represented as $\alpha(u_a, u_b) = tw \times DH(u_a, u_b)$, where $DH(u_a, u_b)$ is the harmonization coefficient, and parameter tw indicates the complexity of performing the delegated task.*

Definition 6. *Trust penalty $\beta(u_a, u_b)$: The trust value of ua relative to u_b correspondingly decreases once u_b does not complete the delegated task of u_a. The decreasing value is regarded as the trust penalty, which is represented as $\beta(u_a, u_b) = tw/DH(u_a, u_b)$.*

To precisely reflect the reliability of the delegation process, according to Definition 5 and Definition 6, the trust degree of ua relative to u_b, which is denoted as $T(u_a, u_b)$, should be updated in time. It can be computed as:

$$T(u_a, u_b) = \begin{cases} T(u_a, u_b) + \alpha(u_a, u_b), \\ \qquad \text{if } u_b \text{ successfully completes} \\ \qquad\qquad \text{the delegated task;} \\ T(u_a, u_b) - \beta(u_a, u_b), \text{ otherwise.} \end{cases}$$

3 Methodology

In this section, the framework of the proposed TDAM_CAC is first presented in Figure 1, which is divided into two parts: The delegation based on collaboration, and the authorization control. The structure of the former is basically consistent with the conventional delegation model, which involves the sets of delegating subjects, delegated objects, regular roles, delegated roles and permissions, as well as the assignment relationships UA and PCA. On the other hand, the latter mainly involves the trust-relation computing among the delegating and delegated users in the collaborative environment, enforcement of the CSOD strategies on UA, and contribution of the collaborative thresholds as well as the authorization decision on PCA, which are related to the basic delegation components by the dotted lines as shown in the figure.

3.1 Process of the Delegation Authorization

First, the following two definitions are presented, in order to determine the security and satisfiability of the collaborative access-control system.

Definition 7. *Security of the system status $sec(\gamma)$: Given a $k-n$ CSOD $e = CSOD < \{r_1, r_2, \cdots, r_n\}, k >$ under the system status γ, if any $(k-1)$ users cannot have all these n roles under γ, then γ is secure, denoted as $sec_e(\gamma) = 1$; otherwise, γ is not secure, denoted as $sec_e(\gamma) = 0$. Let the set of variants of $k-n$ CSOD be $E = \{e_1, e_2, \cdots\}$, if γ is secure with respect to each e_i, then γ is secure with respect to E; otherwise, γ is not secure with respect to E. Whether or not the system status*

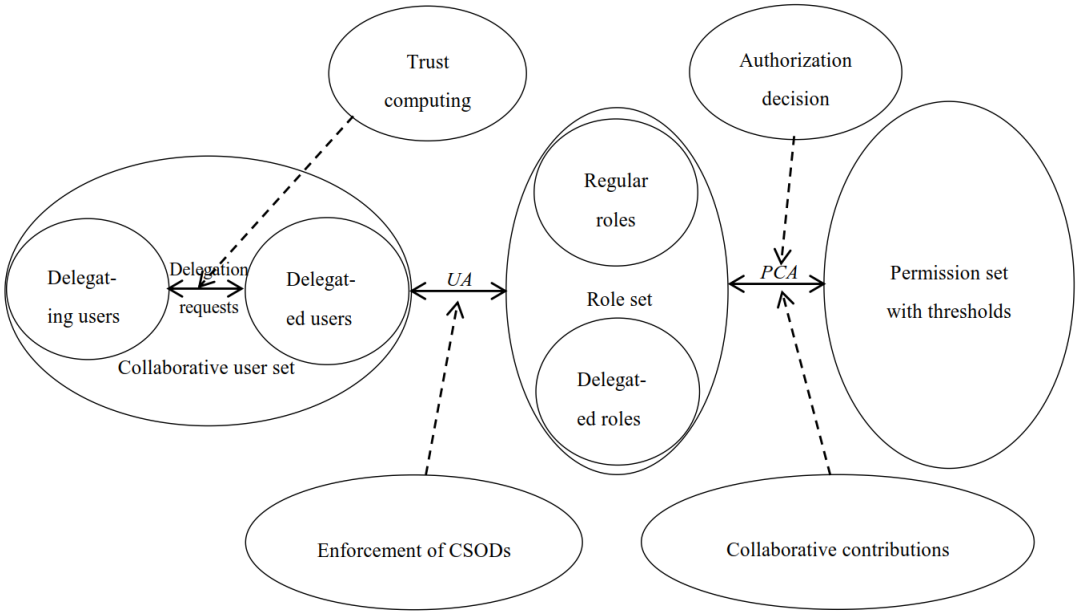


Figure 1: Framework of the TDAM_CAC

is secure can be formalized as follows:

$$\forall e \in E, \exists \{u_1, u_2, \cdots, u_{k-1}\} \subset U :$$
$$\{r_1, r_2, \cdots, r_n\} \nsubseteq \bigcup_{i=1}^{k-1} Roles_\gamma(u_i) \rightarrow sec_e(\gamma) = 1;$$
$$\exists e \in E, \exists \{u_1, u_2, \cdots, u_{k-1}\} \subset U :$$
$$\bigcup_{i=1}^{k-1} Roles_\gamma(u_i) \supseteq \{r_1, r_2, \cdots, r_n\} \rightarrow sec_e(\gamma) = 0.$$

