IJNS
International Journal of Network Security

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

Vol. 22, No. 2 (Mar. 2020)

INTERNATIONAL JOURNAL OF NETWORK SECURITY

PUBLISHING OFFICE

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

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

PUBLISHER: Candy C. H. Lin

Volume: 22, No: 2 (March 1, 2020)

International Journal of Network Security

FI-SIFT Algorithm for Exposing Image Copy-Move Forgery with Reflection Attacks

You-Jian Yu[1], Guang-Fu Wang[2], and Jie Zhao[1]
(Corresponding author: Jie Zhao)
School of Computer and Information Engineering, Tianjin Chengjian University, Tianjin 300384, China[1]
Tianjin Surveillance Technology Company Limited, Tianjin 300392, China[2]
(Email: zhaoj@tju.edu.cn)
(Received July 16, 2018; Revised and Accepted Dec. 6, 2018; First Online Sept. 16, 2019)

Abstract

In order to improve the robustness of SIFT algorithm to reflection attack, a flip-invariant SIFT (FI-SIFT) descriptor is proposed to detect copy-move forgery of digital images based on the study on the arrangement of SIFT descriptor after reflection attack in this paper. The proposed descriptor FI-SIFT is designed to improve the invariance to reflection and perform as well as SIFT in other situations. Our method starts by extracting FI-SIFT descriptors for detected SIFT key points in the suspicious image. Then, the g2NN method is adopted to implement multiple key points matching. Next, the possible affine transform between matched key points is estimated to remove the mismatched key points. Extensive experimental results are presented to confirm that our method performs well to detect copy-move forgeries distorted by common attacks including rotation, scaling, reflections and their mixture, especially for the sophisticated scenario, such as multi-objects forgery with combination of reflections.

Keywords: Copy-Move Forgery; FI-SIFT; Image Forensics; Reflection Attack

1 Introduction

Nowadays, we are living in an era of digital revolution which makes it easier for people to access, process, and share digital information. Digital media is playing a significant role in our daily life. However, with the popularity of sophisticated editing tools like Photoshop, it is becoming very difficult to discriminate between an authentic picture and its manipulated version, which poses a serious social problem of debasing the credibility of photographic images as definite records of events. To tackle this crisis of confidence and attempt to restore the credibility in society regarding digital images, the field of digital forensics aiming to reveal forgery operations in digital images is receiving more and more attention.

Among forgery techniques using typical image processing tools, copy-move is the most common type due to its simplicity and effectiveness, where a region of an image is copied and then pasted to another nonintersecting region in the same image to conceal an important element or to emphasize a particular object. The existing copy-move forgery detection methods are based on the fact that, at the end of the manipulation process, the resulting image will have relatively similar areas since the duplicated regions come from the same image. Although not always necessary, some additional operations are often performed on the duplicated regions before pasting them to make the forgery unnoticeable. These operations are used to provide a type of spatial synchronization and homogeneity between the copied region and its neighbors, including rotation, scaling, reflection, illumination modifying, or chrominance modifying. In a practical situation, the processing could be a combination of two or more operations. Thus, the effectiveness of copy-move forgery detection depends on the ability to detect forgery regions with these attacks.

In this work, we proposed a novel flip-invariant SIFT descriptor called FI-SIFT for automatic detection and localization of copy-move forgery regions based on the classical SIFT algorithm in order to resist to reflection-based attacks. We then compared the performance with two state-of-the-art methods to verify the validity of our algorithm. The remainder of the paper is organized as follows. In Section 2, the related research about the past works is introduced. Section 3 presents FI-SIFT descriptor which is the core contribution and novelty of our method. In Section 4, the proposed detection approach is described in detail. Section 5 gives experimental results and the corresponding analysis. Finally, a brief conclusion is drawn in Section 6.

2 Related Work

During the last decade, a large number of techniques have been proposed to address the problem of copy-move forgery detection. First attempt in identifying tampered areas was investigated by Fridrich *et al.* [5] who proposed

a method using discrete cosine transform (DCT) of overlapping blocks and their lexicographical representation to avoid the computational burden. Later, with the purpose of improving robustness and detection efficiency, Huang *et al.* [8], Cao *et al.* [4] and Zhao *et al.* [17] proposed improved block matching detection schemes based on DCT respectively. Luo *et al.* [11] divided image blocks into four sub-blocks, which were evaluated according to the averages of the red, green, and blue color values. Although these methods proved robust to some attacks such as additive noise, Gaussian blurring, and JPEG compression to some extent, they might fail if the duplicated regions underwent geometrical transformations such as rotation or scaling before they were pasted. To solve the above-mentioned problem, several methods have been explored by matching interest point descriptors to identify forged regions as an alternative to the block-matching based detection methods. Such interest point descriptors include scale invariant feature transform (SIFT) [13] descriptor and speeded up robust feature (SURF) [2] descriptor, which are robust to rotation and scaling. Huang *et al.* [7] exploited the SIFT interest point descriptor to reveal the duplicate regions in the forged image through direct matching among these interest points. Furthermore, Amerini *et al.* [1] proposed a SIFT-based detection scheme that could detect and then estimate the geometric transformation used in the copy-move forgery. Similar to Amerini's algorithm, Pan and Lyu [12] proposed another SIFT-based detection algorithm that had the ability to obtain the precise location and extent of the detected duplicated regions using the estimation of affine transformation between matched key points and the correlation of corresponding regions. Xu *et al.* [15] adopted SURF descriptor to detect this forgery with higher efficiency.

Although the feature points-based methods show promising performance, SIFT and SURF feature extraction techniques have two inevitable weaknesses. Firstly, they have difficulties in locating feature points in flat regions and misdetect in uniform regions. In the recent year, Bi *et al.* [3] proposed a multi-level dense descriptor and a hierarchical feature matching method to address this issue. Zandi *et al.* [16] applied an iterative improvement strategy to a new dense descriptor to improve algorithm performance. Secondly, they fail in the situation of reflection as shown in Figure 1. Despite the invariance of SIFT is remarkably robust, it naturally lacks the ability to describe the reflection transformation of feature points. In view of the above problem, Guo X *et al.* [6] proposed a reflection invariant descriptor inspired from SIFT, which resulted in high false alarming rate for authentic images with planar symmetric objects. Warif *et al.* [14] combined the SIFT-based copy-move forgery detection method with symmetry-based matching to enhance the robustness to reflection attack, which was proven to be inefficient by our experiments as a result of double matching in nature. In this paper, the proposed FI-SIFT descriptor was designed to improve the invariance to reflection and perform as well as SIFT in other situations. Particularly, we reorganize the structure of SIFT descriptor, and also adjust the matching strategy accordingly.

3 FI-SIFT Descriptor

Although the classical SIFT descriptor has been proven to perform better than the other existing local descriptors, it does not gain sufficient robustness in the case of reflection. That is to say, as a consequence, the descriptors extracted from two identical but flipped local patches could be completely different in feature space. To overcome the above limitation, we propose a flip-invariant SIFT descriptor, which enhances SIFT with flip invariance property.

Reflection is one of the most common used operations in copy-move forgery, which can be divided into two types: horizontal and vertical reflection. Since vertical reflection image can be obtained by rotating the horizontal flipped version by 180 degrees, the two kinds of reflections are equivalent by rotating the dominant orientations of coordinate system. Thus, in this section we just consider the case of horizontal reflection.

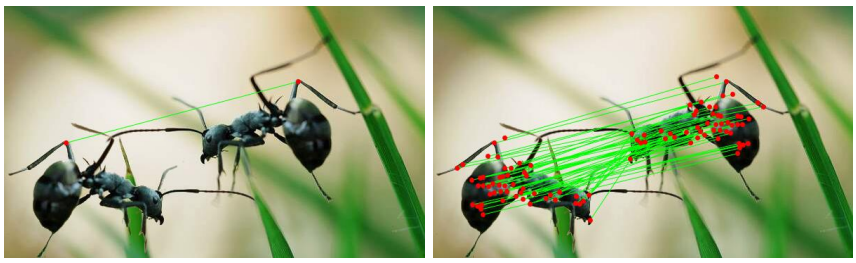
3.1 Analysis on SIFT Descriptor in the Case of Reflection

A SIFT descriptor consists of magnitudes of all the orientations histogram entries in a 4×4 array with 8 orientation bins in each around the corresponding key point. As shown in Figure 2, Figure 2(a) is a key point with its interest region in the original image, and Figure 2(b) is Figure 2(a) in the horizontally reflected image, both of which are after specifying dominant orientation as indicated by the arrow in the figures. Figure 2(d) shows the distribution of 8 orientations in the 14^{th} cell of Figure 2(a). Accordingly, Figure 2(c) is the corresponding version of Figure 2(b).

SIFT employs a fixed order to organize the 16 cells in the interest region. As shown in Figure 2(a), SIFT uses the column-major-order encoding strategy to obtain the key point descriptor. It thus sorts the order of 16 cells as Figure 2(e). However, the order of 16 cells is reversed after horizontal reflection as shown in Figure 2(b). As a result, the original fixed encoding strategy used in SIFT would arrange the 16 cells as Figure 2(f). Although SIFT descriptor is invariant to rotation and scale, and even tolerant to affine transformation, it does not result in the same order in the case of horizontal reflection. Besides, it is not hard to see that the order of 16 cells is the same as Figure 2(b) because of the rotation invariance. For the foregoing reasons, SIFT does not have the ability to resist reflection attack.

3.2 Descriptor Reconstruction

In this paper, we propose a universal encoding technique to generate key point descriptor FI-SIFT, which is also



(a) Detecting result of SIFT (b) Detecting result of FI-SIFT

Figure 1: Comparison of detecting results between SIFT and FI-SIFT in a copy-flip-move distorted image

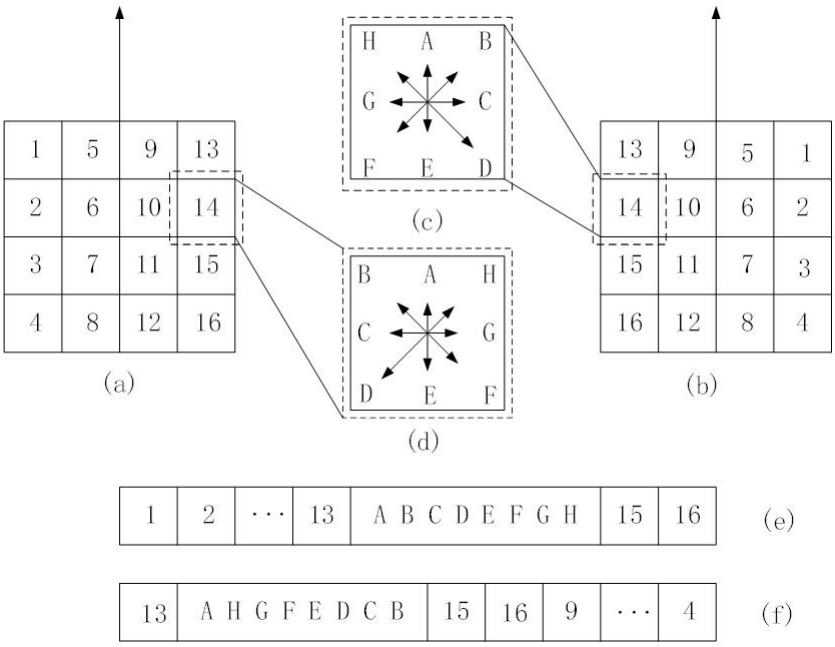


Figure 2: Illustration of the descriptor organization of SIFT in the case of horizontal reflection

invariant to reflection while preserving tolerance to rotation, scale and even affine transformation. First, we determine the location, scale and dominant orientation of key points using the classical SIFT algorithm. Next, for each key point the FI-SIFT descriptor is calculated as follow. Just as there might be multiple descriptors for the same combination of location and scale in the classical SIFT algorithm, FI-SIFT employs two different descriptors to represent the feature of each key point. To be specific, FI-SIFT adopts the anticlockwise order and clockwise order strategies to reorganize the feature descriptor respectively. As shown in Figure 3(a) and Figure 3(b), the 16 cells in the interest region are reorganized in anticlockwise order, and 8 orientation bins in each cell are rearranged into anticlockwise array. In this way, the 16 cells are ordered as Figure 3(e). Similarly, for each key point FI-SIFT reorganizes the 16 cells and 8 orientation bins in each cell in clockwise order as shown in Figure 3(c) and Figure 3(d). As a result, the 16 cells are ordered as Figure 3(f). To summarize, for each key point, FI-SIFT generates two different descriptors as shown in Figure 3(e) and Figure 3(f), where the 16 cells and 8 orientation bins are sorted in anticlockwise and clockwise order respectively.

4 The Proposed Method

In this section, we describe the proposed method in detail to detect duplicated and pasted regions in a tampered image.

4.1 FI-SIFT Features Extraction and Multiple Key Points Matching

In our method, duplicated regions are detected in the illumination domain, thus RGB images are first converted to grayscale images using standard color space conversion. Given a grayscale image, a set of SIFT key points $X = \{x_1, x_2, \cdots, x_n\}$ with their corresponding FI-SIFT descriptors $\{f_1, f_2, \cdots, f_n\}$ are extracted. Since it may happen that the same image region is cloned more than once, multiple key points matching need to be taken into

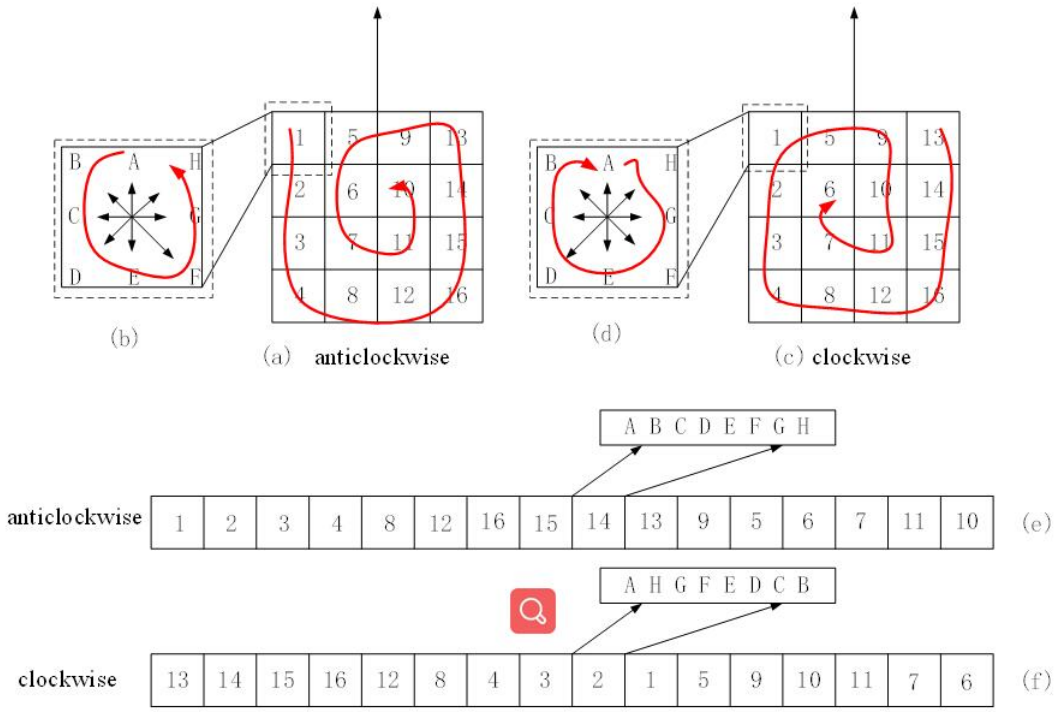


Figure 3: Illustration of the descriptor organization of FI-SIFT

account. For this reason, we adopt g2NN method [1] to implement multiple key points matching. In a high dimensional feature space such as that of FI-SIFT features, for key points that are different from one considered, Euclidean distances of their features share very high and very similar values. Instead, for two similar key points, their features show low Euclidean distances with respect to the others. In the early 2NN method [10], given a key point we need to define a similarity vector $D = \{d_1, d_2, \cdots, d_{n-1}\}$ that represents the sorted Euclidean distances with respect to the other descriptors. The key point is matched only if d_1/d_2 is lower than a preset threshold T_{2NN}. The g2NN method can be viewed as the generalization consisting of iterating the 2NN method between $R_i = d_i/d_{i+1}$ ($i = 1, 2, \cdots, n-2$) until this ratio R_i is greater than a preset threshold T_{g2NN}. If this ratio satisfies $R_k < T_{g2NN}$ $(1 \leq k < n-2)$ and $R_{k+1} \geq T_{g2NN}$, each key point in correspondence to a distance in $\{d_1, d_2, \cdots, d_k\}$ is considered as match points for the inspected key point. We can obtain the set of matched key points by iterating over key points in X.

4.2 Estimating Affine Transform Between Matched Key Points

Next, we need to estimate the possible geometric distortions between duplicated regions and pasted regions. Since almost all the image geometry transforms such as rotation, scaling and shearing can be generalized as affine transform, we model the distortion affine transform of pixel coordinates. Given two corresponding pixel location from a duplicated region and its pasted counterpart as $x = (x, y)^T$ and $\widetilde{x} = (\widetilde{x}, \widetilde{y})^T$ respectively, we can employ a 2-D affine transform to relate them, which is specified by a 2×2 matrix $T = [t_{11}t_{12}; t_{21}t_{22}]$ and a shift vector $x_0 = (x_0, y_0)^T$ as $\widetilde{x} = Tx + x_0$, more definitely

$$\begin{pmatrix} \widetilde{x} \\ \widetilde{y} \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \tag{1}$$

We can obtain unique affine transform parameters T and x_0 by means of randomly selecting three pairs of corresponding key points which are not collinear. Since there are some imprecise matching in practice, Equation (1) may not be satisfied exactly. In order to eliminate deviation as far as possible, we optimize matched key points $(x_1, x_2, \cdots, x_n)$ and $(\widetilde{x_1}, \widetilde{x_2}, \cdots, \widetilde{x_n})$ using least squares objective function to find optimal parameter combination T and x_0 when Equation (2) is minimized.

$$L(T, x_0) = \sum_{i=1}^{N} \|\widetilde{x_i} - Tx_i - x_0\|_2^2 \tag{2}$$

According to the estimated parameters T and x_0, all the putative pairs of matched key points are classified into two groups: inliers and outliers. Specifically, a pair of matched key points $(x, \widetilde{x})$ is an inlier if $\|\widetilde{x} - Tx - x_0\|_2 \leq \beta$, otherwise, it is regarded as an outlier. To remove the impact of mismatched key points and obtain accurate transform parameters, Random Sample Consensus (RANSAC) algorithm is employed to robustly estimate the affine transform parameters, which returns with estimated parameters that generate the largest number of inliers. In our

experiment, we choose default value for N=100 and $\beta = 3$ which lead to better empirical performance.