Definition 8. *Satisfiability of the system status $sat(\gamma)$: Given a $t-m$ SMER $c = smer < \{r_1, r_2, \cdots, r_m\}, t >$ under the system status γ, if no user is allowed to have t or more roles out of all these m roles under γ, then γ is satisfied, denoted as $sat_c(\gamma) = 1$; otherwise, γ is not satisfied, denoted as $sat_c(\gamma) = 0$. Let the set of variants of $t-m$ SMER be $C = \{c_1, c_2, \cdots\}$, if γ is satisfied with respect to each c_i, then γ is satisfied with respect to C; otherwise, γ is not satisfied with respect to C. Whether or not the system status is satisfied can be formalized as follows:*

$$\forall c \in C, \exists u \in U :$$
$$|Roles_\gamma(u) \cap \{r_1, r_2, \cdots, r_m\}| < t \rightarrow sat_c(\gamma) = 1;$$
$$\exists c \in C, \exists u \in U :$$
$$|Roles_\gamma(u) \cap \{r_1, r_2, \cdots, r_m\}| \geq t \rightarrow sat_c(\gamma) = 0.$$

To together obtain the access privileges of information resources in the collaborative environment, the delegation-authorization process is presented in Figure 2. Specifically, when some user is on business trip or is on leave, he needs to delegate the corresponding permissions to another user who participates in the collaboration process as the substitute of the former. First, based on the above definitions, security and satisfiability of the system status are analyzed and determined according to the given collaborative strategy, and then suitable candidate objects that can ensure the consistency of the system status during the delegation process are selected. Next, trust degrees of different candidates are computed and compared, and the delegated objects with higher trustworthiness are chosen as the substitutes, in order to ensure the reliability of the delegation process. Last, according to the authorization condition of the access control system, whether the access to resources is permitted can be determined based on the collaborative contributions of all the participants to the access permissions.

3.2 Implicit Enforcement of the CSOD Strategy

To implicitly enforce $k-n$ CSOD, we present an approach for constructing $t-m$ SMER constraints from the $k-n$ CSOD in Algorithm 1, from which the following statements can be concluded.

Statement 1. For the given collaborative strategy $e = CSOD < \{r_1, r_2, \cdots, r_n\}, k >$ under status γ, it can be precisely enforced by the constraint formalized as $c = smer < \{r_1, r_2, \cdots, r_n\}, t >$, if and only if $t = 2$ when $k = n$ (or $t = n$ when $k = 2$).

Statement 2. For the given collaborative strategy e and a constraint set C under γ, e can be implicitly enforced by C, if and only if $\forall c \in C$: $sat_c(\gamma) \rightarrow sec_e(\gamma)$.

Theorem 1. *For the given collaborative strategy $e = CSOD < \{r_1, r_2, \cdots, r_n\}, k >$ under status γ, the SMER constraint set constructed by Algorithm 1 is minimal.*

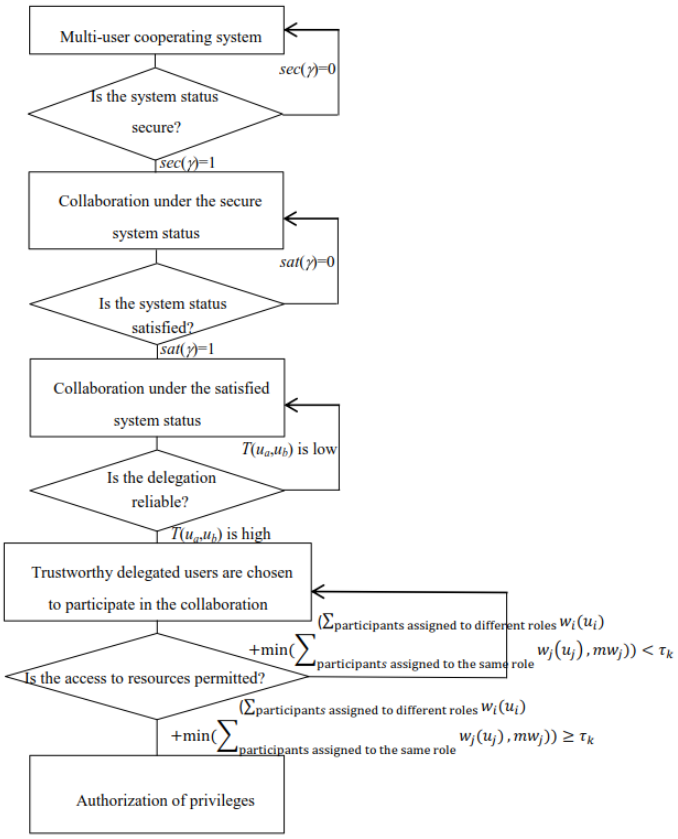


Figure 2: Flow chart of the delegation authorization

Algorithm 1 Construction of $t-m$ SMER constraints

Input: k-n CSOD constraint $CSOD=<\{r_1,r_2,...,r_n\},k>$, where $1<k\leq n$
Output: set C of t-m SMER constraints
1. Initialize $C=\emptyset$;
2. **if** $k=2$ **then**
3. $C=\{<\{r_1,r_2,...,r_n\},n>\}$;
4. **else if** $k=n$ **then**
5. $C=\{<\{r_1,r_2,...,r_n\},2>\}$;
6. **else**
7. **for** $(t=2;\left\lfloor\frac{n-1}{k-1}\right\rfloor+1;t++)$ **do**
8. $m=(k-1)\times(t-1)+1$;
9. **for** any subset $\{r_1,r_2,...,r_m'\}$ in $\{r_1,r_2,...,r_n\}$ **do**
10. $C=C\cup\{<\{r_1,r_2,...,r_m'\},t>\}$;
11. **end for**
12. **end for**
13. **end if**

Proof. According to Statement 1, $\{<\{r_1,r_2,\cdots,r_n\},2>\}$ and $\{<\{r_1,r_2,\cdots,r_n\},n>\}$ is the minimal set required. Next, we need to verify whether if holds true when $2<k<n$. The verification process considers the following two sides.

On one hand, Any $(k-1)$ users have $(k-1)\times(t-1)$ roles from $\{r_1,r_2,\cdots,r_m\}$ at most, since any user at most is allowed to have $(t-1)$ roles. Without loss of generality, let t takes the value $(\lfloor\frac{n-1}{k-1}\rfloor+1)$, the number of roles covered by any $(k-1)$ users is: $(k-1)\times(\lfloor\frac{n-1}{k-1}\rfloor+1-1)<n$, that is $sat_c(\gamma)\rightarrow sec_e(\gamma)=1$. When $t<(\lfloor\frac{n-1}{k-1}\rfloor+1)$, it also holds true. Thus, for each $c=smer<\{r_1,r_2,\cdots,r_m\},t>$ in the constructed C, c can enforce e.

On the other hand, the contradiction method is used. Assuming that c is not the minimal constraint to enforce e, and there exists another enforceable $t'-m'$ SMER c' that is less strict than c, then m' should not be greater than m, and t' should be greater than t. There are two cases:

1) When $m'=m$ and $t'>t$, the assumption is true. Then, it is concluded that $t'>\lfloor\frac{n-1}{k-1}\rfloor+1$. Without loss of generality, let t takes the value $(\lfloor\frac{n-1}{k-1}\rfloor+2)$. For c', there exists $(k-1)$ users, and the number of roles covered by these users is: $(k-1)\times(\frac{n-1}{k-1}+2-1)=n-1+k-1>n$, then $sec_e(\gamma)=0$, which breaches e. Thus, the assumption is false.

2) When $m'<m$ and $t'=t$, the assumption is true. If $sat_{c'}(\gamma)=1$, then $sat_c(\gamma)=1$, which indicates that c' that is not weaker than c. Thus, the assumption is false, and the theorem is proved.

□

4 Experimental Analysis

To evaluate the performance of the TDAM_CAC, in this section we implement experiments using the simulated system and real-world datasets, in order to demonstrate the security and reliability of the proposal. All the experiments are compiled and run under the Java environment.

4.1 Performance Evaluations Using the Simulated System

4.1.1 Problem Statements

In the following simulated system, Figure 3 presents the role-hierarchy relationship (RH) of the RBAC system when a specified organization is purchasing a batch of products. Table 1 and Table 2 represent the relationship of the original user–role assignments (UA), and that of the role–permission assignments with contributions (PCA) in the multi-department collaborative organization, respectively. When department manager a is absent, it is observed that assistant manager b, c, d,e, f and g collaborate with each other, and they can make common decision for

the purchase. This is because $0.2\tau_1+0.2\tau_1+0.3\tau_1+0.1\tau_1 \times 3 = \tau_1$, which reaches the permission threshold. Consider the following two delegation cases:

1) d is on business trip, and he temporarily delegates permission $(p_1, 0.3\tau_1)$ to b or c. Assume that at least 6 users are required to participate in the collaboration before the delegation, in order to ensure the system security. However, the collaboration only consists of 5 participants after the delegation using the conventional models, which cannot meet the security requirement.

2) b is on leave, and he temporarily delegates permission $(p_1, 0.2\tau_1)$ to one user out of h, i, and j, while ensuring 6 participants in the collaboration. b may have different degrees of satisfaction to h, i, and j who have participated in the previous collaborations. If the satisfaction degree just represents the trust degree, then the greater the degree value, the more reliable the delegation is. However, the delegation is not reliable once the delegated object with lower trustworthiness is chosen using the conventional models.