5 Experimental Results and Discussion

In this section, we evaluate the performance of the proposed method through a comprehensive set of experiments. First, the experimental setup and evaluation metric used in the experiments are introduced. Next, the effectiveness of our method is evaluated in different situations. Then, we compare our method with two state-of-the-art methods which also are developed to improve the invariance to reflection based on SIFT.

5.1 Experimental Setup and Evaluation Metrics

At present, almost all the public datasets for copy-move forgery detection contain only simple geometrical transformation attacks, including translation, rotation, scaling, as well as the mixture of theirs, which lack the corresponding images for reflection attacks. In the recent year, a new dataset called NB-CASIA [14] was created to evaluate the performance of detection methods against reflection attacks. This dataset is composed of 510 images: 255 are original images and 255 are forged images, which the original images are taken from the CASIA v2.0 dataset [9]. The resolution of the images vary from 240 160 to 900 600. The forged images in NB-CASIA consist of translation, rotation, scaling, reflection and the mixture with different parameters as follow.

1) Translation: The duplicated region is translated to the target location with no distortion.

2) Rotation: The duplicated region is rotated with an angle $\theta \in \{20^o, 40^o, 60^o, 120^o, 240^o\}$.

3) Scaling: The duplicated region is scaled with a scaling factor $s \in \{0.6, 0.8, 1.2, 1.4, 1.6\}$.

4) Reflection: The duplicated region is flipped horizontally or vertically.

5) Mixture of attacks: The duplicated region is distorted with a mixture of attacks.

Our experiments were implemented using MATLAB R2015a on an Intel Core i7 3.4GHz processor with 8GB memory. The detection performance was measured in terms of F-score by the image-level, which is defined as

$$F = \frac{2TP}{2TP + FN + FP} \tag{3}$$

where true positive (TP), false negative (FN) and false positive (FP) represent the number of detected forged images, undetected forged images and wrongly detected original images, respectively.

5.2 Effectiveness Test and Comparisons

In the following experiment, we employed NB-CASIA dataset to test the effectiveness of our algorithm. All the forged images in this experiment were without any post-processing operation. Examples of detected results were illustrated in Figure 4. It was noted that the proposed method output detection result maps with color lines connecting all the matching points to identify the duplicated region and forgery region. Although the forged region cannot be localized precisely to pixel level, we can easily identify the tampered region by color lines, which is sufficient for practical detection requirements. Figure 4(a) shows the authentic image. Figures 4(b), 4(c) and 4(d) give the detected results of rotation, scaling and horizontal reflection respectively, which indicate that our method can expose copy-move forgeries effectively in the case of geometric transformations attacks. It is not hard to see that our method can detect stable results by sufficient matching of key points, especially for horizontal reflection attack, which surpasses the classical SIFT algorithm.

Next, we present the analysis of the performance of our method in detecting forged images. The results were compared with two promising methods: Amerini *et al.* [1] and Warif *et al.* [14]. Table 1 shows the overall performance of all the forgery detection methods which were implemented and applied to the NB-CASIA dataset. The input parameters required by the two methods were set as the papers gave. TP, FP and FN values were used to calculate the F-score for each method. As shown in Table 1, our method achieved the best performance compared to the other two methods, which indicated that our method is effective in detecting common transformation attacks, including rotation, scaling, reflections and their mixture. Experimental results show that wrongly detected original images almost have intrinsically similar areas and undetected forged images all have highly uniform region resulting in unreliable feature points.

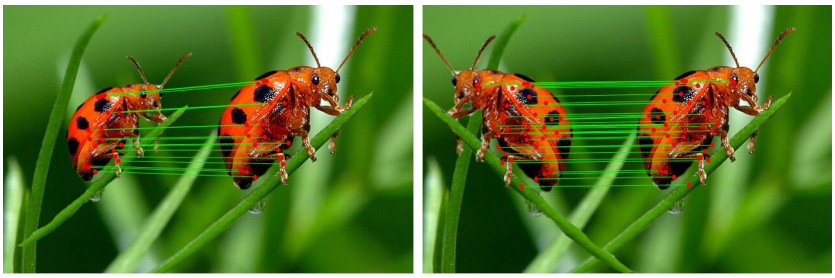
Table 1: The F-score with TP, FP, FN for each method using the NB-CASIA dataset

Methods	TP	FP	FN	F-score
Amerini *et al.* [1]	215	9	40	0.898
Warif *et al.* [14]	237	9	18	0.946
Our method	242	7	13	0.960

5.3 Robustness Test

Based on the previous analysis that showed the effectiveness of our method in terms of reflection attack, in this section we further explore the robustness of the proposed method especially in the case of reflection attack. Thus, we selected an original image at random from NB-CASIA dataset to test the robustness. First, the bird in the image was selected as target area. Then, the target area was

(a) Authentic image (b) Detected result of rotation



(c) Detected result of scaling (d) Detected result of horizontal reflection

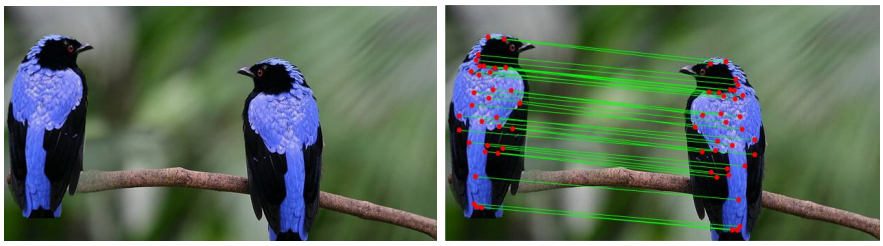
Figure 4: Examples of detected results using our method

copied, flipped horizontally and vertically respectively to create two forged images. The forged images and detected results for horizontal reflection and vertical reflection are shown in Figures 5(a) and 5(b), which indicate that our method performs well in the case of simple reflection.

Next, we created a forged image in the case of vertical reflection with occlusion, where it was actually quite common. The forged image and detected result are shown in Figure 5(c). In view of this kind of situation, the proposed method remains valid. Besides, in practical situations rotation and scaling might be used in combination with reflection attacks, which is a direct challenge to most existing techniques. On account of this, we made the corresponding experiments. Figure 5(d) showed the forged image and detected result, which was created by horizontal reflection and 15 degrees rotation. And Figure 5(e) showed the forged image and detected result, which was generated by horizontal reflection and 70% scaling. Experimental results illustrate that our method is robust enough against combined attacks of geometric transformation and reflection. In the end, we would create a sophisticated forged image involved combined attacks of rotation, scaling and reflection. We copied the bird, flipped it horizontally, scaled it to 75%, rotated it by 17 degrees clockwise, and then pasted it to the left side of the original image. In a similar way, the other duplicate was flipped vertically, scaled to 50%, and then pasted to the right side of the original image. The forged image and the corresponding result detected using our method are shown in Figure 5(f), which demonstrate that our algorithm work well even when the forged image have multiple duplicated regions. The forged image in Figure 5(f) shows the specific scenario that three kinds of attacks including rotation, scaling, reflections and multiple forgery regions coexist simultaneously in an image. Due to the sophisticated scenario in the suspicious image, it is challenge to discern the forgery. To the best of our knowledge, a number of existing methods cease to be effective under the circumstances, however, the detection result of our method is satisfactory.

6 Conclusions

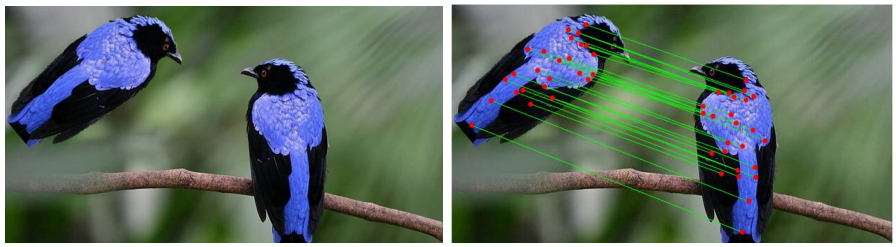
Copy-move forgery detection has been widely studied in the past ten years. However, reflection-based transformation attacks have not been highlighted by prior researchers. The purpose of this work is to achieve high robustness against reflections and any combination of reflection with other geometrical transformation attacks. Thus, we propose a novel feature descriptor called FI-SIFT based on the classical SIFT algorithm which is the core contribution of this paper, and then presented a detection scheme to resist to reflection-based attacks. FI-SIFT cover the reflection-based features by means of modifying the arrangement of feature descriptors. A series of experimental results reveal that the proposed method performs well to detect copy-move forgeries distorted by common attacks including rotation, scaling, reflections and their mixture, especially for the sophisticated scenario, such as multi-objects forgery with combination of reflections. Though having achieved promising performance in detecting sophisticated forgeries with duplicated regions under reflection-based attacks, our method relies on the detection of reliable SIFT key points. For some images with large uniform areas, the SIFT algorithm cannot find sufficient number of reliable key points. In addition, some images have intrinsically identical or similar areas that cannot be differentiated from intentionally pasted copied regions by our method. In the future work, we will con-



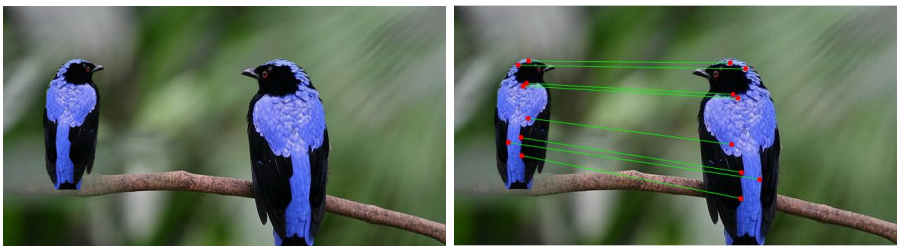
(a) Horizontal reflection

(b) Vertical reflection

(c) Vertical reflection with occlusion



(d) Horizontal reflection with rotation



(e) Horizontal reflection with scaling

(f) Multi-objects forgery with combination of reflections

Figure 5: Examples of forged images and detected results in terms of reflection attacks

sider effective approaches to improve the detection performance for such cases.

Acknowledgments

This work was supported by Natural Science Foundation of Tianjin (Grant # 15JCYBJC15500), China.

References

[1] I. Amerini, L. Ballan, R. Caldelli, *et al.*, "A SIFT-based forensic method for copy-move attack detection and transformation recovery," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1099-1110, 2011.

[2] H. Bay, A. Ess, T. Tuytelaars, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] X. Bi, C. M. Pun, X. C. Yuan, "Multi-level dense descriptor and hierarchical feature matching for copy-move forgery detection," *Information Sciences*, vol. 345, no. C, pp. 226-242, 2016.

[4] Y. J. Cao, T. G. Gao, L. Fan, *et al.*, "A robust detection algorithm for copy-move forgery in digital images," *Forensic Science International*, vol. 214, no. 1-3, pp. 33-43, 2012.

[5] J. Fridrich, D. Soukalm, J. Lukas, "Detection of copy-move forgery in digital images," in *Proceedings of Digital Forensic Research Workshop*, pp. 55-61, 2003.

[6] X. Guo, X.. Cao, "MIFT: A framework for feature descriptors to be reflection invariant," *Image & Vision Computing*, vol. 30, no. 8, pp. 546-556, 2012.

[7] H. L. Huang, W. Q. Guo, Y. Zhang, "Detection of copy-move forgery in digital images using SIFT algorithm," *Proceedings of IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, pp. 272-276, 2008.

[8] Y. P. Huang, W. Lu, W. Sun, *et al.*, "Improved DCT-based detection of copy-move forgery in images," *Forensic Science International*, vol. 206, no. 1-3, pp. 178-184, 2011.

[9] D. Jing, W. Wei, T. Tieniu, "CASIA image tampering detection evaluation database," in *IEEE China Summit & International Conference on Signal and Information Processing*, 2013. (`file:///C:/Users/user/Downloads/06625374.pdf`)

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[11] W. Q. Luo, J. W. Huang, G. P. Qiu, "Robust detection of region-duplication forgery in digital image," in *Proceedings of International Conference on Pattern Recognition*, pp. 746-749, 2006.

[12] X. Pan, S. Lyu, "Region duplication detection using image feature matching," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 857-867, 2010.

[13] E. Silva, T. Carvalho, A. Ferreira, *et al.*, "Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes," *Journal of Visual Communication and Image Representation*, vol. 29, pp. 16–32, 2015.

[14] N. B. A. Warif, A. W. A. Wahab, M. Y. I. Idris, *et al.*, "SIFT-Symmetry: A robust detection method for copy-move forgery with reflection attack," *Journal of Visual Communication & Image Representation*, vol. 46, pp. 219-232, 2017.

[15] B. Xu, J. W. Wang, G. J. Liu, "Image copy-move forgery detection based on SURF," in *Proceedings of International Conference on Multimedia Information Networking and Security*, pp. 889-892, 2010.

[16] M. Zandi, A. Mahmoudi-Aznaveh, A. Talebpour, "Iterative copy-move forgery detection based on a new interest point detector," *IEEE Transactions on Information Forensics & Security*, vol. 11, no. 11, pp. 2499-2512, 2011.

[17] J. Zhao, J. C. Guo, "Passive forensics for copy-move forgery using a method based on DCT and SVD," *Forensic Science International*, vol. 233, no. 1-3, pp. 158-166, 2013.

Biography

Youjian Yu is a lecturer in the department of computer science and technology, Tianjin Chengjian University. He received the M.S. degree from Communication University of China in 2013. Since 2004, he has been working in the school of computer and information engineering, Tianjin Chengjian University. His current research interests include image processing and computer vision.

Guangfu Wang is an engineer in Tianjin Surveillance Technology Company Limited. In 2011 he obtained his M.S. degree from Warwick University in computer science and application, UK. His main research interests are computer vision and deep learning.

Jie Zhao is a lecturer in the department of electronic information engineering, Tianjin Chengjian University. In 2015 he received the Ph.D. degree from Tianjin University in information and communication engineering, China. Since 2009, he has been working in the school of computer and information engineering, Tianjin Chengjian University. His current research interests include digital image forensics and computer vision.

A Coefficient of Variation Method to Measure the Extents of Decentralization for Bitcoin and Ethereum Networks

Keke Wu[1], Bo Peng[2], Hua Xie[2], and Shaobin Zhan[1]
(Corresponding author: Shaobin Zhan)
Shenzhen Institute of Information Technology, Shenzhen, 518172, China[1]
(Email: {wukk, zhansb}@sziit.edu.cn)
Shenzhen Silico Design Technology Co., LTD, Shenzhen, China[2]
(Received Nov. 20, 2018; Revised and Accepted Aug. 5, 2019; First Online Aug. 5, 2019)

Abstract

The most primary advantage of Bitcoin and Ethereum systems is widely understood to be decentralization. However, despite the widely acknowledged importance of this property, most studies on this topic lack quantification, and none of them performs a measurement on the extent of decentralization they achieve in practice. In this paper, we present a coefficient of variation method in probability theory and statistics to quantify decentralization. Using the coefficient of variation, we calculate the dispersion extents of blocks mined and address balances to quantify the extents of decentralization for Bitcoin and Ethereum systems, and the results of calculations indicate that Bitcoin's mining is more approximately 27.3% decentralized than Ethereum with top 19 pool samples, and Bitcoin's wealth is more approximately 16.5% decentralized than Ethereum with 100 samples. Our method can be used to measure the extent of decentralization for any blockchain system.

Keywords: Bitcoin; Blockchain; Coefficient of Variation; Decentralization; Ethereum

1 Introduction

Bitcoin is a digital currency implementation based on blockchain technology that was invented by Satoshi Nakamoto in 2008 [14]. Bitcoin network is the first digital currency system that has been tested in large scale and long time in history. As a public blockchain platform, for adapting to more complex and flexible application scenarios, Ethereum [3] has further extended the functions of Bitcoin for digital currency transactions, supporting the important feature of smart contract. The common advantage of Bitcoin and Ethereum systems is widely understood to be decentralization that does not have any central authority or server and their networks are peer-to-peer. By storing data across its decentralized network, the blockchain eliminates a number of risks that come with data being held centrally.

Since decentralization is the most important property in blockchain, many studies about the decentralization were proposed. Croman and Gencer *et al.* proposed the technical evaluation of blockchain decentralization systems, mainly focusing on the network congestion or delay to evaluate the performance of the blockchain distributed network [5, 8, 12, 16, 20]. They analyze how fundamental and circumstantial bottlenecks in Bitcoin limit the ability of its current peer-to-peer overlay network to support substantially higher throughputs and lower latencies. Their results suggest that reparameterization of block size and intervals should be viewed only as a first increment toward achieving next-generation, high-load blockchain protocols, and major advances will additionally require a basic rethinking of technical approaches. They offer a structured perspective on the design space for such approaches. Within this perspective, they enumerate and briefly discuss a number of recently proposed protocol ideas and offer several new ideas and open challenges.

Gervais *et al.* revealed that there are many important operations and decisions in Bitcoin system which is not decentralized, and they revealed that some nodes control services, decision-making, transactions and mining in Bitcoin system, finally they gave a way to optimize the decentralization of Bitcoin network [4, 9, 10, 13, 17]. They show that the vital operations and decisions that Bitcoin is currently undertaking are not decentralized. They also show that third-party entities can unilaterally decide to "devalue" any specific set of Bitcoin addresses pertaining to any entity participating in the system. Finally, they explore possible avenues to enhance the decentralization in the Bitcoin system. Ron and Shamir analyzed the transaction data of Bitcoin and revealed the occurrence of large transactions in the Bitcoin system at a certain point in time [11, 15, 18].

These existed research papers above are based on data

analysis of Bitcoin transaction behavior to illustrate the drawbacks of the low extent of decentralization in Bitcoin system. Although these studies are mainly concerned about the decentralization of blockchain, none of them performs a measurement on the extent of decentralization they achieve in practice. The closest research work to ours is the paper [19] that only focuses on evaluating a critical value of the number of nodes needed to control over 51% of the network by using a Nakamoto coefficient, rather than quantifying the dispersion of a set of data of blockchain systems, such as blocks mined and address balance and so on.

Herein, we must be able to measure the data dispersion extents of the targets of nodes in blockchain systems before we improve the decentralization. In this paper, we present a coefficient of variation method in probability theory and statistics to measure and quantify the extents of decentralization for blockchain systems. Using the coefficient of variation, we measure the dispersion extents of blocks mined and address balances to quantify the extents of decentralization for blockchain systems. The reminder of this paper is organized as follows. In section 2, we introduce the theory of the coefficient of variation in probability theory and statistics, and the meanings of decentralization in blockchain systems. In section 3, we propose a quantitative measurement method to measure the data dispersion extent based on the coefficient of variation, and in section 4, we calculate the dispersion extents of blocks mined and address balances by using the measurement method. In section 5, we compare the results of the coefficient of variation between Bitcoin and Ethereum systems. Finally, we conclude the paper in section 6.

2 Background

In this section, we first introduce the theory of the coefficient of variation in probability theory and statistics, and we illustrate the meanings of decentralization in blockchain systems to introduce the measurements of decentralization.

2.1 Coefficient of Variation

In probability theory and statistics, the coefficient of variation, also known as relative standard deviation, is a standardized measure of dispersion of a probability distribution or frequency distribution. The coefficient of variation (c_v) is defined as the ratio of the standard deviation (σ) to the mean (μ): $c_v = \sigma/\mu$. It shows the extent of variability in relation to the mean of the population. The coefficient of variation should be computed only for data measured on a ratio scale, as these are the measurements that allow the division operation.

The coefficient of variation is useful because the standard deviation of data must always be understood in the context of the mean of the data. In contrast, the actual value of the coefficient of variation is independent of the unit in which the measurement has been taken, so it is a dimensionless number. For comparison between data sets with different units or widely different means, one should use the coefficient of variation instead of the standard deviation. The value of coefficient of variation is larger, the greater the degree of dispersion.

In this paper, we use the coefficient of variation to measure the extents of decentralization with dimensionless numbers for Bitcoin and Ethereum systems. We consider the extents with the two targets: blocks mined, and address balance.

2.2 Decentralization

Decentralization is the process by which the activities of an organization, particularly those regarding planning and decision making, are distributed or delegated away from a central, authoritative location or group.

In blackchain systems, the decentralization means that no single individual can destroy transactions in the network, and any transaction request requires the consensus of most participants. Bitcoin and Ethereum also have a peer-to-peer network for disseminating block and transaction information. Both Bitcoin and Ethereum also contain full nodes, which serve two critical roles: (1) to relay blocks and transactions to miners (2) and to answer queries for end users about the state of the blockchain. In the Bitcoin and Ethereum protocols, users submit transactions for miners to sequence into blocks. Better decentralization of miners means higher resistance against censorship of individual transactions. Specifically, a decentralized system (like Bitcoin or Ethereum) is composed of a set of decentralized subsystems (like mining, exchanges, nodes, developers, clients, and so on). Srinivasan *et al.* used these six subsystems to calculate a critical value with a Nakamoto coefficient, and to illustrate how many nodes needed to control over 51% of the network in Bitcoin or Ethereum [19].

In this paper, we will calculate the dispersion degrees by two targets (blocks mined and address balance) to measure the extents of decentralization for Bitcoin and Ethereum systems. Please note: you may decide to use different subsystems or targets based on which ones you consider essential to decentralization of the system as a whole.

3 Measurement Method

As mentioned above, we use the coefficient of variation to measure the dispersion degree for dimensionless data sets in Bitcoin and Ethereum systems. Herein, we elaborate the inferring process of coefficient of variation according to the variance and standard deviation, and then we present the formula of the coefficient of variation as a measurement method.

In probability theory and statistics, variance is the expectation of the squared deviation of a random variable

from its mean. Informally, it measures how far a set of (random) numbers are spread out from their average value. Variance is a central role in statistics, where some ideas that use it include descriptive statistics, statistical inference, hypothesis testing, and Monte Carlo sampling. Variance is an important tool, where statistical analysis of data is common. Variance is the square of the standard deviation, the second central moment of a distribution, and the covariance of the random variable with itself, and it is often represented by $Var(X)$. If the generator of random variable X is discrete with probability mass function $x_1 \mapsto p_1$, $x_2 \mapsto p_2$, $\cdots$, $x_n \mapsto p_n$ then

$$Var(X) = \sum_{i=1}^{n} p_i \cdot (x_i - \mu)^2,$$

where μ is the expected value, i.e. $\mu = \sum_{i=1}^{n} p_i x_i$. When such a discrete weighted variance is specified by weights whose sum is not 1, one divides by the sum of the weights. Therefore, in statistics, the variance of a set of n equally likely values can be written as

$$Var(X) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2,$$

where μ is the average value, i.e. $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$.

In statistics, the standard deviation (SD, also represented by the lower case Greek letter sigma σ) is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values. The standard deviation of a random variable, statistical population, data set, is the square root of its variance, i.e.,

$$\sigma = \sqrt{Var(X)} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2}$$

where $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$.

Coefficient of variation is another statistic to measure the degree of variation of observed values in data. When comparing the degree of variability of two or more data, the standard deviation can be used directly if the unit of measurement is the same as the average. If the unit and/or average are different, the standard deviation could not be used to compare the degree of variation, but the ratio of the standard deviation to the average (relative value) should be used to compare. The ratio of standard deviation to average is called coefficient of variation (c_v), i.e.,

$$c_v = \frac{\sigma}{\mu}$$

where $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2}$ and $\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$. Coefficient of variation can eliminate the effect of unit and/or average differences on the comparison of variability between two or more data sets. Therefore, coefficient of variation can be used to calculate the different dimensionless data sets between Bitcoin and Ethereum. More theories about variance, standard deviation, and coefficient of variation, please refer to the probability theory and statistics textbooks.

4 Calculations

According to the measurement method presented above, let's now calculate coefficients of variation for the blocks mined and address balance in Bitcoin and Ethereum networks. We can calculate the decentralized extents each of them according to coefficients of variation.

4.1 Blocks Mined

The quantity of blocks mined reflects the priority to account in blockchain networks. The data is more dispersed (or polarized), the ability of the miners controlling the entire blockchain network is more powerful, and the extent of decentralization of the blockchain network is lower. On the contrary, the data is more average, the ability of the miners controlling the entire blockchain network is weaker, and the extent of decentralization of the blockchain network is higher.

Hence, we use the coefficient of variation as the measurement method to calculate and quantify the degrees of data dispersion for Bitcoin and Ethereum networks, and we can compare the extents of decentralization between them.

4.1.1 Coefficient of Variation of Bitcoin Blocks Mined

We catch the data of Bitcoin blocks mined over the last 7 days from the website btc.com on Oct. 25, 2018, where Bitcoin's data will be updated in real time, as show in Figure 1.

The green frame in Figure 1 is the data top list of Bitcoin blocks mined. The data distribution of top 19 blocks mined in Bitcoin network is as show in Figure 2. We can see that the top 7 miners mined most blocks, and they can influence the decentralized extent of entire Bitcoin network.

According to the measurement method presented above, we use the random variable $X = \{161, 136, 110, 101, 95, 90, 73, 16, 15, 15, 12, 11, 10, 10, 7, 6, 4, 2, 1\}$ and sample number $n = 19$ to calculate the coefficient of

	Pool	Hashrate Share	Hashrate	Blocks Mined	Empty Blocks Count	Empty Blocks Percentage	Avg. Block Size (Bytes)	Avg. Tx Fees Per Block (BTC)	Tx Fees % of Block Reward
1	BTC.com	16.65 %	8.20 EH/s	161	4	2.48 %	1,055,771	0.13199030	1.06 %
2	AntPool	14.06 %	6.93 EH/s	136	1	0.74 %	730,825	0.09653624	0.77 %
3	ViaBTC	11.38 %	5.60 EH/s	110	1	0.91 %	953,433	0.10927257	0.87 %
4	SlushPool	10.44 %	5.14 EH/s	101	1	0.99 %	974,467	0.11446046	0.92 %
5	BTC.TOP	9.82 %	4.84 EH/s	95	1	1.05 %	1,005,207	0.11535621	0.92 %
6	F2Pool	9.31 %	4.58 EH/s	90	0	0.00 %	1,062,745	0.12640051	1.01 %
7	Poolin	7.55 %	3.72 EH/s	73	1	1.37 %	1,069,483	0.12907499	1.03 %
8	Huobi.pool	1.65 %	814.82 PH/s	16	0	0.00 %	940,827	0.11471356	0.92 %
9	BitClub	1.55 %	763.89 PH/s	15	0	0.00 %	877,048	0.10521577	0.84 %
10	DPOOL	1.55 %	763.89 PH/s	15	0	0.00 %	959,886	0.17730335	1.42 %
11	BitFury	1.24 %	611.11 PH/s	12	0	0.00 %	683,110	0.12251172	0.98 %
12	Bixin	1.14 %	560.19 PH/s	11	0	0.00 %	1,000,779	0.07636004	0.61 %
13	58COIN	1.03 %	509.26 PH/s	10	0	0.00 %	633,567	0.05980556	0.48 %
14	WAYI.CN	1.03 %	509.26 PH/s	10	0	0.00 %	1,207,203	0.14989612	1.20 %
15	Bitcoin.com	0.72 %	356.48 PH/s	7	0	0.00 %	800,584	0.07289596	0.58 %
16	BWPool	0.62 %	305.56 PH/s	6	0	0.00 %	1,149,830	0.10385478	0.83 %
17	KanoPool	0.41 %	203.70 PH/s	4	0	0.00 %	992,993	0.19477824	1.56 %
18	BTPOOL	0.21 %	101.85 PH/s	2	0	0.00 %	1,125,270	0.08888516	0.71 %
19	CKPool	0.10 %	50.93 PH/s	1	0	0.00 %	831,358	0.09131175	0.73 %

Figure 1: Miner distribution sorted by blocks mined in Bitcoin network over the last 7 days (Data from https://btc.com/ on Oct. 25, 2018 [1])

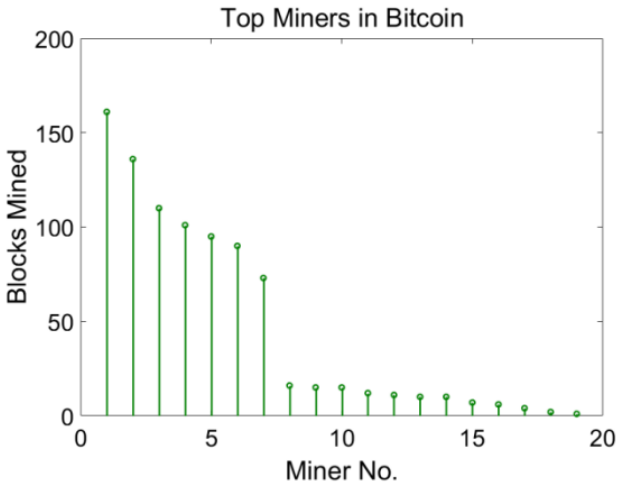


Figure 2: The distribution of top 19 blocks mined in Bitcoin network

variation that is introduced above as follows.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \approx 46.05$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2} \approx 51.38$$

$$c_v = \frac{\sigma}{\mu} \approx 1.12.$$

We obtain the value of coefficient of variation of Bitcoin blocks mined is approximately equal to 1.12.

4.1.2 Coefficient of Variation of Ethereum Blocks Mined

In the same way, we obtain the Ethereum block data over the last 7 days on Oct. 25, 2018 from the website etherscan.io, where Ethereum's data will be updated in real time, as show in Figure 3.

The red frame in Figure 3 is the data top list of Ethereum blocks mined. The data distribution of top 19 blocks mined in Ethereum network is as show in Figure 4. We can see that the only top 5 miners mined most blocks, and they can influence the decentralized extent of entire Ethereum network.

We also use the random variable $Y = \{11389, 9569, 5711, 4334, 3748, 816, 729, 591, 589, 461, 378, 329, 315,$

Rank	Address	Blocks Mined	Percentage
1	0xea674fdde714fd979de3edf0f56aa9716b898ec8 (Ethermine)	11389	26.6073%
2	0x5a0b54d5dc17e0aadc383d2db43b0a0d3e029c4c (SparkPool)	9569	22.3554%
3	0x829bd824b016326a401d083b33d092293333a830 (F2Pool_2)	5711	13.3422%
4	0x52bc44d5378309ee2abf1539bf71de1b7d7be3b5 (Nanopool)	4334	10.1252%
5	0xb2930b35844a230f00e51431acae96fe543a0347 (MiningPoolHub_1)	3748	8.7562%
6	0x2a65aca4d5fc5b5c859090a6c34d164135398226 (DwarfPool_1)	816	1.9064%
7	0xd4383232c8d1dbe0e03bdfab849871fa17e61807	729	1.7031%
8	0x52e44f279f4203dcf680395379e5f9990a69f13c (bw)	591	1.3807%
9	0x70aec4b9cffa7b55c0711b82dd719049d615e21d	589	1.3760%
10	0x2a5994b501e6a560e727b6c2de5d856396aadd38	461	1.0770%
11	0x35f61dfb08ada13eba64bf156b80df3d5b3a738d	378	0.8831%
12	0xcc16e3c00dbbe76603fa833ec20a48f786dfe610	329	0.7686%
13	0x09ab1303d3ccaf5f018cd511146b07a240c70294 (MinerallPool)	315	0.7359%
14	0x005e288d713a5fb3d7c9cf1b43810a98688c7223	301	0.7032%
15	0xb75d1e62b10e4ba91315c4aa3facc536f8a922f5	288	0.6728%
16	0x84a0d77c693adabe0ebc48f88b3fff010577051	256	0.5981%
17	0x6a7a43be33ba930fe58f34e07d0ad6ba7adb9b1f (Coinotron_3)	241	0.5630%
18	0x4bb96091ee9d802ed039c4d1a5f6216f90f81b01 (Ethpool_2)	229	0.5350%
19	0x4c549990a7ef3fea8784406c1eecc98bf4211fa5	198	0.4626%

Figure 3: Miner distribution sorted by blocks mined in Ethereum network over the past 7 days (Data from https://etherscan.io/ on Oct. 25, 2018 [6])

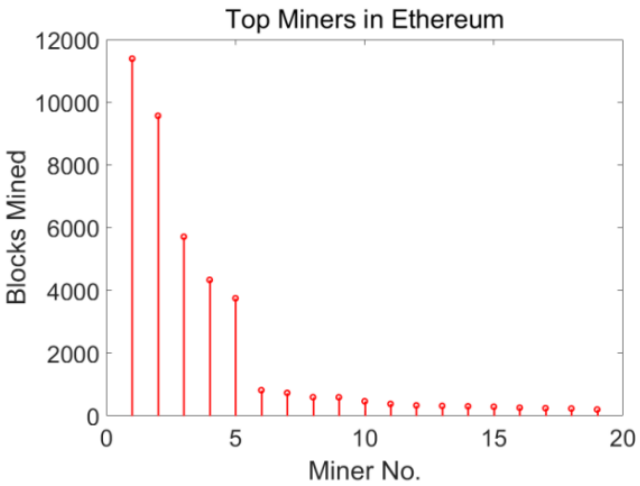


Figure 4: The distribution of top 19 blocks mined in Ethereum network

301, 288, 256, 241, 229, 198} and sample number $n = 19$ to calculate the coefficient of variation as follows.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i \approx 2130.11$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu)^2} \approx 3271.55$$

$$c_v = \frac{\sigma}{\mu} \approx 1.54.$$

We obtain the value of coefficient of variation of Ethereum blocks mined is approximately equal to 1.54.

4.2 Address Balance

This index examines the addresses of the first 100 tokens, and accumulative total tokens as a percentage of the total tokens in the blockchain. We believe that the decentralized blockchain should also decentralize wealth, and the more centralized tokens means that institutions or individuals with a large number of tokens are more likely to manipulate token prices.

4.2.1 Coefficient of Variation of Bitcoin Address Balance

We catch the data of Bitcoin address balance (token) from the website btc.com on Oct. 25, 2018 as show in Figure 5.