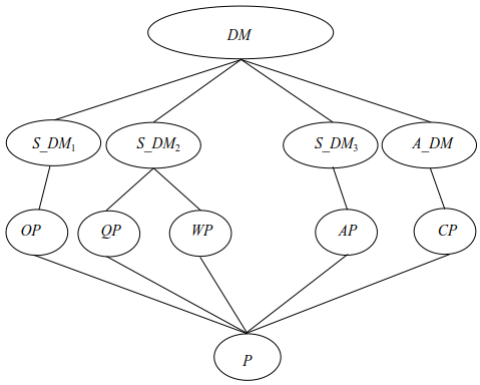


Figure 3: RH

Table 1: UA

User	Role	Annotation for role
a	DM	Department manager
b	S_DM_1	Assistant manager
c	S_DM_2	Assistant manager
d	S_DM_3	Assistant manager
e, f, g, h, i, j	A_DM	Assistant manager
k	OP	Ordering person
l	QP	Inspector
m	WP	Store keeper
n	AP	Accountant
o	CP	Cashier
q	P	Common staff

4.1.2 Security Analysis of the TDAM_CAC

d is on business trip, he needs to delegate permission $(p_1, 0.3\tau_1)$ to another participant in the collaboration. For the given collaborative strategy $e = csod < \{S\_DM_1, S\_DM_2, S\_DM_3\}, 3 >$, and constraint $c = smer < \{S\_DM_3, A\_DM\}, 2 >$, the system status γ is secure before the delegation. This is because γ is satisfied to c and is also secure to e. the delegation process is analyzed as follows:

First, if b or c is selected as the delegated object, then it is observed that $Roles_\gamma(b) \cup Roles_\gamma(c) \supseteq \{S\_DM_1, S\_DM_2, S\_DM_3\}$, which violates e, and the system security cannot be guaranteed.

Second, if e is regarded as the delegated object, it is observed that $\{S\_DM_1, S\_DM_2, S\_DM_3\} \nsubseteq Roles_\gamma(b) \cup Roles_\gamma(c)$, which meets $csod < \{S\_DM_1, S\_DM_2, S\_DM_3\}, 3 >$. However, $Roles_\gamma(e) \cap \{S\_DM_3, A\_DM\} \geq 2$, which violates $smer < \{S\_DM_3, A\_DM\}, 2 >$. Thus, the system security cannot be guaranteed yet.

Further, if n is chosen as the candidate, it is observed that $csod < \{S\_DM_1, S\_DM_2, S\_DM_3\}, 3 >$ and $csod < \{S\_DM_1, S\_DM_2, S\_DM_3\}, 3 >$ can be met simultaneously, then the system security can be guaranteed after the delegation.

Last, according to the determination condition of the collaborative access control, it can be concluded that b, c, n, e, f, g can together make the decision for the purchase. This is because $w_\gamma(b) + w_\gamma(c) + w_\gamma(n) + \min((w_\gamma(e) + w_\gamma(f) + w_\gamma(g)), 0.3\tau_1) \geq \tau_1$.

4.1.3 Reliability Analysis of the TDAM_CAC

b is on leave, he needs to delegate permission $(p_1, 0.2\tau_1)$ to another participant in the collaboration. To demonstrate the reliability of the proposal, On the basis of ensuring the consistency of the system security during the delegation process, the harmonization coefficient and trust degree are considered as the evaluation measures, respectively.

First, we take the number of successful delegations as inputs, repeatedly implement the experiments 20 times in the simulated system, and output the median values of the experimental results as shown in Figure 4, which demonstrates that the harmonization coefficient varies as the number of successful delegations varies, where the ratios of successful completion of the delegated tasks implemented by h, i, and j to the delegation requests made by b are 100%, 80%, and 50%, respectively. Specifically, we consider three cases: $SD(u_b, u_h)/DR(u_b, u_h) = 1.0$, $SD(u_b, u_i)/DR(u_b, u_i) = 0.8$, and $SD(u_b, u_j)/DR(u_b, u_j) = 0.5$. It is observed from the figure that the harmonization coefficient grows linearly as the number of successful delegations increases for all the three cases. However, it does not vary obviously once the successful number exceeds 1000. Specifically,

Table 2: PCA

Role	Maximum contribution by role	Permission with weight	Annotation for permission
DM	τ_1	(p_1, τ_1)	It is responsible for general management
S_DM_1	$0.2\tau_1$	$(p_1, 0.2\tau_1)$	It is responsible for ordering products
S_DM_2	$0.2\tau_1$	$(p_1, 0.2\tau_1)$	It is responsible for ordering products
S_DM_3	$0.3\tau_1$	$(p_1, 0.3\tau_1)$	It is responsible for ordering products
A_DM	$0.3\tau_1$	$(p_1, 0.1\tau_1)$	It is responsible for ordering products
OP	2	(p_2, τ_2)	Order products
QP	3	(p_3, τ_3)	Inspect the product quality
WP	4	(p_4, τ_4)	Store products
AP	5	(p_5, τ_5)	Keep accounts
CP	6	(p_6, τ_6)	Revenue and expenditure cash
P	7	(p_7, τ_7)	Collect and organize documents

when the number reaches 1000, the successful ratios of delegations for these cases are 3, 2.5, and 1.45, respectively, which are very close. Thus, it is difficult to choose an appropriate object from h, i, and j.