The green frame in Figure 5 is the data top list of Bitcoin address balances (tokens). The data distribution

#	Address	Balance	Last 30 Days Tx Count	First Tx Time	Last Tx Time
1	3D2oetdNuZUqQHPJmcMDDHYoqkyNVsFk9r	133,317.23128155	83	2017-01-05 20:34:15	2018-10-25 02:53:50
2	16ftSEQ4ctQFDtVZiUBusQUjRrGhM3JYwe	129,234.33723943	19	2017-12-08 15:51:10	2018-10-25 11:39:32
3	16rCmCmbuWDhPjWTrpQGaU3EPdZF7MTdUk	107,203.07546044	6	2016-02-28 02:00:09	2018-10-25 11:39:32
4	3Cbq7aT1tY8kMxWLbitaG7yT6bPbKChq64	98,042.49937302	8	2017-09-09 00:41:05	2018-10-25 11:39:32
5	3Nxwenay9Z8Lc9JBiywExpnEFiLp6Afp8v	97,848.28496144	6	2015-10-16 22:43:06	2018-10-25 11:39:32
6	183hmJGRuTEi2YDCWy5iozY8rZtFwVgahM	85,947.34736772	6	2018-07-01 21:29:21	2018-10-25 11:39:32
7	1FeexV6bAHb8ybZjqQMjJrcCrHGW9sb6uF	79,957.19635956	6	2011-03-01 18:26:19	2018-10-25 11:39:32
8	1HQ3Go3ggs8pFnXuHVHRytPCq5fGG8Hbhx	69,370.12335943	6	2013-04-10 05:03:36	2018-10-25 11:39:32
9	1PnMfRF2enSZnR6JSexxBHuQnxG8Vo5FVK	66,452.07925125	7	2013-11-23 03:06:31	2018-10-25 11:39:32
10	1AhTjUMztCihiTyA4K6E3QEpobjWLwKhkR	66,378.81086167	6	2014-02-25 13:33:06	2018-10-25 11:39:32
11	1DiHDQMPFu4p84rkLn6Majj2LCZZZRQUaa	66,235.82586355	6	2013-11-23 08:08:37	2018-10-25 11:39:32
12	1EBHA1ckUWzNKN7BMfDwGTx6GKEbADUozX	66,233.75898250	7	2013-11-23 01:05:19	2018-10-25 11:39:32
13	18rnfoQgGo1HqvVQaAN4QnxjYE7Sez9eca	63,600.04702741	20	2014-10-24 18:40:08	2018-10-25 11:39:32
14	34xp4vRoCGJym3xR7yCVPFHoCNxv4Twseo	55,482.91464262	28	2018-10-18 20:59:18	2018-10-25 11:39:32
15	1LdRcdxfbSnmCYYNdeYpUnztiYzVfBEQeC	53,880.05876207	6	2014-05-28 06:49:42	2018-10-25 11:39:32
	...	...		...	
97	1aXzEKiDJKzkPxTZy9zGc3y1nCDwDPub2	10,900.00003664	5	2016-08-04 18:29:39	2018-10-23 19:50:56
98	33ZNiyx5Z5CMkULX7ENvcKKxFNCzGJv5vQ	10,885.20660310	4	2018-07-06 18:17:44	2018-10-23 19:50:56
99	155fzsEBHy9Ri2bMQ8uuuR3tv1YzcDywd4	10,845.61928398	90	2015-01-28 09:25:06	2018-10-26 04:27:50
100	1F34duy2eeMz5mSrvFepVzy7Y1rBsnAyWC	10,770.52537305	3	2011-08-09 06:14:47	2018-10-23 19:50:56

Figure 5: Address Balance in Bitcoin network (Data from https://btc.com/ on Oct. 25, 2018 [2])

of top 100 address balances (tokens) in Bitcoin network is as show in Figure 6.

Herein we can ignore the decimal digits since the values of address balances are very huge. Therefore, we use the random variable $X = \{133317, 129234, 107203, 98042, 97848, 85947, 79957, 69370, 66452, 66379, 66236, 66234, 63600, 55483, 53880, 53000, 52431, 51830, 48500, 45899, 40593, 40474, 40438, 40414, 40054, 40000, 36000, 35612, 34010, 32957, 32841, 32796, 32500, 32490, 31925, 31270, 31085, 31000, 30108, 29999, 29772, 29683, 28151, 27833, 27683, 27496, 26215, 25489, 25409, 25403, 25378, 25302, 25272, 25160, 25064, 24000, 23228, 22891, 22211, 22173, 22100, 21603, 20934, 20263, 20008, 20000, 19414, 17955, 17817, 16252, 16224, 16000, 15746, 15500, 15000, 15000, 15000, 14850, 14627, 14500, 14316, 14000, 13900, 13576 ,13000 12800, 12553, 12000, 11927, 11837, 11800, 11337, 11251, 11102, 10960, 10910, 10900, 10885, 10846, 10771\}$ and sample number $n = 100$ to calculate the coefficient of variation as follows.

$$\begin{aligned} \mu &= \frac{1}{n}\sum_{i=1}^{n} x_i \approx 32906.88 \\ \sigma &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2} \approx 25055.40 \\ c_v &= \frac{\sigma}{\mu} \approx 0.76. \end{aligned}$$

We obtain the value of coefficient of variation of Bitcoin address balances is approximately equal to 0.76.

4.2.2 Coefficient of Variation of Ethereum Address Balance

We also catch the data of Ethereum address balance from the website etherscan.io on Oct. 25, 2018 as show in Figure 7.

The red frame in Figure 7 is the data top list of Ethereum address balances (tokens). The data distribution of top 100 address balances (tokens) in Ethereum network is as show in Figure 8.

Herein we still can ignore the decimal digits since the values of address balances are very huge. Therefore, we use the random variable $Y = \{1538423, 1510066, 1507810, 1483159, 1378754, 1024185, 1004999, 1000000, 988888, 959123, 825000, 817061, 801053, 672785, 672524, 670941, 658443, 560000, 558117, 552124, 549774, 530000, 505000, 493015, 483000, 450000, 450000, 450000, 436000, 427828, 403085, 395433, 380000, 369023, 365003, 350001, 345741, 325000, 319500, 306276, 281380, 275000, 267786, 254248, 250000, 245342, 245300, 234322, 232419, 221195, 220523, 219824, 207438, 204364, 204176, 203527, 203468, 200782, 195524, 193737, 190905, 190121, 189000, 187068, 185591, 183371, 180001, 176650, 172224, 169032, 166602, 164998, 163197, 150000, 142943, 141354, 137476, 135284, 132930, 132288, 131340, 130379, 130000, 128529, 126850, 125266, 123450, 122862, 121861, 120347, 114939, 113762, 110195, 109488, 109381, 108761, 107866, 107371, 106712, 105114\}$ and sample number $n = 100$ to calculate the coefficient

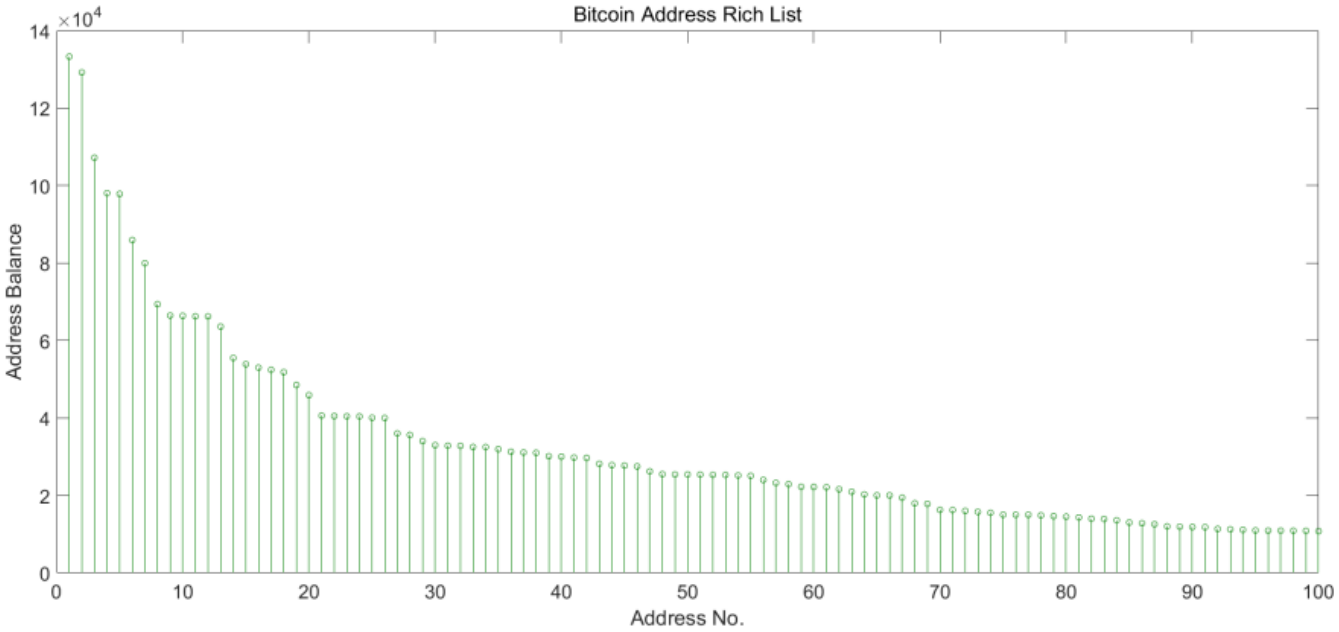


Figure 6: The distribution of top 100 address balance in Bitcoin network

Rank	Address	Balance	Percentage	TxCount
1	0x281055afc982d96fab65b3a49cac8b878184cb16	1,538,423.10656596 Ether	1.49652056%	519
2	0x6f46cf5569aefa1acc1009290c8e043747172d89	1,510,065.64213014 Ether	1.46893548%	499
3	0x90e63c3d53e0ea496845b7a03ec7548b70014a91	1,507,810.43875773 Ether	1.46674170%	446
4	0x742d35cc6634c0532925a3b844bc454e4438f44e	1,483,159.05734310 Ether	1.44276176%	2163
5	0x53d284357ec70ce289d6d64134dfac8e511c8a3d	1,378,754.09306818 Ether	1.34120050%	14994
6	0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2	1,024,180.85887157 Ether	0.99628490%	192870
7	0x61edcdf5bb737adffe5043706e7c5bb1f1a56eea	1,004,999.00001000 Ether	0.97762550%	118
8	0xab7c74abc0c4d48d1bdad5dcb26153fc8780f83e	1,000,000.01146312 Ether	0.97276267%	403
9	0xbe0eb53f46cd790cd13851d5eff43d12404d33e8	988,888.05476810 Ether	0.96195338%	5
10	0xfbb1b73c4f0bda4f67dca266ce6ef42f520fbb98	959,169.02268714 Ether	0.93304381%	6966260
11	0xfca70e67b3f93f679992cd36323eeb5a5370c8e4	824,999.89932905 Ether	0.80252910%	36
12	0xdc76cd25977e0a5ae17155770273ad58648900d3	817,060.63884765 Ether	0.79480608%	138
13	0xe853c56864a2ebe4576a807d26fdc4a0ada51919	801,052.79895972 Ether	0.77923425%	147
14	0xf27daff52c38b2c373ad2b9392652ddf433303c4	672,784.62216252 Ether	0.65445976%	97
15	0x3d2e397f94e415d7773e72e44d5b5338a99e77d9	672,524.35429759 Ether	0.65420658%	81
	...	...	...	
95	0x955a27306f1eb21757ccbd8daa2de82675aabc36	109,380.81181740 Ether	0.10640140%	34
96	0x21346283a31a5ad10fa64377e77a8900ac12d469	108,761.24794052 Ether	0.10579871%	38
97	0xe8507b1532fc44e41b48efe45cf4abf92c5767c3	107,866.00000000 Ether	0.10492785%	1
98	0x3bc643a841915a267ee067b580bd802a66001c1d	107,371.19229881 Ether	0.10444652%	108
99	0xdb8c6862ea4f5cc843c4b3ed75eb8951714b7635	106,711.69360026 Ether	0.10380498%	1309
100	0x692190b4a5d3524b6fed0465e7400c07d09db954	105,113.55510868 Ether	0.10225038%	28

Figure 7: Address Balances in Ethereum network (Data from https://etherscan.io/ on Oct. 25, 2018 [7])

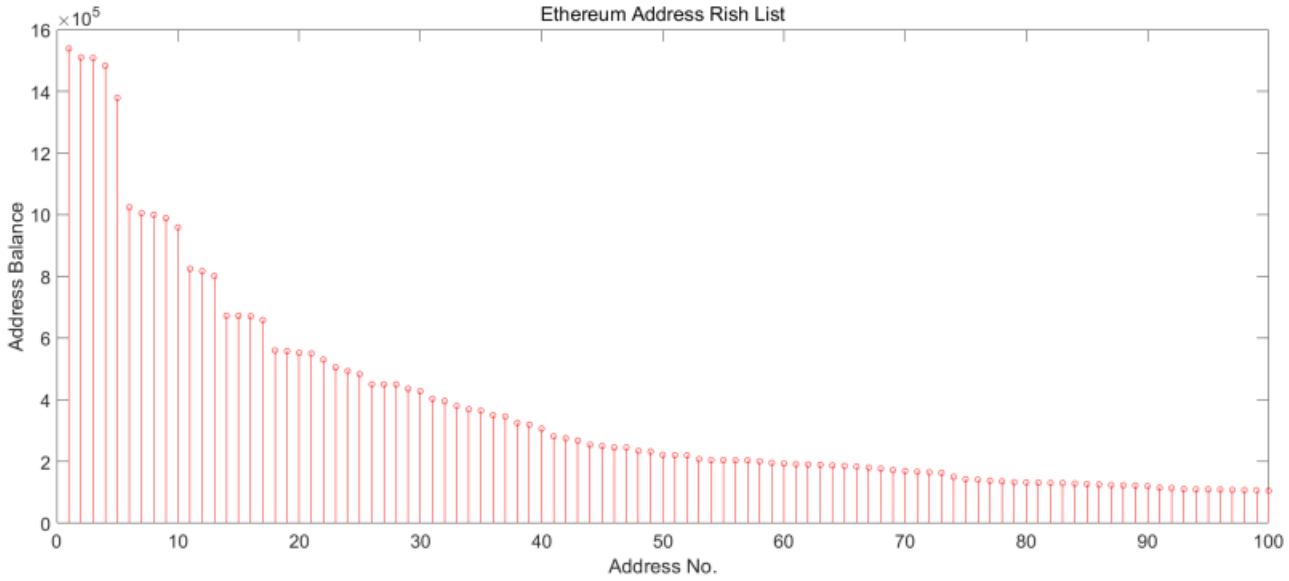


Figure 8: The distribution of top 100 address balances in Ethereum network

of variation as follows.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i \approx 377230.01$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\mu)^2} \approx 345119.60$$

$$c_v = \frac{\sigma}{\mu} \approx 0.91.$$

We also obtain the value of coefficient of variation of Ethereum address balances is approximately equal to 0.91.

5 Comparison and Analysis

We present the comparison of decentralization extents between Bitcoin and Ethereum, and we analyze the centralized and decentralized influences in Bitcoin and Ethereum networks.

5.1 Blocks Mined Index

This index examines how many individual or organizational unions are needed to control more than 50% account power. For example, how many pools in PoW will add up to 50% of the total net power. This index intuitively reflects the difficulty of controlling a digital currency through 51% attacks. We believe that the more decentralized the blockchain, the less likely it is to control the entire blockchain by controlling a few individuals or organizations.

According to the results of calculations of coefficients of variation above, the value of coefficient of variation of Ethereum blocks mined is larger than the value of coefficient of variation of Bitcoin blocks mined, as show in Figure 9.

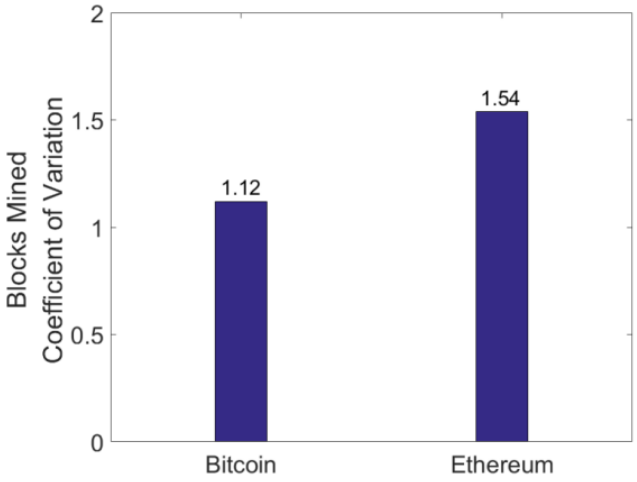


Figure 9: Coefficients of variation of blocks mined comparison between Bitcoin and Ethereum

As shown in Figure 9, Bitcoin mining is more decentralized than Ethereum as measured by blocks mined over the past 7 days. Ethereum mining is somewhat more centralized.

5.2 Address Balance Index

This index is a more controversial indicator, because many people would argue that addresses with a large number of tokens may be exchanges. Actually, those tokens are not exchanges, but are temporarily deposited in

exchanges. Herein, we still believe that address balance decentralization is an important factor in the real decentralization of digital money.

According to the results of calculations of coefficients of variation above, the value of coefficient of variation of Ethereum address balance is also larger than the value of coefficient of variation of Bitcoin address balance, as show in Figure 10.

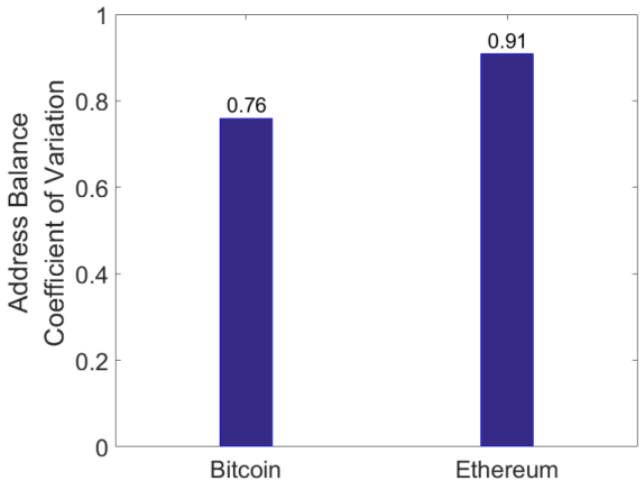


Figure 10: Coefficients of variation of address balances comparison between Bitcoin and Ethereum

As shown in Figure 10, Bitcoin's wealth is more decentralized than Ethereum as measured by address balances. Ethereum's wealth is somewhat more centralized.

5.3 Extents of Decentralization

To sum up the coefficient of variation results of calculations above, we list the quantitative extents of decentralization for Bitcoin and Ethereum networks as the following table.

Table 1: Extents of decentralization

Coefficient of Variation	Blocks Mined (Top 19)	Address Balance (Top 100)
Bitcoin Network	1.12	0.76
Ethereum Network	1.54	0.91
Comparison (Bitcoin is less than Ethereum)	27.3%	16.5%

This table shows that the extents of decentralization of Bitcoin network are 1.12, 1.54 respectively, and the extents of decentralization of Ethereum network are 0.76, 0.91 respectively. Hence, the extents of decentralization of Bitcoin network are more 27.3%, 16.5% large than Ethereum respectively.

6 Conclusions

Decentralization is the most important property of networks like Bitcoin and Ethereum. It is critical to be able to measure the extents of decentralization. More importantly, the coefficient of variation method is one such general measurement method adapting to quantify dispersion for any data set. Therefore, given a proposed blockchain network, we can calculate its coefficient of variation for kinds of targets which you think they are important, and analyze whether this is plausibly a decentralization bottleneck for the network.

Acknowledgments

This work was supported by the Guangdong Natural Science Foundation (Grant No. 2018A030313746), and the Basic Research Project of Shenzhen (Grant No. JCYJ20170817114239348).

References

[1] BTC, *Pool Distribution*, Oct. 25, 2018. (`https://btc.com/stats/pool`)

[2] BTC, *Address Rich List*, Oct. 25, 2018. (`https://btc.com/stats/rich-list`)

[3] V. Buterin, "A next generation smart contract and decentralized application platform," *Ethereum White Paper*, 2013. (`http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf`)

[4] M. Conoscenti, A. Vetr, J. C. D. Martin, "Peer to peer for privacy and decentralization in the internet of things," in *IEEE/ACM 39th IEEE International Conference*, 2017. DOI: 10.1109/ICSE-C.2017.60.

[5] K. Croman, C. Decker, and I. Eyal, *et al.*, "On scaling decentralized blockchains," in *Financial Cryptography and Data Security*, pp. 106-125, 2016.

[6] Etherscan, *Top Miners by Blocks*, Oct. 25, 2018. (`https://etherscan.io/stat/miner?range=7&blocktype=blocks`)

[7] Etherscan, *Top Accounts by ETH Balance*, Oct. 25, 2018. (`https://etherscan.io/accounts`)

[8] A. E. Gencer, S. Basu, and I. Eyal, *et al.*, "Decentralization in bitcoin and ethereum networks," *Financial Cryptography and Data Security (FC'18)*, 2018. (`https://fc18.ifca.ai/preproceedings/75.pdf`)

[9] A. Gervais, G. O. Karame, and S. Capkun, *et al.*, "Is bitcoin a decentralized currency?," *IEEE Security & Privacy*, vol. 12, no. 3, pp. 54-60, 2014.

[10] I. Grishchenko, M. Maffei, and C. Schneidewind, "A semantic framework for the security analysis of ethereum smart contracts," in *Principles of Security and Trust*, pp. 243–269, 2018.

[11] L. Guo, X. Li, and J. Gao, "Multi-party fair exchange protocol with smart contract on bitcoin," *International Journal of Network Security*, vol. 21, no. 1, pp. 71-82, 2019.
[12] Z. Li, J. Huang, D. Gao, Y. Jiang and L. Fan, "ISCP: An improved blockchain consensus protocol," *International Journal of Network Security*, vol. 21, no. 3, pp. 359-367, 2019.
[13] I. Lin and T. Liao, "A survey of blockchain security issues and challenges," *International Journal of Network Security*, vol. 19, no. 5, pp. 653-659, 2017.
[14] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. (`https://bitcoin.org/bitcoin.pdf`)
[15] D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, and C. Yang, "The blockchain as a decentralized security frame-work," *IEEE Consumer Electronics Magazine*, vol. 2, no. 2, pp. 18-21, 2018.
[16] M. Risius, K. Spohrer, "A blockchain research framework," *Business & Information Systems Engineering*, vol. 59, no. 6, pp. 385–409, 2017.
[17] B. Rodrigues, T. Bocek, A. Lareida, D. Hausheer, S. Rafatil, and B. Stiller, "A blockchain-based architecture for collaborative DDoS mitigation with smart contracts," in *Security of Networks and Services in an All-Connected World*, pp. 16–29, 2017.
[18] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *International Conference on Financial Cryptography and Data Security*, pp. 6-24, 2013.
[19] B. S. Srinivasan and L. Lee, *Quantifying Decentralization*, 2017. (`https://news.earn.com/quantifying-decentralization-e39db233c28e`)
[20] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," *IEEE 6th International Congress on Big Data*, 2017. DOI: 10.1109/BigDataCongress.2017.85.

Biography

Ke-Ke Wu received the Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences in 2011. He is currently an associate professor in the Shenzhen Institute of Information Technology. His research interests include blockchain, information security, Cryptography, and side channel analysis technology. (E-mail: kk.wu@sziit.edu.cn)

Bo Peng received the Ph.D. degree from South China University of Technology in 2001. He is currently an advanced engineer in the Shenzhen Silico Design Technology Co., LTD. His research interests include blockchain, information security, Cryptography, and Internet of Things. (Email: pengbo@silico.design)

Hua Xie received the bachelor degree from Tsing hua University in 2004. He is currently an advanced engineer in the Shenzhen Silico Design Technology Co., LTD. His research interests include blockchain, information security, Cryptography, and Internet of Things. (Email: xiehua@silico.design)

Shao-bin Zhan received the Ph.D. degree from Jilin University in 2007. He is currently an associate professor in the Shenzhen Institute of Information Technology. His research interests include blockchain, information security, and Cryptography. (E-mail: zhansb@sziit.edu.cn)

Security Access Solution of Cloud Services for Trusted Mobile Terminals Based on TrustZone

Hui Xia[1] and Weiji Yang[2]
(Corresponding author: Weiji Yang)
Shenyang Normal University, Shenyang 110034, China[1]
Zhejiang Chinese Medical University, HangZhou 310000 , China[2]
(Email: yangweiji@163.com)
(Received Oct. 15, 2018; Revised and Accepted May 17, 2019; First Online Sept. 21, 2019)

Abstract

Trusted cloud architecture provides secure and trustworthy execution environment for cloud computing users, which protects the private data's computing and storage security. However, with the rapid development of mobile cloud computing, there is currently still no secure solution for mobile terminals accessing trusted cloud architecture. Aiming at the above issues, a secure access scheme of cloud services for trusted mobile terminals is proposed. The program fully considers the background of mobile cloud computing applications, uses ARM TrustZone hardware-based isolation technology to build a trusted mobile terminal that could protect cloud service customers and security-sensitive operations on the terminal from malicious attacks. Physical unclonable function (PUF), the key and sensitive data management mechanism is put forward. The secure access protocol is designed based on the trusted mobile terminal and by employing trusted computing technology. The protocol is compatible with trusted cloud architecture and establishes end-to-end authenticated channel between cloud server and the mobile client. Six security properties of the scheme are analyzed and a scenario-based mobile cloud storage example is presented. Finally a prototype system is implement. Experimental results show that the proposed scheme has good expandability and secure controllability. Moreover, the scheme achieves small TCB((trusted computing base) for mobile terminal and high operating efficiency for cloud users.

Keywords: Mobile Cloud Computing; PUF; Secure Access;Trusted Computing; TrustZone

1 Introduction

With the rapid development of cloud computing technology, mobile terminal equipment, international mobile communication technology and mobile internet applications, the concept of mobile cloud computing (MCC) are gradually affecting people's daily life. International Mobile Cloud Computing Forum [4] and Intel Aepona [3] give the relevant definition, mobile cloud computing is a comprehensive technology for mobile terminal devices to outsource data processing and data storage to resource-rich computing platforms through mobile cloud applications. Mobile cloud computing can be effective reduce the cost of computing resources, storage resources and electricity and can enhance the usability of complex applications in mobile terminals [2]. Mobile cloud computing presents software as a service's (referred to as SaaS) level [10] to the users, users use mobile devices to thin client software or web browser through a wireless network to access remote cloud services. In recent years, mobile cloud computing promotion areas include cloud office, cloud mail, cloud storage, cloud payment, cloud games and cloud video. Companies have also launched corresponding support technologies and products for mobile cloud computing, including Apple's iCloud, Google's Cloud Console, Microsoft's OneDrive and Amazon's App-Strea,which have greatly improved the convenience of mobile users to experience cloud services. The main contributions of this paper are as follows:

- For the mobile cloud computing scenario, a method based on TrustZone technology to build a trusted mobile terminal is proposed to ensure the security and reliability of the cloud service client program and related sensitive operations in the mobile terminal.
- A key and sensitive data management mechanism based on PUF technology is proposed. This mechanism cooperates with TrustZone technology to provide a trusted root function for trusted mobile terminals and cloud service security access.
- A trusted mobile terminal cloud service security access protocol is proposed, which uses trusted computing technology to establish end-to-end bi-directional authentication channel between the cloud server and the mobile client to protect user data and cloud service access requests's authentication, confidentiality

and integrity in the access' process. The protocol is compatible with the trusted cloud architecture.

Section 1 of this paper discusses the work in this area. Related work and some research achievement about the topic is discussed in Section 2. Section 3 presents preliminary knowledge about key and sensitive data management for trusted mobile terminal cloud.Section 4 expatiates on the design of secure access scheme for trusted mobile terminal cloud: trusted mobile terminal architecture, and cloud service security.Example and Scheme evaluation based on the system is given in Sections 5 & 6. Section 7 summarizes the full text and looks forward to future research.

2 Related Work

In order to solve the pro blem of information security from the underlying computer hardware, the concept of trusted computing has been proposed and popularized in scientific research and industry. Trusted Computing Group (TCG) has launched a TPM (Trusted Platform Module) security solution for x86 hardware platform. Trusted platform module(referred to as TPM), the TCM's main TPM1.2 specification [16], was revised several times in 2009 to receive the ISO in 2009. In 2013, TCG officially released a new security solution TPM2.0 standard [17]. In China, the National Cryptology Authority in 2007 proposed a trusted cryptography module (TCM) [19] with independent intellectual property rights and related interface specifications. As a basic security technology, an important application scenario of trusted computing is to construct a trusted virtualization platform. The virtual trusted platform module(vTPM [6]) can be used to protect the security of virtual machine monitor. TrustVisor [14] provides trusted services for isolated code by creating a virtual TPM instance . Because of the advantages of security isolation, security intervention and data protection, infrastructure virtualization technology is widely used in cloud computing architecture, and it is a hot research area in recent years to build trusted cloud computing environment by trusted virtualization technology. Literature [15] outlines the concept of trusted cloud computing platform (TCCP), which provides a closed operating environment for user virtual machines by extending the functionality of the trusted platform to the cloud infrastructure, thus user data confidentiality and integrity can be effectively protected.Wuhan University Professor Zhao Bo et al. [23] summarizes the trusted cloud computing environment to build the technical methods and challenges. TrustCloud [1] designed a framework for trust building and security auditing for cloud computing. Cloud Terminal [9] uses a trusted authentication method to outsource the data-processing security of user-sensitive applications to the cloud service provider, where the user's local host only displays the interface. CloudProxy [8] uses trusted computing technology to establish an end-to-end trusted connection between the cloud host and the user's host to protect the security of user data during transmission and cloud operations. However, the above-mentioned building methods of trusted cloud computing environment are designed for x86 hardware platform. How to access the trusted cloud environment safely and effectively by mobile terminal equipment is still a problem to be solved.

In the rapid development of mobile cloud computing today, mobile cloud computing security has attracted more and more people's attention. Related research [7,13] pointed out that: the mobile user's private data in the mobile terminal, cloud host and communication channel on the confidentiality and integrity, are the key to mobile cloud computing security. Trusted virtualization technology and cloud architecture can protect user's data in the cloud host, but lack of suppling and supporting the mobile terminal and mobile network communications in the protection of user data and aim at cloud environment for the design of trusted solutions. Literature [20] gives a trusted security isolation method based on mobile operating system access control strategy, which is based on the premise of mobile operating system security. However, the successful use of Android system vulnerabilities in the implementation of the attack is endless, the operating system itself does not provide high-intensity security. For the research of trusted mobile terminals, the Mobile Trusted Module (MTM) specification [18] has been released for mobile terminals, but it is not promoted in the mobile industry due to the need to rely on additional hardware modules, and the specification has not been promoted in the mobile industry.

3 Problem Statement and Preliminaries

3.1 Root Key Seed Extraction

In this paper, the literature [24] proposed SRAM (Static Random Access Memory) PUF technique to extract the root key seed S, S is a piece of unique bit string randomly selected by the mobile terminal manufacturer M in the production process of the device, M uses the physical characteristics of the SRAM-specific area in the mobile terminal T to store S, therein. S is only reproduce from the SRAM PUF component every time T is normally powered up and is safely cached by the key manager in the SW (Secure World)). S confidentiality is strictly protected by TrustZone.

3.2 Key Derivation

In the mobile terminal SW, KDF (key derivation function) is the key manager of the trusted service and has a key generation function, which is a deterministic mapping: $\widetilde{S} \times \widetilde{P} \longrightarrow \widetilde{K}$, where $\widetilde{S}$ is the key seed space, $\widetilde{P}$ is a set of string parameters for declaring the usage of the key, and $\widetilde{K}$ is the space for generating the keys, using the KDF and the root key seed S, a public private

key pairs (dpk_T, dsk_T) that uniquely identify the identity of the mobile terminal can be generated, the generation method is:

$$(dpk_T, dsk_T) \longleftarrow KDF_s(\text{"}identity\text{"}).$$

Similarly, srk (Storage Root Key) can be generated in the form $srk \leftarrow KDFs(\text{"}storage\_root\text{"})$. srk is used to further generate a storage key to store and protect the actual sensitive data, and this set of storage key can enhance the isolation and security. It is worth emphasizing that the private keys of all storage keys and device keys generated here never leave the SW and are not stored on the nonvolatile memory of the mobile device. If needed, they will be used in the same way as the KDF way refactoring, which can reduce the risk of key loss.

3.3 Sensitive Data Management

A variety of storage keys derived from srk encapsulate and store the public key apk of the application service provider A and the key package $(ID, k^{Enc}, k^{MAC}, n_i)$ is required for the cloud service session. The key data's specific meaning and usages will be described in detail in Section 4.1. The encapsulation operation is performed in the data processor of the SW trusted service. The data processor implements the data encapsulation function $Data\_Seal()$. The encapsulated data block can be stored in the public nonvolatile memory of the device. In this paper, $MAC_k(m)$ represent the calculation of the message authentication code for the data m by using the key k; $Enc_k(m)$ means that the data m is encrypted with the key k, and the symmetric and asymmetric encryption can be expressed according to the type of k; $Sign_k(m)$ represents the signature operation; $\|$ indicates the connection of the data. The following are the specific encapsulation methods:

- For the public key apk, the encapsulation only needs to protect the integrity of the public key to prevent mobile applications are malicious tampering caused public key damage, the steps are as follows:

$$\begin{aligned} mk_{apk} &\longleftarrow KDF_{srk}(\text{"}storage\_key\text{"}, \text{"}MAC\text{"}, apk), \\ blob_{apk} &\longleftarrow Data\_Seal(\text{"}MAC\text{"}, mk_{apk}, apk); \end{aligned}$$

  where KDF_{srk} means storage_root from key derivation function, among them,

$$blob_{apk} = apk \| MAC_{mk_{apk}}(apk).$$

- For the key package $(ID, k^{Enc}, k^{MAC}, n_i)$, when packing,we need to protect their confidentiality and integrity to prevent the rival's theft or tampering, the steps are as follows:

$$\begin{aligned} (sk_{ID}, mk_{ID}) &\longleftarrow KDF_{srk}(\text{"}storage\_key\text{"}, \\ &\qquad \text{"}Enc + MAC\text{"}, ID), \\ blob_{ID} &\longleftarrow Data\_Seal(\text{"}Enc + MAC\text{"}, sk_{ID}, \\ &\qquad mk_{ID}, (ID, k^{enc}, k^{MAC}, n_i)); \end{aligned}$$

$$\begin{aligned} blob_{ID} = {} & Enc_{sk_{ID}}(ID, k^{Enc}, k^{MAC}, n_i) \| \\ & MAC_{mk_{ID}}(Enc_{sk_{ID}}(ID, k^{Enc}, \\ & k^{MAC}, n_i)). \end{aligned}$$

With the storage key refactored in the key manager, the data processor can call the Data_Unseal() function to recover and validate the sensitive data from the corresponding data block.

4 Design of Secure Access Scheme for Trusted Mobile Terminal Cloud Service

4.1 Trusted Mobile Terminal Architecture

With TrustZone and PUF technology, we designed a trusted mobile terminal architecture for cloud computing scenarios. On the basis of the existing mobile terminal hardware architecture, our trusted terminal program is based on software design and implementation as a focus, targeting low cost, flexibility and scalability. Figure 1 shows the proposed trusted mobile terminal architecture and the interaction between the various components of the details in this paper.

Using the method given in literature [21], it is possible to construct TEE (Trusted Execution Environment) safely and efficiently in the TrustZone SW, The TEE implemented in the SW is physically isolated from the universal mobile system environment implemented in the NW (Normal World), running a custom TEE OS in OS to executing security-sensitive program code. There is a Universal Mobile OS running on the NW, which can be an Android or iOS system capable of performing regular mobile applications, and the functions of each component described in details below.

1) Trusted agent: The trusted agent interacts directly with the mobile application in the NW. The component receives a trusted service request from the mobile application, assembles the command for calling the trusted service component in the SW (Secure World) according to the request type, and prepares for the substantial security operation in the SW. The component contains the following two subcomponents:

   - Software stack: Mobile applications to provide high-level trusted service interface, responsible for resolving the application request data, and return the results of service response;
   - Command caller: Assembling the trusted service invocation command, interacting with the trusted service component in the SW, and transmitting the command through the canonical Global Platform TEE client API [5], requesting the NW to switch to the SW by means of

the NW underlying driver and wait for the data to return.

2) Trusted service: Trusted service component is a core component of Trusted Mobile Terminal, which not only realizes Trusted Computing related functions, such as Trust Root Rendering, Key and Sensitive Data Management and Trusted Environment Authentication, but also implements the security crediting protocol execution in mobile terminal Logic. The code execution of this component is protected by the TrustZone quarantine mechanism and consists of the following five subcomponents:

   - API functions: Receive trusted service requests from trusted agents in the NW, parse command data, pass operational instructions to the logic engine, and wait for the results to return to the trusted agent;
   - Key manager: Use the root key seed extracted from the SRAM PUF to produce a variety of cryptographic keys and provide the key to the data processor for user;
   - Data handlers: In order to prevent adversaries from forging security parameters (usually user names and passwords), the data processor receives only the parameter input from the mobile application trusted cell in the SW and passes the parameters to the logic engine. In addition, the subcomponent is also responsible for the encapsulation and de-encapsulation of sensitive data, encapsulated data can be stored in the general non-volatile memory of the mobile device;
   - Crypto library: It provides cryptographic algorithm support for key manager, data processor and logic engine, which implements symmetric and asymmetric encryption,decryption and signature verification algorithms and a variety of message digest algorithms.
   - Logic engine: Obtains the necessary parameter input from other sub-components. According to the designed security access protocol logic, it performs the security-sensitive trusted service operation of the mobile terminal and outputs the execution result. In addition, the subcomponent implements the load measurement and start-up control of the application process in SW.