Next, based on the above experimental setting, Figure 5 presents the trust degree of the delegated objects such as $T(b, h)$, $t(b, i)$, $t(b, j)$, where the complexity of the delegated task is set as 1.0, and the initial trust degree is set as 100. It is observed from the figure that, the value of $T(b, h)$ increases significantly with the increasing number of successful delegations, while the value of $T(b, i)$ grows slightly, which respectively reach 600, and 385 with 250 successful delegations. However, the value of $T(b, j)$ tends to decrease slightly as the number of successful delegations increases, which only reaches 80. Thus, h is the most trustworthy delegated object among the three candidates.

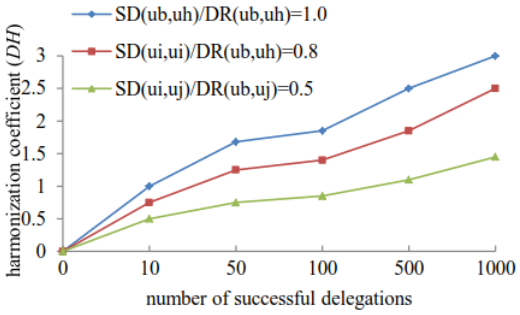


Figure 4: Evaluations of harmonization coefficient for delegation

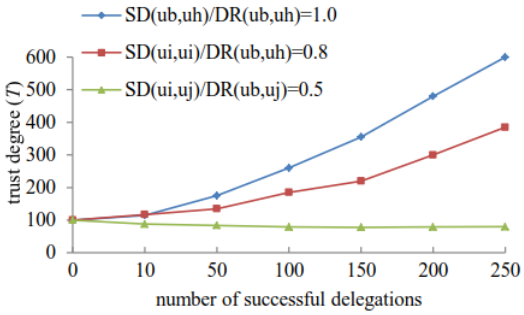


Figure 5: Evaluations of trust degree

4.2 Performance Evaluations Using the Real-world Datasets

Datasets: The real-world datasets from the work [16] are taken into consideration, in order to further evaluate the effectiveness of the implicit enforcement of the CSOD strategy. However, only five datasets can be used in the experiments, including the Americas-large, Americas-small, Apj, Customer, and Emea. Other four sets could not reflect the performance of the proposal, since the strategy cannot be enforced using the Domino, Firewall1, Firewall2, or Healthcare dataset.

Experimental setting: Four different types of the $k-n$ CSOD strategy, including 2-2 CSOD, 2-3 CSOD, 3-5 CSOD, and 5-10 CSOD, are synthetically generated by a simulator, where n permissions are randomly chosen from the permission set. The scales of $k-n$ CSODs take 30, and 50, respectively. Moreover, the initial roles with no constraints and UA are constructed by the mining tool rminer [11].

Evaluation measures: The total number of the constructed SMER constraints and the proportion of the collaborative strategies that can be precisely enforced by SMER constraints are regarded as the main evaluation measures.

We consider the initial mining results and the $k-n$ CSOD constraints as inputs, repeatedly conduct the ex-

periments 20 times, and output the average values of the experimental result and the results of the compared methods RMP_SOD and RMO_SODSDA, as shown in Tables 3 $\sim$ 10, where $E_{k-nCSOD}$ and $C_{t-mSMER}$ represent the CSOD strategy set, and SMER constraint set, respectively.

From Tables 3 $\sim$ 6 for $|E_{k-nCSOD}| = 30$, it is intuitively observed that there is a certain number of CSOD strategies that could not be enforced. It is also observed that the number of the constructed $t-t$ SMERs is greater than that of the $t-m$ SMERs, this is because the $t-t$ SMER is more restricted than the $t-m$ SMER. Take the Customer dataset as an example as shown in the tables, the number of the $t-t$ SMERs is 17, 20, 373, and 2193, respectively. However, the number of the $t-m$ SMERs is 15, 17, 369, and 2189, respectively. A further intuitive observation from the results of the proposal is that, the number of the constructed SMERs increases significantly with the increasing length of the CSOD. Similar to the above analysis for $|E_{k-nCSOD}| = 30$, the detailed analysis of the experimental results for $|E_{k-nCSOD}| = 50$, which is presented in Tables 7 $\sim$ 10, is omitted owing to the limited space.

Finally, and also most importantly, the results using different methods are comparable. Specifically, taking the Americas-small dataset as an example, when $|E_{2\text{–}3CSOD}| = 30$, the number of the $t-m$ SMERs is 3307, 3304, and 3305, respectively; when $|E_{5\text{–}10CSOD}| = 30$, the number of the $t-m$ SMERs is 24908, 24901, and 24954, respectively. Thus, TDAM_CAC performs as well as the RMP_SOD and RMO_SODSDA, in order to implicitly enforce the CSOD strategy.

4.3 Benefits of the TDAM_CAC

From the above performance evaluations for the delegation-authorization process, we find the TDAM_CAC has the following main benefits.