3) Mobile application (App) and App trustlet: When the mobile user wants to access the cloud services at C, whether it is browser or client mode, you need to start the appropriate mobile applications. The mobile application provided by the application service provider A comprises two parts: a mobile application running in the NW and a mobile application trustlet running in the SW. App only provides the user with a graphical user interface (GUI) and basic non-security-sensitive functions, App trustlet is responsible for collecting and pre-processing sensitive data information needed to access the cloud service and presenting it to the trusted service for operation of the secure access protocol. When the App needs to access the cloud service through the implementation of secure access protocol, it calls trusted agent software stack to make trusted service requests. After TrustZone uses the system interrupt to complete the NW to SW switchover, the Trusted Service will load the startup App trustlet whose code integrity is measured by the logical engine of the trusted service. Based on our previous study work [22], once the App trustlet is found to have been tampered with by an adversary using the whitelist mechanism in the SW, it can be disabled. When the App trustlet is properly started, the user can send cloud service user name and password and other sensitive data information into App trustlet in the security mode, and then hand over to the trusted service for processing. The App implements the communication with the App trustlet through the inter-domain communication mechanism [12] provided by TrustZone, where the mobile application design conforms to the current TrustZone's normal application mode.

4) Components in the kernel: In SW's TEE operating system kernel, there is a drive component SW-Driver; in the NW mobile operating system kernel, there is a drive component NW-Driver. The above two driver components are used to handle the request and response commands of the two world switching in the TrustZone, which contains the communication data of the two. As an implementation of the security monitor defined by TrustZone, the monitor is located in the system kernel of the SW, which controls the underlying hardware to perform the specific actions of the TrustZone world switch. In addition to these special components, the OS kernel implementation in the NW has a variety of generic hardware's driver, including network communication drivers, which is relied by the data communication between the trusted mobile terminal and the cloud service.

5) Components in the hardware: Trusted mobile terminal hardware support ARM TrustZone extension technology, by the protection of the technology, the SRAM PUF physical components located in the hardware can only be accessed by the SW, and the software algorithm of the PUF is implemented by the key manager of the trusted service.

4.2 Cloud Service Security Access Protocol

The interactive participant entities of the cloud security access protocol are T, A, and C, and an overview of the protocol implementation is shown in Figure 2 under normal circumstances. In a certain period of time, when T

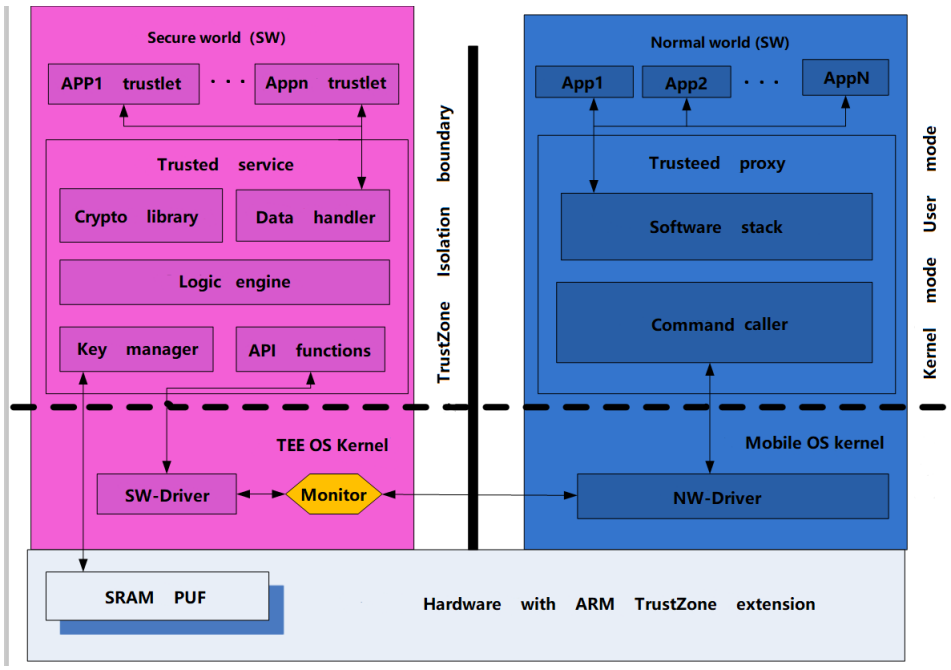


Figure 1: Architecture of trusted mobile terminal for cloud computing

first accesses a cloud service located at C, it first sends an authorization request for access service to A; after charging and legitimacy authentication, A generates a session key package and issues it to T; at the same time, A will be certified to the user data sent to C through the security channel, here, we do not distinguish the user management host in C and service operations host; after obtaining the authorized session key package, T sends the service access request to C by using the relevant key and the authentication information; after C verified the request, the verification results will return to T; then complete access authentication, T and C start a normal cloud service interaction. The secure access protocol we provide can be interfaced with the trusted cloud architecture to achieve bidirectional authentication between the T and C secure execution environments. This paper assumes that C adopts the trusted cloud architecture proposed in literature [8], when returning the T service access request verification result, C will attach the integrity metric of the cloud service program from cloud host.

The secure access protocol with the trusted mobile terminal as the core consists of 4 parts, authorization application, access request, authentication response and authorization revocation. Among them, the authorization application is only executed in three cases: (1) The first time users use T to request access to cloud services; (2) The last authorization application has expired; (3) Due to network error or malicious attack, authorization was revoked. After successfully executing the authorization request, the user can use T to request access to the cloud service several times within a certain period of time. The access request and authentication response of the protocol can be executed multiple times.

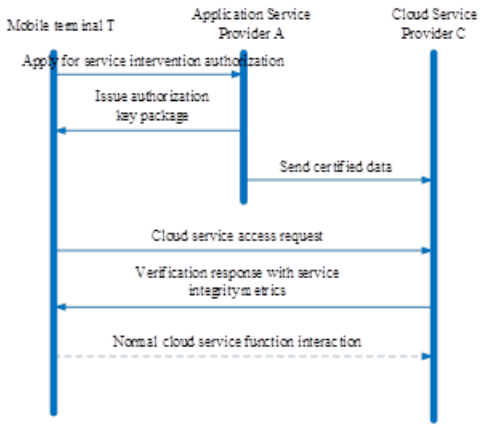


Figure 2: An overview of secure access protocol under normal condition

4.2.1 Application for Authorization

In this part of the agreement, the user using T sends a cloud service access authorization request to A, and A verifies the relevant parameters in the application. After determining the legality of the T and its users, generate a secret for the future conversation between T and C. The key package is sent to both parties, as follows:

1) The user operates the App in NW to request access to the cloud service, TrustZone switches to SW, and T calls KDF to generate the message integrity protection key mkauth, which is used to protect the integrity of data communication when A sends a session key package to T. The key generation method is as follows:

$$mk_{auth} \longleftarrow KDF_s(\text{"}session\_key\text{"}, \text{"}MAC\text{"}, r),$$

among them, r is the key generated by the key manager for generating different mk_{auth}.

2) When T loads the launch App trustlet in the SW, it performs integrity metrics on the loaded code, and uses the hash function to get the metric $\mu(app)$. Based on our whitelist mechanism [12], we can find the tampering of the App trustlet, if tampered, the agreement will terminate execution.

3) The user input user name user and password pswd of the login cloud service to the SW's App trustlet, and the App trustlet calculates the password hash value $H(pswd)$, and sends it to the data processor of the trusted service along with the username, and App trustlet will be shut down by the trusted service.

4) The logic engine of the trusted service in SW calls the authorization application API: $Apply()$, generates the authorization request message m_apply: $m\_apply \longleftarrow Apply(Cert_T, dsk_T, blob_{apk}, mk_{auth}, \mu(app), user, H(pswd))$. The API specifically performs the following operations:

   a. Call the key manager to reconstruct the device key private key dsk_T;

   b. Call the key manager to reconstruct the apk storage protection key, call the data processor to unblock $blob_{apk}$, then get the correct apk;

   c. Call the signature function Generate the signature:

$$w := Sign_{dsk_T}(mk_{auth}, u(app), user, H(pswd));$$

   d. Call the encryption function to generate the end of the communication message:

$$m\_apply :== Enc_{apk}(Cert_T, mk_{auth}, u(app), user, H(pswd), w).$$

   Here, apk is derived from A, in fact, A generates a pair of public-private key pairs (apk, ask) for each application issued, apk can be extracted by T for authentication communication with A when the application is installed; in addition, apk can uniquely identify an application.

5) T switches from SW to NW, sends m_apply to A, A decrypts the message with its own private key ask, uses the public key M certificate $Cert_T$ issued by authority, and obtains the device key public key dpk_T of T, verifying the signature of the relevant data, extracting valid data tuples of the authorization request message:

$$(mk_{auth}, \mu(app), user, H(pswd)).$$

6) A verifies the integrity measure $\mu(app)$ of the App trustlet in T according to their published application code, and uses the user and $H(pswd)$ verify the legitimacy of the user account, you can check the balance of the account: if the relevant authentication fails, returns the message and reason why the application of T failed; if all authentication passes, A generates a session key packet tuple $(ID, k^{Enc}, k^{MAC}, n_0)$ for T and C, where ID uniquely identifies the key package; k^{Enc} is used to protect the confidentiality of the conversation; k^{MAC} is used to protect the integrity of the session; n_0 is a randomly selected nonce value and to prevent replay attacks. T and C access once correct service connection, each of the value will plus 1, so the $(i+1)^{th}$ connection get n_i;

7) A is the authorized session key package for T, and A will use the dpk_T encryption to generate σ after signing the session cipher package:

$$\sigma := Enc_{dpk_T}(apk, (ID, k^{Enc}, k^{MAC}, n_0), Sign_{ask}(ID, k^{Enc}, k^{MAC}, n_0)).$$

Among them, apk is used to identify the application corresponding to the encrypted data, and A generates an authorization response message in the following manner:

$$m\_reply := \sigma \| MAC_{mk_{auth}}(\sigma).$$

8) A sends m_reply to T and sends (ID, k^{Enc}, k^{MAC}, n_0) along with user and $\mu(app)$ to C through the security channel. If C finds the same user's previous session in the database package through user, the old key package is deleted. The lifetime of the session key package can be set to a different length depending on the security sensitivity of the cloud service, which can be 1 day, 7 days, or 30 days. The validity period is recorded at C, and if the expiration date is exceeded, the session key package will automatically invalidated, and the expired session key package is periodically cleaned and deleted by C;

9) After T receives m_reply, it switches to SW, and the trusted service parses m_reply. After verifying

the message integrity and signature correctness, the extracted session key package $(ID, k^{Enc}, k^{MAC}, n_0)$ is encapsulated as a $blob_{ID}$ and stored in a mobile device.

4.2.2 Access Request

In this part of the protocol, T uses the session key package to send a cloud service access request to C. The specific steps are as follows:

1) T reloads start App trustlet, and measures the loaded the integrity of the code once again, the use of hash function to obtain the metric $\mu'(app)$, you can use the white list mechanism to check again whether the App trustlet is tampered;

2) The logical engine of the trusted service in SW calls the cloud service access request API: $Request()$ generates $m\_request$:

$$m\_request \Longleftarrow Request(blob_{ID}, \mu'(app)).$$

The API specifically performs the following operations:

1) Call the key manager to reconstruct the storage protection key of the session key package, and call the data processor to unlock the $blob_{ID}$ to get the correct session key packet tuple $(ID, k^{Enc}, k^{MAC}, n_i)$;

2) Generate cloud service access request communication message $m\_request$:

$$\begin{aligned} m\_request \quad := \quad & ID \| Enc_{k^{Enc}}(\text{"}request\text{"}, n_i, \mu'(app)) \| \\ & MAC_{k^{MAC}}(ID, Enc_{k^{Enc}}(\text{"}request\text{"}, \\ & n_i, \mu'(app))). \end{aligned}$$

Among them, ID is used to tell C which session key packet to use to decrypt and verify the message; the request is a command parameter that identifies the execution of the cloud service at the request C. Here, the command to apply for access to the cloud service is indicated.

3) T switches to NW and sends $m\_request$ to C.

4.2.3 Verify the Response

In this part of the protocol, C uses the session key package to parse the access request from T, and returns the verification result and the integrity measure of the cloud service executive to T, and the concrete steps are as follows:

1) After receiving the $m\_request$, C finds the corresponding session key package in the database according to ID, to check whether the session key package is still valid, expires or revokes, or fails to find the key package. In case, C sends a response marked as verification failure to T, and T will re-execute the authorization application agreement.

2) C uses the legitimate key package parsing $m\_request$, and matches the value of n_i to the current nonce value of the session key package record in the database, if they are not the same, then return the validation failure response to T, T will re-execute the authorization request protocol.

3) C matches the $\mu'(app)$ in $m\_request$ with the original $\mu(app)$ of the corresponding session key package in the database, if not the same, that App Trustlet in T is likely to have been tampered, C will reject the access request of T, and the response to the authentication failure is returned.

4) After the above 3-step verification, C uses the security method in the trusted cloud architecture to generate an integrity metric $\mu(csp)$ of the program running the T-requested cloud service in the virtual machine. In some specific application scenarios , the metric may be derived from a hash metric for the entire virtual machine image, for $\mu(csp)$ certification methods can refer to the specific agreement of the cloud architecture.

5) C generates a communication message $m\_response$ for the verification response:

$$\begin{aligned} & m\_response \\ := \quad & ID \| Enc_{k^{Enc}}(\text{"}response\text{"}, passed\text{"}, n_i, apk, \\ & \mu(csp)) \| MAC_{k^{MAC}}(ID, Enc_{k^{Enc}}(\text{"}response\text{"}, \\ & \text{"}passed\text{"}, n_i, apk, \mu(csp))). \end{aligned}$$

6) After C sends $m\_response$ to T, it updates the nonce value: $n_{i+1} = n_i + 1$, and sets the nonce limit value of the access service to be $limit\_n = n_i + j$, j represents the T access to the cloud service after the verification. If $n_{i+x} > limit\_n$ of C at the x^{th} access, T needs to re-execute the access request protocol to let C update and set $limit\_n$;

7) After receiving $m\_response$, T switches to SW to parse and verify it. Meanwhile, it updates the nonce value of the itselves' session key package: $n_{i+1} = n_i + 1$. The security method with trusted cloud architecture can verify the integrity of the cloud service program through $\mu(csp)$. After the verification is passed, the App trustlet transmits command parameters to the trusted service according to the specific function of the user requesting the cloud service. The trusted service uses the session key package to assemble the cloud service operation command, and communicates with C to complete the specific Cloud service features.

5 Mobile Cloud Storage Application Examples

Based on the proposed secure access solution for mobile terminal cloud services, we designed a mobile cloud stor-

age application instance MCFile. MCFile security objectives including: (1) protecting the confidentiality and integrity of mobile terminal users sending and receiving files to the cloud server; (2)The cloud server can enforce mandatory security authentication and access control policies for mobile terminal users' access requests and file access. The cloud storage service operation commands implemented by MCFile can be: create files (create). Delete files, write files, read files, addrights, and removerights. Assuming that there are two users whose user names are user1 and user2, user1 stores the file named Fuser1 in the cloud. Then, after the user1 successfully performs the authentication response of the cloud service security access protocol using the mobile terminal T, he can generate the following cloud service request command in the SW to read the cloud file Fuser1 :

$$ID \| Enc_{k^{Enc}}(\text{"}read\text{"}, F_{user\ 1}, n_{i+1}) \| MAC_{k^{MAC}}(ID, Enc_{k^{Enc}}(\text{"}read\text{"}, F_{user\ 1}, n_{i+1})).$$

After C received the command, find the associated session key package and user name of user1 according to the ID. After parsing the request command and verifying the legitimacy of the command, C checks the user1's permission for the file Fuser1. If it is judged as readable, the following service response will be returned: where File (Fuser1) represents the file entity specified by Fuser1, the use of data block technology can achieve large volume of the file network encryption transmission. If user1 wants to share the file Fuser1 with user2, it can add the read permission of Fuser1 to user2. The corresponding cloud service request command is as follows:

$$ID \| Enc_{k^{Enc}}(\text{"}addright\text{"}, user2, \text{"}read\text{"}, F_{user1}, n_{i+1}) \| MAC_{k^{MAC}}(ID, Enc_{k^{Enc}}(\text{"}addright\text{"}, user2, \text{"}read\text{"}, F_{user1}, n_{i+1})).$$

After receiving the column verification, C adds a user2 readable entry in the permission list of file Fuser1 , but at this time the owner of Fuser1 is still user1, and user1 can send a command to cancel the read permission of user2 to Fuser1 . In addition, when the symbol * is used in the above request command instead of user2, user1 assigns the readable authority of Fuser1 to all legitimate users, that is, the public sharing of files is realized. The other functions of MCFile other cloud storage service functions can be implemented by this method. Analogy, no more description here.

6 Assessment

We simulated and realized the mobile terminal T, the application service provider A and the cloud service provider C respectively. For the simulation and realization of mobile terminal equipment, we used the embedded development board Zynq-7000 AP Soc Evaluation Kit. The board supports the TrustZone security extension with an ARM Cortex-A9 MPCore processor, 1GB of DDR3 memory, and OCM (on-chip memory) module with 256KB SRAM.

For the application service provider's simulation implementation, we used a Dell OptiPlex 990 desktop computer with a 3.3GHz Intel i3-2120 dual-core processor and 4GB of memory, running the Ubuntu10.04 operating system with kernel version Linux 2.6.32. For cloud service provider simulation implementation, we used a Lenovo ThinkCentre M8500t desktop computer, equipped with 3.4GHz Intel i7-4770 quad-core processor and 8GB of memory, the operating system is the same as the former.

6.1 Code Amount and Trusted Computing Based

In the program prototype system, the realization of the components of the C code's approximate lines number (lines of code, referred to as LoC) in Table 1. The trusted computing base (TCB) of a device is a collection of software, hardware, and firmware required to achieve device security. The smaller the scale is, the more difficult to be attacked by rivals, and the security is relatively easy to be guaranteed. In this scheme, the TCB of the trusted mobile terminal contains only the mobile device hardware and the software running in the SW. According to the literature [11], a certain type of SW security OS currently in the mobile commercial market has 6000 LoC. If this type of OS is used, plus the trusted service and App trustlet that we implement, the TCB software part of the scheme is only 9100 LoC. This scale is relatively small, and the controllability of system security is relatively high.