It inherits the flexibility, convenience, and strategy-irrelevant characteristic of the RBAC model, the structure of which is basically consistent with the conventional delegation model based on RBAC. Only minor modifications to the existing model can effectively meet various delegation requirements, such as the delegating granularity and hierarchy.

The security of the system status is vulnerable to be destroyed using the existing delegation models, since the collaborative division strategies will be violated easily. To address this issue, the method of constructing the minimal set of mutually exclusive constraints is introduced using our method, in order to indirectly implement the collaborative strategy and ensure the system security.

The delegation processes are unreliable using most of the existing delegation models, since there is the danger of the privileges abuse once the delegated objects with lower trustworthiness are selected. To address this issue, the trust degrees of different candidate objects are computed and compared according to the delegation-authorization process of the proposed model, which can improve the reliability of the delegation process.

Compared with the existing delegation models, the characteristics of the proposed method are shown in Table 11, where a tick $\surd$ indicates that the characteristic is available.

5 Conclusions

A novel delegation-authorization model based on collaborative access control, called TDAM_CAC, was proposed in this study. According to its delegation-authorization process, we implemented the collaborative division strategies by construction of mutually exclusive constraints, in order to choose the candidate delegated objects in the specific collaborative scenario. Next, we utilized the trust incentive and trust penalty to comprehensively compute the trust degrees of different candidate objects, and then chose the objects with higher trust values to participate in the collaboration. We also presented the algorithm of constructing the minimal set of role constraints, and verified its correctness. The experiments using a specific simulated system and the real-world datasets demonstrated that, the proposed method could ensure the consistency of the system status and improve the reliability of the delegation process. Our future work will focus on studying how to implement the TDAM_CAC in practical scenarios such as the IoT, blockchain, and online social networks.

References

[1] D. Abdelfattah, H. A. Hassan, and F. A. Omara, "Enhancing highly-collaborative access control system using a new role-mapping algorithm," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, p. 2765, 2022.

[2] A. M. Abdul, A. A. K. Mohammad, P. Venkat Reddy, P. Nuthakki, R. Kancharla, R. Joshi, and N. Kannaiya Raja, "Enhancing security of mobile cloud computing by trust-and role-based access control," *Scientific Programming*, vol. 2022, pp. 1–10, 2022.

[3] M. U. Aftab, Z. Qin, N. W. Hundera, O. Ariyo, N. T. Son, and T. V Dinh, "Permission-based separation of duty in dynamic role-based access control model," *Symmetry*, vol. 11, no. 5, p. 669, 2019.

[4] G. Ali, N. Ahmad, Y. Cao, M. Asif, H. Cruickshank, and Q. E. Ali, "Blockchain based permission delegation and access control in Internet of Things (BACI)," *Computers & Security*, vol. 86, pp. 318–334, 2019.

[5] F. A. Alsulaiman, A. Miege, and A. E. Saddik, "Threshold-based collaborative access control(T-CAC)," in *Proceedings of 2007 International Symposium on Collaborative Technologies and Systems*, pp. 46–56, IEEE, 2007.

[6] S. Ameer, J. Benson, and R. Sandhu, "Hybrid approaches (ABAC and RBAC) toward secure access

Table 3: Performance comparison for $|E_{2\text{–}2CSOD}| = 30$

Dataset	RMP_SoD		RMO_SODSDA		TDAM_CAC	
	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER
Americas-large	984 (95%)	984 (95%)	984 (96%)	982 (96%)	984 (96%)	983 (96%)
Americas-small	441 (80%)	441 (80%)	439 (80%)	436 (80%)	439 (80%)	438 (80%)
Apj	40 (80%)	39 (80%)	39 (80%)	37 (80%)	39 (80%)	39 (80%)
Customer	19 (95%)	18 (84%)	15 (80%)	11 (80%)	17 (81%)	15 (82%)
Emea	57 (80%)	56 (80%)	54 (80%)	51 (80%)	54 (80%)	53 (80%)

Table 4: Performance comparison for $|E_{2\text{–}3CSOD}| = 30$

Dataset	RMP_SoD		RMO_SODSDA		TDAM_CAC	
	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER
Americas-large	3307 (100%)	3307 (100%)	3306 (100%)	3304 (100%)	3306 (100%)	3305 (100%)
Americas-small	995 (80%)	994 (80%)	993 (80%)	991 (80%)	994 (80%)	990 (80%)
Apj	146 (100%)	143 (100%)	128 (96%)	119 (98%)	129 (95%)	123 (97%)
Customer	23 (60%)	20 (100%)	19 (100%)	17 (100%)	20 (100%)	17 (100%)
Emea	172 (95%)	165 (95%)	163 (93%)	155 (93%)	165 (94%)	165 (96%)