6.2 Performance Evaluation

Using the prototype system, we experimented the related operations required by the mobile terminal T to perform the solution in this paper. The program include encapsulation and unblocking sensitive data and communication interactions in the process of generating and resolving authorization requests, access requests, and requests for cloud services, among them, cloud service request and response messages do not consider specific cloud service commands. The operating time cost statistics take the average of 100 runs, and the experimental results are shown in Table 2.

6.3 Performance Evaluation of Server-side Program

Using the prototype system, we experimented with the related operations required by the application service provider A and the cloud service provider C in the implementation of this program. In the scheme, A is responsible for receiving and resolving the authorization application messages sent by the mobile terminal and verifying it to generate an authorization response message. In this experiment, we take this process as a response. First,

Table 1: Code size and TCB implemented components

Entity	Components	Loc	TCB
Mobile Terminal T	Trusted service	2300	V
	Trusted proxy	1500	X
	App trustlet	800	V
	App	500	-
Application Service Provider A	Authorization procedure	5600	-
Cloud Service Provider C	Access authentication procedure	6300	-

Table 2: Time overheads of the operations on mobile terminal

Operatations	Time Consumption (ms)
Encapsulation apk	0.030
Unblock apk	0.020
Encapsulation session key package	0.081
Unblock session key package	0.093
Generate m_apply	129.906
Resolve m reply	128.738
Generate m request	0.117
Resolve m response	0.110
Generate cloud service access request	0.152
Resolve Cloud Service Authentication Response	0.150

we experimented with the time cost of A single thread to complete a response, taking the average of 100 runs independently. The experimental results are single-threaded single time-consuming 13.225ms. Then, we experimented with the time required to complete a single response when using A thread pool concurrent execution with a large number of authorization requests. The experimental results are shown in Figure 3 . It can be seen from the graph that as the number of concurrent requests increases from 100 to 500 on the abscissa, the response time of a single request increases from about 400 ms to about 2300 ms on the ordinate. This substantial increase is due to the fact that A needs to be in a single response execute asymmetric encryption,decryption,signature and verify every time, these operations consume more system resources. Our experiments are based on a generic desktop computer, taking into account that A is usually implemented by several professional server clusters in practical application, and the optimized concurrency response will have a lot of room for improvement. In addition, the frequency of mobile users to implement authorization applications is not high compared to the mobile network delay. Moreover, server response delay about 2 000ms can not be accepted.

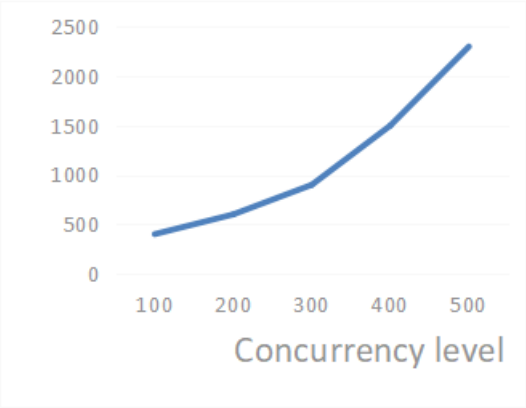


Figure 3: Authorization response latency in A

C is responsible for receiving and resolving the access request message sent by the mobile terminal in the scheme and validating it to generate a verification response message. In the experiment, we take this process as a response. Similarly,we first experimented with the execution of a single response of C single thread, and the ex-

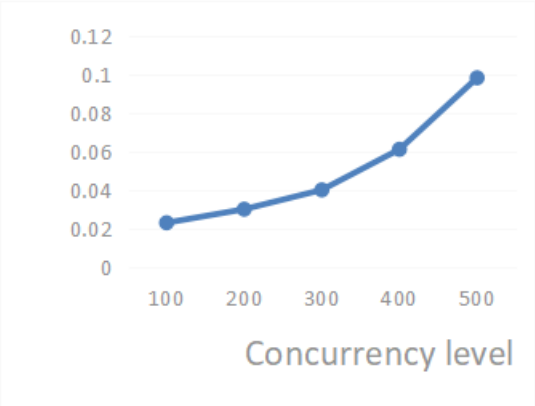


Figure 4: Authentication response latency in C

perimental results were 0.016 ms for single-thread single response. Then, we experimented with C in the case of concurrent execution to complete a single response time, the experimental results can be shown in Figure 4. It can be seen from the figure that as the number of concurrent requests increases, the response time of a single request increases from about 0.030ms to about 0.100ms, and the absolute value and the growth rate are not large, which is due to the operations that the C in the process of one response does not consume a lot of system resources. In addition to sending a verification response message to the mobile terminal, C will also interact with the mobile terminal in large numbers to complete the specific cloud service function response, which is basically consistent with the authentication response. Regardless of the specific cloud service operation, the response time overhead will be at the same level as the experimental results in Figure 4. The request response handled by C will be executed frequently in this scenario, which occupies a large proportion of the actual running interaction of the scheme. The low response delay reflected in the experiment indicates that the solution has good performance at the cloud service provider.

7 Discussion and Conclusions

7.1 Discussion

In many anonymous authentication systems, the direct anonymous attestation (DAA) protocol was officially released by TCG for anonymous proof based on TPM and has now been accepted as an ISO standard. The protocol can be modified to apply to the program in order to achieve anonymous authentication of the service provider to the user. In the previous work, we designed the DAA-TZ scheme for mobile terminals based on the DAA protocol. The scheme used TrustZone's good features to provide secure and efficient anonymous authentication services. The program system has been fully implemented and tested. Therefore, combining with DAA-TZ, it is possible to design and implement a trusted terminal cloud service secure access scheme with anonymous attributes.

7.2 Conclusion and Future Studies

This paper analyzes the related security issues of mobile terminal access cloud service for mobile cloud computing scene, and proposes a trusted mobile terminal cloud service security access scheme. The scheme uses TrustZone security extension technology to construct trusted mobile terminal architecture. Trusted mobile terminal uses SRAM PUF to obtain root key seed, and realized the security management mechanism of key and sensitive data.Secondly, based on the idea of trusted computing technology, the cloud service security access protocol is designed on the basis of trusted mobile terminal, and the protocol is compatible with trusted cloud computing architecture. The analysis and experimental results show that the security access scheme proposed in this paper can effectively realize the security authentication of the mobile terminal in the process of accessing the cloud service and protect the private data security of the mobile user in the cloud service. The program has better scalability and smaller mobile terminal TCB, its overall operation efficiency is higher, mobile users wait for the delay within the acceptable range. In the future work,we will do a formal analysis for the security access protocol which presented in the program , and give a more detailed proof of security.

Acknowledgments

This work is supported by Scientific Study Project for Institutes of Higher Learning, Ministry of Education, Liaoning Province (LQN201720), and Natural Science Foundation of LaioNing Province, China (20170540819). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] A. Alaqra, S. Fischer-Hübner, T. Groß, *et al.*, "Signatures for privacy, trust and accountability in the cloud: Applications and requirements," *Privacy and Identity Management. Time for a Revolution?*, vol. 476, pp. 79-96, 2016.

[2] S. Al-Janabi, I. Al-Shourbaji, M. Shojafar, *et al.*, "Mobile cloud computing: Challenges and future research directions," in *International Conference on Developments in Esystems Engineering*, 2017. DOI: 10.1109/DeSE.2017.21.

[3] A. Alzahrani, N. Alalwan, M. Sarrab, "Mobile cloud computing: Advantage, disadvantage and open challenge," in *Proceedings of the 7th Euro American Conference on Telematics and Information Systems*, 2014. DOI:10.1145/2590651.2590670.

[4] H. T. Dinh, C. Lee, D. Niyato, P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications & Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[5] GlobalPlatform device technology, *TEE Client API Specification Version 1.0*, 2010. (http://globalplatform.org)

[6] S. Hosseinzadeh, S. Laurén, V. Leppänen, "Security in container-based virtualization through vTPM," in *IEEE/ACM International Conference on Utility & Cloud Computing*, 2017. DOI: 10.1145/2996890.3009903.

[7] Q. Jiang, J. Ma, F. Wei, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1-4, 2016.

[8] M. John, R. Tom, S. Fred, *The Cloud-Proxy Tao for Trusted Vomputing*, Technical Report, No.UCB/EECS-2013-135, 2013. (http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-135.html)

[9] C. Lee, *Security Control Apparatus and Method for Cloud-based Virtual Desktop*, 2017. (https://patents.google.com/patent/US9674143B2/en)

[10] Y. Li, Z. Han, Z. Huang, *et al.*, "A remotely keyed file encryption scheme under mobile cloud computing," *Journal of Network & Computer Applications*, vol. 106, pp. 90-99, 2018.

[11] W. H. Li, H. B. Li, H. B. Chen, Y. B. Xia, "AdAttester: Secure online mobile advertisement attestation using TrustZone," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 75–88, 2015.

[12] J. Lind, I. Eyal, F. Kelbert, *et al.*, "Teechain: Scalable blockchain payments using trusted execution environments," *ArXiv*, 2017. (https://www.researchgate.net/publication/318528079_Teechain_Scalable_Blockchain_Payments_using_Trusted_Execution_Environments)

[13] M. B. Mollah, M. A. K. Azad, A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way ahead," *Journal of Network and Computer Applications*, vol. 84, pp. 34-54, 2017.

[14] W. Pan, Y. Zhang, M. Yu, *et al.*, "Improving virtualization security by splitting hypervisor into smaller components," in *Data and Applications Security and Privacy XXVI*, pp. 298-313, 2012.

[15] N. Santos, K. P. Gummadi, R. Rodrigues, "Towards trusted cloud computing," *Proceedings of the Conference on Hot Topics in Cloud Computing*, 2009. (https://www.usenix.org/legacy/event/hotcloud09/tech/full_papers/santos.pdf)

[16] Trusted computing group, *TPM Main Specification, Version1.2, Revision 116*, 2011. (http://www.trustedcomputinggroup.org)

[17] Trusted computing group, *Trusted Platform Module Library, Family 2.0, Revision 01.16*, 2014. (http://www.trustedcomputinggroup.org)

[18] Trusted computing group, *TCG Mobile Trusted Module Specification, Version1.0, Revision 7.02*, 2010. (http://www.trustedcomputinggroup.org)

[19] Q. X. Wu, X. W. Yang, H. Zou, F. J. Yu, X. K. Ning, Z. Wang, "Technic specification of cryptography supporting platform for trusted computing," *China State Password Administration Committee*, 2007. (http://www.oscca.gov.cn)

[20] C. Wu, Y. J. Zhou, K. Patel, Z. K. Liang, X. X. Jiang, "AirBag: Boosting smartphone resistance to malware infection," in *Proceedings of Network and Distributed System Security Symp (NDSS'14)*, 2014. (https://pdfs.semanticscholar.org/4823/f6af261a88716980485638f2d06f94bbf2d4.pdf)

[21] B. Yang, D. G. Feng, Y. Qin, "A lightweight anonymous mobile shopping scheme based on DAA for trusted mobile platform," in *Proceedings of the IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 9–17, 2014.

[22] Y. J. Zhang, D. G. Feng, Y. Qin, B. Yang, "A TrustZone based trusted code execution with strong security requirements," *Journal of Computer Research and Development*, vol. 52, no. 10, pp. 2224–2238, 2015.

[23] B. Zhao, F. Yan, L. Q. Zhang, J. Wang, "Build trusted cloud computing environment," *Communications of China Computer Federation (CCF'12)*, vol. 8, no. 7, pp. 28–34, 2012.

[24] S. J. Zhao, Q. Y. Zhang, G. Y. Hu, Y. Qin, D. G. Feng, "Providing root of trust for ARM TrustZone using on-chip SRAM," in *Proceedings of the 4th International Workshop on Trustworthy Embedded Devices*, pp. 25-36, 2014.

Biography

Hui Xia received his B.S. and M.S. degree from Xidian University,China, in 2003, 2006,respectively. He is currently a associate professor with Shenyang Normal University. His research interests include cloud computing, cryptography and information security.

WeiJi Yang received his B.S. degree from Zhejiang Chinese Medical University , China, in 2005, M.S. degrees from Beijing University of Posts and Telecommunications, China, in 2009. He is currently a Research Associate with ZheJiang Chinese Medical University. His research interests include Information Security and Traditional Chinese Medicine Informatics.

Security Analyses of a Data Collaboration Scheme with Hierarchical Attribute-based Encryption in Cloud Computing

Wei-Liang Tai[1], Ya-Fen Chang[2], and Wen-Hsin Huang[2]
(Corresponding author: Ya-Fen Chang)
Department of Information Communications, Chinese Culture University[1]
No. 55, Hwa-Kang Road, Taipei, Taiwan
Department of Computer Science and Information Engineering, National Taichung University of Science and Technology[2]
No. 129, Section 3, Sanmin Road, Taichung, Taiwan
(Email: cyf@nutc.edu.tw)
(Received Oct. 16, 2018; Revised and Accepted Feb. 7, 2019; First Online June 25, 2019)

Abstract

With the prevalence of cloud computing, users store and share confidential data in the cloud while this approach makes data security become an important and tough issue. To ensure data security, cloud service providers must provide efficient and feasible mechanisms to provide a reliable encryption method and a suitable access control system. In order to realize this ideal, Huang *et al.* proposed a data collaboration scheme with hierarchical attribute-based encryption. After analyzing Huang *et al.*'s scheme, we find that one weakness exists in their scheme such that the semi-trusted cloud service provider can decrypt the protected data to obtain the plaintext. Data confidentiality is not ensured as claimed. In this paper, we will explicitly indicate how this weakness damages Huang *et al.*'s scheme.

Keywords: Cloud Computing; Data Confidentiality; Data Collaboration; Hierarchical Attribute-based Encryption

1 Introduction

With rapid progress of network technologies, plenty of various applications and services are proposed and realized, and cloud computing revolutionizes the way how services are provided. Cloud computing possesses superior properties to benefit users such that resources including storage can be easily accessed, shared, and virtualized. Moreover, distributed computing is also allowed in cloud computing.

In addition to the above advantages, cloud computing can help users to save time and money because they do not need to construct the infrastructure by themselves completely. Cloud computing ensures flexibility. For example, users can obtain the required resources or services provided in the cloud and keep essential data secretly and locally. The flexible property makes more and more enterprises utilize cloud-based services.

Although cloud computing brings great benefits to enterprises and cloud users, many security issues are raised. Data confidentiality and access control in cloud computing are serious and urgent. It is because the cloud service provider (CSP) is semi-trusted and the data stored in the cloud may be disclosed by an unauthorized user or a malicious employee in CSP. This denotes that data leakage will take place if these security issues are not well and appropriately addressed [2]. As a result, data confidentiality and access control are important issues in cloud computing.

The reliable approach to protect data is encrypting data before being outsourced. The traditional methods for data encryption include symmetric encryption and asymmetric encryption. However, the above two traditional encryption methods are not suitable for data access control in cloud systems. As a result, attribute-based encryption (ABE) is proposed to ensure data access control with high precision [10]. An ABE mechanism enables access control over encrypted data with access policies and attributes among private keys and ciphertexts. Moreover, ciphertext-policy attribute-based encryption (CP-ABE) makes the data owner define access policies on all attributes that users need to decrypt the ciphertext. By CP-ABE, data confidentiality and data access control can be guaranteed [3].

However, the previous methods are designed to provide users with secure data reading while how multiple users collaboratively manipulate encrypted data in cloud computing is not taken into consideration.

Data collaboration service offered by CSP supports availability and consistency of the data shared among users [1]. In short, cloud computing is providing most

of the functions originally provided by computers via the Internet. A user only needs one terminal to complete all functions such as the website setup, program development, and file storage. In order to realize and provide secure data collaboration services in cloud computing, only authorized users have the right to access or modify data in the cloud. That is, CSP needs to verify the user′s legitimacy. A cryptographic technique, Attribute-Based Signature (ABS), can help CSP verify the user when he/she requests to modify the data stored in the cloud. In an ABS system, the user can sign messages with his/her attributes key. Then, from the signature, CSP can check whether the signer′s attributes meet the access policy while the signer′s identity is unknown.

In recent years, many researches about the topics have been proposed. In 2011, Hur *et al.* proposed an attribute-based access control scheme in data outsourcing systems [5]. In 2012, Wan *et al.* proposed a hierarchical attribute-based access control in cloud computing scheme [8]. However, the above two schemes only take data sharing into consideration and cannot support write operations over stored data. In 2013, Li *et al.* proposed a secure sharing scheme based on attribute-based encryption for personal health records in cloud computing [6]. Li *et al.*′s scheme allows write operations. Unfortunately, the cloud still cannot verify the user′s write permission after receiving the re-encrypted modified data. In 2015, Yang *et al.* proposed one outsourcing scheme for big data access control in cloud and claimed that it cloud ensure security and verifiability [9]. Unfortunately, Liu *et al.* show that Yang *et al.*′s outsourcing scheme for big data access control in cloud suffers from some security flaws [7].

In 2017, Huang *et al.* proposed a data collaboration scheme with hierarchical attribute-based encryption in cloud computing [4]. Huang *et al.*′s scheme applies ABE, attribute-based signature (ABS), and bilinear map to ensuring data confidentiality and data access control. In their system model, there are five entities, central authority, domain authority, CSP, data owner, and user. The central authority, a trusted third party, manages domain authorities, sets up system parameters, and issues the secret parameter to the domain authority at the top level. A domain authority is a trusted third party, manages multiple domain authorities and domain users, and generates the master key for each domain authority at the next level and attribute secret keys for users. CSP, a semi-trusted party, provides data storage and collaboration service, offers partial decryption and partial signing, and is responsible for verifying the re-encrypted data before accepting it. The data owner outsources the encrypted data to CSP for collaboration. A user possessing a set of attributes satisfying the access policy can access and modify the data in cloud computing. Huang *et al.* also claimed that their scheme ensured data confidentiality. After analyzing Huang *et al.*′s scheme, we find that the semi-trusted cloud service provider can decrypt the protected data and obtain the plaintext after an authorized user modifies the data. That is, Huang *et al.*′s scheme cannot provide data confidentiality as claimed.