Table 5: Performance comparison for $|E_{3\text{–}5CSOD}| = 30$

Dataset	RMP_SoD		RMO_SODSDA		TDAM_CAC	
	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER
Americas-large	11335 (91%)	11335 (91%)	11335 (91%)	11335 (91%)	11335 (91%)	11335 (91%)
Americas-small	2120 (50%)	2120 (50%)	2120 (50%)	2120 (50%)	2120 (50%)	2120 (50%)
Apj	553 (95%)	564 (96%)	574 (97%)	571 (97%)	585 (98%)	577 (97%)
Customer	46 (17%)	46 (17%)	372 (100%)	369 (100%)	373 (100%)	369 (100%)
Emea	696 (100%)	698 (100%)	698 (100%)	693 (100%)	699 (100%)	697 (100%)

Table 6: Performance comparison for $|E_{5\text{–}10CSOD}| = 30$

Dataset	RMP_SoD		RMO_SODSDA		TDAM_CAC	
	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER
Americas-large	24908 (70%)	24908 (70%)	24908 (85%)	24901 (85%)	24988 (85%)	24954 (85%)
Americas-small	1307 (15%)	1307(15%)	1307(15%)	1307(15%)	1307(15%)	1307(15%)
Apj	1459 (85%)	1459 (85%)	1614 (90%)	1603 (88%)	1614 (90%)	1652 (89%)
Customer	2180 (100%)	2177 (100%)	2180 (100%)	2177 (100%)	2193 (100%)	2189 (100%)
Emea	1591 (100%)	1594 (100%)	1594 (100%)	1594 (100%)	1594 (100%)	1594 (100%)

Table 7: Performance comparison for $|E_{2\text{–}2CSOD}| = 50$

Dataset	RMP_SoD		RMO_SODSDA		TDAM_CAC	
	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER
Americas-large	2461 (92%)	2577 (92%)	2568 (92%)	2574 (92%)	2568 (92%)	2574 (92%)
Americas-small	1279 (84%)	1284 (85%)	1272 (84%)	1285 (85%)	1272 (84%)	1285 (85%)
Apj	124 (84%)	124 (84%)	124 (84%)	124 (84%)	124 (84%)	124 (84%)
Customer	48 (96%)	46 (95%)	36 (78%)	33 (76%)	36 (78%)	36 (78%)
Emea	142 (84%)	137 (83%)	138 (83%)	135 (82%)	138 (83%)	138 (83%)

Table 8: Performance comparison for $|E_{2\text{–}3CSOD}| = 50$

Dataset	RMP_SoD		RMO_SODSDA		TDAM_CAC	
	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER
Americas-large	7697 (98%)	7685 (98%)	7694 (98%)	7692 (98%)	7694 (98%)	7693 (98%)
Americas-small	3365 (85%)	3363 (85%)	3360 (84%)	3358 (84%)	3360 (84%)	3358 (84%)
Apj	345 (100%)	343 (100%)	297 (85%)	277 (83%)	297 (85%)	281 (84%)
Customer	81 (58%)	53 (43%)	49 (98%)	39 (95%)	49 (98%)	43 (96%)
Emea	425 (98%)	417 (98%)	404 (98%)	387 (95%)	404 (98%)	406 (96%)

Table 9: Performance comparison for $|E_{3\text{–}5CSOD}| = 50$

Dataset	RMP_SoD		RMO_SODSDA		TDAM_CAC	
	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER
Americas-large	31155 (88%)	31155 (88%)	31155 (88%)	31147 (88%)	31155 (88%)	31147 (88%)
Americas-small	5081 (48%)	5081 (48%)	5081 (48%)	5078 (48%)	5081 (48%)	5081 (48%)
Apj	1498 (93%)	1499 (93%)	1556 (95%)	1534 (95%)	1558 (95%)	1558 (95%)
Customer	931 (100%)	929 (100%)	930 (100%)	930 (100%)	931 (100%)	930 (100%)
Emea	1829 (100%)	1827 (100%)	1831 (100%)	1829 (100%)	1830 (100%)	1828 (100%)

Table 10: Performance comparison for $|E_{5\text{–}10CSOD}| = 50$

Dataset	RMP_SoD		RMO_SODSDA		TDAM_CAC	
	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER	$C_{t\text{-}t}$ SMER	$C_{t\text{-}m}$ SMER
Americas-large	62058 (71%)	62058 (71%)	62058 (71%)	62058 (71%)	62058 (71%)	62058 (71%)
Americas-small	3831 (16%)	3831 (16%)	3831 (16%)	3831 (16%)	3831 (16%)	3831 (16%)
Apj	3926 (90%)	3929 (90%)	4258 (90%)	4214 (88%)	4258 (90%)	4214 (88%)
Customer	5461 (98%)	5455 (98%)	5461 (98%)	5455 (98%)	5473 (98%)	5470 (98%)
Emea	4168 (100%)	4168 (100%)	4178 (100%)	4172 (100%)	4178 (100%)	4172 (100%)

Table 11: Comparisons of characteristics

Characteristic	Pal *et al.* [18]	Ali *et al.* [4]	Aftab *et al.* [3]	Khan *et al.* [9]	Abdul *et al.* [2]	Our Method
Delegating granularity	√	√				√
Delegating hierarchy						√
Security analysis	√	√	√	√	√	√
Reliability analysis				√	√	√

control in smart home IoT," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–18, 2022.

[7] S. Craß, A. Lackner, N. Begic, S. A. M. Mirhosseini, and N. Kirchmayr, "Collaborative administration of role-based access control in smart contracts," in *Proceedings of the 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS'22)*, pp. 87–94, IEEE press, 2022.

[8] M. Ghafoorian, D. Abbasinezhad-Mood, and H. Shakeri, "A thorough trust and reputation based RBAC model for secure data storage in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 4, pp. 778-788, 2019.

[9] K. H. Khan, I. U. Din, A. Almogren, H. A. Khattak, M. Ibrahim, and S. Nazir, "Secure delegation using enhanced capability model," *Security and Communication Networks*, vol. 2022, pp. 1–9, 2022.

[10] B. Lang, J. Wang, and Y. Liu, "Achieving flexible and self-contained data protection in cloud computing," *IEEE Access*, vol. 2017, no. 5, 1510–1523, 2017.

[11] R. Li, H. Li, W. Wei, X. Ma, and X. Gu, "RMiner: a tool set for role mining," in *Proceedings of the 18th ACM Symposium on Access Control Models and Technologies*, pp. 193–196, 2013.

[12] Q. Li, X. Zhang, M. Xu, and J. Wu, "Towards secure dynamic collaborations with group-based RBAC model," *computers & security*, vol. 28, no. 5, pp. 260–275, 2009.

[13] F. Liu and C. Chang, "Access control model based on multidimensional measurement and context," *Computer Engineering*, vol. 37, no. 24, pp. 129–131, 135, 2011.

[14] W. Liu, H. Duan, H. Zhang, P. Ren, and J. Wu, "TRBAC: trust-based access control model," *Journal of Computer Research and Development*, vol. 48, no. 8, pp. 1414–1420, 2011.

[15] M. A. Madani, M. Erradi, and Y. Benkaouz, "A collaborative task role based access control model," *Journal of Information Assurance & Security*, vol. 11, no. 6, pp. 348–358, 2016.

[16] B. Mitra, S. Sural, J. Vaidya, and V. Atluri, "A Survey of Role Mining," *ACM Computing Surveys*, vol. 48, no. 4, pp. 1–37, 2016.

[17] F. Nazerian, H. Motameni, H. Nematzadeh, "Emergency role-based access control (E-RBAC) and analysis of model speci-fications with alloy," *Journal of Information Security and Applications*, vol. 45, pp. 131–142, 2019.

[18] S. Pal, T. Rabehaja, M. Hitchens, V. Varadharajan, and A. Hill, "On the design of a flexible delegation model for the Internet of Things using blockchain," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3521–3530,2020.

[19] N. C. Rathore, and S. Tripathy, "A trust-based collaborative access control model with policy aggrega-

tion for online social networks," *Social Network Analysis and Mining*, vol. 7, pp. 1–13, 2017.

[20] A. Roy, S. Sural, A. K. Majumdar, J. Vaidya, and V. Atluri, "Enabling workforce optimization in constrained attribute-based access control systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 4, pp. 1901–1913, 2019.

[21] P. Sarana, A. Roy, S. Sural, J. Vaidya, and V. Atluri, "Role mining in the presence of separation of duty constraints," in *Proceedings of the 11th International Conference (ICISS'15)*, pp. 98-117, 2015.

[22] W. Sun, H. Su, and H. Xie, "Policy-engineering optimization with visual representation and separation-of-duty constraints in attribute-based access control," *Future Internet*, vol. 12, no. 10, p. 164, 2020.

[23] W. Sun, S. Wei, H. Guo, and H. Liu, "Role-mining optimization with separation-of-duty constraints and security detections for authorizations," *Future Internet*, vol. 11, no. 9, p. 201, 2019.

[24] W. Sun, X. Yuan, and H. Su, "Role-engineering optimization with user-oriented cardinality constraints in role-based access control," *International Journal of Network Security*, vol. 23, no. 5, pp. 845–855, 2021.

[25] C. Uikey and D. S. Bhilare, "RBACA: Role-based access control architecture for multi-domain cloud environment," *Inter-national Journal of Business Information Systems*, vol. 28, no. 1, pp. 1–17, 2018.

[26] G. Yu, R. Li, Z. Lu, W. Song, and Z. Tang, "Locale-based access control model in collaborative environment," *Computer Science*, vol. 36, no. 1, pp. 81–85, 2009.

[27] J. Zhang, T. Li, Z. Ying, and J. Ma, "Trust-based secure multi-cloud collaboration framework in cloud-fog-assisted IoT," *IEEE Transactions on Cloud Computing*, 2022.

[28] Y. Zhu, "Dynamic multi-level access control model based on user trust," *Computer Engineering*, vol. 37, no. 23, pp. 129–131, 2011.

Biography

Wei Sun received his B.S. and M.S. degrees from the School of Information Engineering, Zhengzhou University, China, in 2003 and 2008, respectively. He is currently working in the School of Computer and Information Technology, Xinyang Normal University. His current research interests include access control and system security.

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.