The rest of this paper is organized as follows. Section 2 reviews Huang *et al.*′s data collaboration scheme with hierarchical attribute-based encryption in cloud computing. Analyses on Huang *et al.*′s scheme are given in Section 3. At last, some conclusions are drawn in Section 4.

2 Review of Huang *et al.*′s Scheme

Huang *et al.*′s scheme is composed of six phases:

1) System setup phase;
2) Domain setup phase;
3) Key generation phase;
4) Data encryption phase;
5) Data decryption phase;
6) Data modification phase.

In this section, we first introduce the symbols and nine algorithms used in Huang *et al.*′s scheme. Then we review Huang *et al.*′s scheme. The details are as follows.

2.1 Notations

Notations used in Huang *et al.*′s scheme are listed in Table 1.

Table 1: Notations used in Huang *et al.*′s scheme

Symbol	Definition
CSP	Cloud service provider
PK	Central authority′s public key
MK	Entity′s master key
S	A set of attributes
SK	User′s attribute secret keys
AK	User′s attribute key
GK	Global key
T	Access policy
DK	Data encryption key
CT	Ciphertext
ST	Signature
Enc/Dec	Symmetric encryption/decryption

2.2 Algorithms

Huang *et al.* proposed nine algorithms and used them to define the designed system. The definitions of these nine algorithms are shown as follows.

1) $Setup(K)$. The central authority takes a security parameter K as input and outputs the system public key PK and the central authority′s master secret key MK_0.

2) *CreateDM*(PK, MK_l, S). The central authority or a domain authority takes PK, the master key MK_l and a set of attributes S as inputs and outputs the master secret key MK_{l+1} for the domain authority at the next level.

3) *KeyGen*(PK, MK_l, S). A domain authority takes PK, MK_l and S as inputs and outputs the attribute secret keys SK for each domain user.

4) *Encrypt*(PK, M, T). The data owner takes PK, a message M and an access policy T as inputs and outputs the ciphertext CT.

5) *PartDec*(CT, AK). A user uses SK to generate the attribute key AK and sends AK to CSP. CSP takes CT and AK as inputs. If the attributes in AK satisfy T in CT, CSP outputs a partial decrypted ciphertext CT_P.

6) *Decrypt*(CT_P, SK). A user takes CT_P and SK as inputs, recovers the data encryption key DK, and outputs the plaintext M.

7) *PartSign*(Q, AK). CSP takes a data collaboration request Q and AK as inputs and outputs a partial signature ST_P and a global key GK.

8) *Sign*(ST_P, SK). A user takes ST_P and SK as inputs and outputs the signature ST.

9) *Verify*(T, ST, GK). CSP takes T, ST and GK as inputs. If ST is the user′s valid signature such that S satisfies T, it outputs true.

2.3 System Setup Phase

In the beginning, the central authority executes *Setup* algorithm as follows:

Step 1. Selects a bilinear group G_1 of prime order p and generator g and the bilinear map $\hat{e} : G_1 \times G_1 \to G_2$.

Step 2. Selects random numbers α and β in Z_p and defines hash functions $H_1, H_2 : \{0,1\}^* \to G_1$.

Step 3. Sets the master key $MK_0 = (\alpha, \beta)$ that is kept secret by the central authority and obtains the system public key PK, where $PK = (g^\alpha, g^\beta)$.

2.4 Domain Setup Phase

The central authority or a domain authority will be involved in this phase to execute $CreateDM$ algorithm. For clarity, two cases are given.

Case 1: The central authority executes $CreateDM$ algorithm as follows:

Step 1. Selects a unique number δ_l and chooses $\delta_{l,i} \in Z_p$ randomly for each attribute in A for $i = 1, 2, \ldots, m$, where A is a set of m attributes and $A = \{a1, a2, ..., a_m\}$.

Step 2. Computes $MK_l = (A, \overline{D_l} = g^{(\alpha+\delta_l)\beta}, \{\overline{D}_{l,i} = g^{\delta_l \beta} H_1(i)^{\delta_{l,i}}, \overline{D}'_{l,i} = g^{\delta_{l,i}} | a_i \in A,\ i \in \{1, 2, ..., m\}\})$ for the domain authority at the top level.

Case 2: The high level domain authority with MK_l executes $CreateDM$ algorithm as follows:

Step 1. Selects a unique number ε_l and chooses $\varepsilon_{l,i} \in Z_p$ randomly for each attribute in A' for $i = 1, 2, ..., n$, where A' is a set of n attributes $A' = \{a_1, a_2, ..., a_n\}$.

Step 2. Computes $MK_{l+1} = (A', \overline{D}_{l+1} = \overline{D}_l \cdot g^{\varepsilon_l \beta}, \{\overline{D}_{l+1,i} = \overline{D}_{l,i} \cdot g^{\varepsilon_l \beta} H_1(i)^{\varepsilon_{l,i}}, \overline{D}'_{l+1,i} = \overline{D}'_{l,i} \cdot g^{\varepsilon_{l,i}} | a_i \in A', i \in \{1, 2, ..., n\}\})$ for the domain authority at the next level.

2.5 Key Generation Phase

When a user joins in the domain, the corresponding domain authority with MK_l executes $KeyGen$ algorithm as follows:

Step 1. Selects $\gamma \in Z_p$ randomly for the user and chooses $\gamma_i \in Z_p$ randomly for each a_i in S, where S is a set of the user′s attributes.

Step 2. Computes the attribute secret keys $SK = (S, D = \overline{D}_l \cdot (g^\beta)^\gamma, \{D_i = \overline{D}_{l,i} \cdot g^{\gamma\beta} H_1(i)^{\gamma_i}, D'_i = \overline{D}'_{l,i} \cdot g^{\gamma_i} | i \in S\})$ for the user, where $i \in S$ is the shorthand for $a_i \in S$.

2.6 Data Encryption Phase

The data owner executes *Encrypt* algorithm to encrypt the data M, defines the access policy T, and outsources the ciphertext to the cloud. The data owner performs as follows:

Step 1. Selects a random number $DK \in Z_p$ to encrypt the data M by using a symmetric encryption algorithm. Note that M will be encrypted under the access policy T.

Step 2. Generates a polynomial p_x for each node x in the access tree T with a top-down manner starting from the root node R. Sets the degree d_x of p_x to be $k_x - 1$ for each node x in T, where k_x is the threshold value of x and $k_x = 1$ if x a leaf node. On the root node R, chooses a random number $s \in Z_p$, sets $p_R(0) = s$, and chooses other d_R nodes randomly to define p_R. For other node x, sets $p_x(0) = p_{parent(x)}(index(x))$ and chooses other d_x nodes randomly to define p_x, where $index(x)$ is the label associated with x and $index(x)$ will be from 1 to $num(p)$ when x is the child node of node p and $num(p)$ denotes the number of p′s child nodes.

Step 3. Computes the ciphertext $CT = (T, E = Enc_{DK}(M), \tilde{C} = DK \cdot \hat{e}(g,g)^{\alpha\beta_s}, C = g^s, \{C_y = g^{p_y(0)}, C'_y = H_1(attr_y)^{p_y(0)}\}_{y \in Y})$ and outsources CT to CSP, where Y is a set of leaf nodes in access policy T.

2.7 Data Decryption Phase

This phase is composed of two parts, partial decryption phase and decryption phase. The details are as follows.

2.7.1 Partial Decryption Phase

When a user wants to access the data owner′s outsourced ciphertext from CSP, he/she first generates the attribute key $AK = \{D_i, D_i' | i \in S\}$ to CSP.

After getting AK, CSP executes $PartDec$ algorithm to partially decrypt the ciphertext. Then CSP executes a recursive algorithm, $DecryptNode$ algorithm. The recursive algorithm $DecryptNode(CT, AK, p)$ takes the ciphertext CT, the attribute key AK associated with S, and a node p from T as inputs.

If the node p is a leaf node y of T, $i = attr_y$, where $attr_y$ denotes an attribute associated with the leaf node y. If $i \in S$, $DecryptNode(CT, AK, p) = DecryptNode(CT, AK, y) = \frac{\hat{e}(D_i, C_y)}{\hat{e}(D_i', C_y')}$; otherwise, if $i \notin S$, $DecryptNode(CT, AK, y) = \perp$.

If the node p is a non-leaf node x of T, $DecryptNode(CT, AK, x)$ is executed by calling $DecryptNode(CT, AK, z)$ for all child nodes z of x and storing the output F_z. If no S_x, an arbitrary k_x-sized set of child nodes z of x such that $F_z \neq \perp$, exists, the node does not meet T and $DecryptNode(CT, AK, x) = \perp$. Otherwise, it denotes the subtree rooted at node x meets the access policy T if and only if k_x subtrees rooted at x′s children meet T. F_x is computed as follows, where $parent(z)$ is a parent node of z, $index(z)$ is the label associated with z, and $\Delta_{r,S_x}(x) = \prod_{j \in S_x, j \neq r} \frac{x-j}{r-j}$. Because z is a child node of x, $index(z)$ will be from 1 to $num(x)$, where $num(x)$ is the number of x′s children.

$$F_x = \prod_{Z \in S_x} F_Z^{\Delta_{j,S_x'(0)}} = \hat{e}(g,g)^{(\delta_l + \gamma)\beta p_x(0)}$$

where $S_x' = \{index(z) | z \in S_x\}$ and $j = index(z)$. With the recursive approach, calling $DecryptNode(CT, AK, R)$ can have the masking factor W efficiently obtained to decrypt CT such that $W = DecryptNode(CT, AK, R) = \hat{e}(g,g)^{(\delta_l + \gamma)\beta_s}$. Then, CSP sends the partial decrypted ciphertext $CT_P = (E, \tilde{C}, C, W)$ to the user.

2.7.2 Decryption Phase

After receiving CT_P, the user executes $Decrypt$ algorithm to retrieve the plaintext. The user first computes $DK = \tilde{C}/\hat{e}(C, D)/W)$ and obtains DK. Then the user can use DK to retrieve $M = Dec_{DK}(E)$.

2.8 Data Modification Phase

When a user needs to modify the stored data in the cloud to work collaboratively, he/she must use his/her attributes to sign the data collaboration request by using attribute-based signature, ABS. Only the user's signature satisfying the access policy can be authorized to outsource the re-encrypted data. Data modification phase is composed of four parts, writing data phase, partial signing phase, signing phase and verification phase. The details are as follows.

2.8.1 Writing Data Phase

The collaborative user obtains the plaintext in data decryption phase. After the user modifies the data, he/she re-encrypts data with T. Then the user sends the data collaboration request Q, AK and the re-encrypted data to CSP.

2.8.2 Partial Signing Phase

After receiving the collaboration request, CSP executes $PartSign$ algorithm. CSP selects a random number $\mu \in Z_p$ and computes $\tilde{S}_0 = H_2(Q)^\mu$ and $S_0 = g^\mu$. CSP generates a polynomial q_x for each node x in the access tree T with a top-down manner starting from the root node R. CSP sets the degree b_x of q_x to be $k_x - 1$ for each node x in T, where k_x is the threshold value of x. On the root node R, CSP chooses a random number $t \in Z_p$, sets $q_R(0) = t$, and chooses other b_R nodes randomly to define q_R. For other node x, CSP sets $q_x(0) = q_{parent(x)}(index(x))$ and chooses other b_x nodes randomly to define q_x. CSP computes the global key $GK = \{K_y = g^{q_y(0)}, K_y' = H_1(attr_y)^{q_y(0)} | y \in Y\}$ for each $y \in Y$, where Y is a set of leaf nodes in access policy T.

CSP selects a random number $t_i \in Z_p$ for each $i \in Y$ and uses AK to compute $\{S_i, S_i'\}$, where $\{S_i = D_i H_1(i)^{t_i}, S_i' = D_i' g^{t_i} | i \in S \cap Y\}$ and $\{S_i = H_1(i)^{t_i}, S_i' = g^{t_i} | i \in Y/S \cap Y\}$. Then CSP generates the partial signature $ST_P = (\tilde{S}_0, S_0, \{S_i, S_i' | i \in Y\})$ and sends ST_P to the user.

2.8.3 Signing Phase

After receiving ST_P, the user executes $Sign$ algorithm to generate the signature. The user computes $\tilde{S} = \tilde{S}_0 D$ and $S = S_0$ and generates the signature $ST = (\tilde{S}, S, \{S_i, Si' | i \in Y\})$. Then the user sends ST to CSP.

2.8.4 Verification Phase

After receiving ST, CSP executes $Verify$ algorithm to verify the signature ST. CSP executes $VerifyNode$ algorithm that is a recursive algorithm. The recursive algorithm $VerifyNode$ takes ST, a node p from T and GK associated with a set of attributes as inputs.

If the node p is a leaf node y of T, $i = attr_y$. If $i \in S \cap Y$, $VerifyNode(ST, GK, p) = VerifyNode(ST, GK, y) = \frac{\hat{e}(S_i, K_y)}{\hat{e}(S_i', K_y')}$. If $i \in Y/S \cap Y$, $VerifyNode(ST, GK, p) = VerifyNode(ST, GK, y) = \frac{\hat{e}(S_i, K_y)}{\hat{e}(S_i', K_y')} = 1$.

If the node p is a non-leaf node x, $VerifyNode(ST, GK, x)$ is executed by calling $VerifyNode(ST, GK, z)$ for all child nodes z of x

and storing the output G_z. If no G_z, an arbitrary k_x-sized set of child nodes z of x such that $G_z \neq \perp$, exists, the node does not meet T and $VerifyNode(ST, GK, x) = \perp$. Otherwise, it denotes the subtree rooted at node x meets the access policy T if and only if k_x subtrees rooted at x's children meet T. G_x is computed as follows, where $parent(z)$ is a parent node of z, $index(z)$ is the label associated with z, and $\Delta_{r,S_x}(x) = \prod_{j \in S_x, j \neq r} \frac{x-j}{r-j}$. Because z is a child node of x, $index(z)$ will be from 1 to $num(x)$, where $num(x)$ is the number of x's children.

$$G_x = \prod_{Z \in S_x} G_Z^{\Delta_{i,S_x'(0)}} = \hat{e}(g,g)^{(\delta_l+\gamma)\beta q_x(0)}$$

where $S_x' = \{index(z) | z \in S_x\}$ and $i = index(z)$. With the recursive approach, calling $VerifyNode(ST, GK, R)$ can have the masking factor I efficiently obtained to verify the signature such that $I = VerifyNode(ST, GK, R) = \hat{e}(g,g)^{(\delta_l+\gamma)\beta t}$. Then, CSP checks if $\frac{\hat{e}(g,\tilde{S})}{\hat{e}(H_2(Q),S)\cdot(I)^{1/t}}$ equals $\hat{e}(g,g)^{\alpha\beta}$. If they are equal, CSP accepts the signature and the re-encrypted data form the collaborative user. Otherwise, CSP rejects this data collaboration request.

3 Analysis on Huang *et al.*'s Scheme

After analyzing Huang *et al.*'s scheme, we find that their scheme cannot provide data confidentiality as claimed. Because the cloud service provider CSP is semi-trusted in Huang *et al.*'s scheme, CSP should neither know nor retrieve what the original data is even when users use data collaboration service. In Huang *et al.*'s scheme, the data M is protected by being encrypted by the data encryption key DK, and only users who meet the access policy can work collaboratively. Because it is only mentioned that the user modifies the data and re-encrypts data with T in writing data phase of data modification phase, this makes two cases possible. First, the modified data is re-encrypted with the same DK. Second, the modified data is re-encrypted with new DK by executing data encryption phase. No matter which case is true, CSP can obtain DK to retrieve data. For clarity, the details are given in the following.

3.1 Re-encrypting Data with The Same DK

Suppose the modified data is re-encrypted with the same DK in writing data phase. In partial decryption phase of data decryption phase, when a user wants to access the data owner′s outsourced ciphertext from CSP, he/she first generates the attribute key $AK = \{D_i, D_i' | i \in S\}$ to CSP. After getting AK, CSP executes $PartDec$ algorithm to partially decrypt the ciphertext with a recursive algorithm, $DecryptNode$ algorithm. Then, CSP sends the partial decrypted ciphertext $CT_P = (E, \tilde{C}, C, W)$ to the user, where $W = DecryptNode(CT, AK, R) = \hat{e}(g,g)^{(\delta_l+\gamma)\beta_S}$. In decryption phase of data decryption phase, after receiving CT_P, the user executes $Decrypt$ algorithm to retrieve the plaintext by computing $DK = \tilde{C}/(\hat{e}(C,D)/W)$ and $M = Dec_D K(E)$. The above denotes that CSP is aware of $(E, \tilde{C}, C, W)$ after an authorized user accesses the data owner′s outsourced ciphertext from CSP.

Suppose that U_1, who is an authorized user and has accessed the outsourced ciphertext from CSP, wants to modify the data. That is, data modification phase will be executed. In partial signing phase, CSP executes $PartSign$ algorithm by generating the partial signature $ST_P = (\tilde{S}_0, S_0, \{S_i, S_i' | i \in Y\})$ and sending ST_P to the user, where $\tilde{S}_0 = H_2(Q)^\mu$ and $S_0 = g^\mu$. In signing phase of data modification phase, after receiving ST_P, the user executes $Sign$ algorithm to generate the signature by computing $\tilde{S}_0 = \tilde{S}_0 D$ and $S = S_0$ and generating the signature $ST = (\tilde{S}, S, \{S_i, S_i' | i \in Y\})$. In verification phase, after receiving ST, CSP executes $Verify$ algorithm to verify the signature ST. The above denotes that CSP is aware of $(\tilde{S}_0, S_0, \tilde{S}, S, \{S_i, S_i' | i \in Y\})$ after an authorized user wants to modify the data.

From then on, CSP knows $(E, \tilde{C}, C, W)$ and $(\tilde{S}_0, S_0, \tilde{S}, S, \{S_i, S_i' | i \in Y\})$. To retrieve the data, CSP performs as follows:

Step 1. Computes $\tilde{S} \times (\tilde{S}_0)^{-1} = (\tilde{S}_0 D) \times (\tilde{S}_0)^{-1} = D$.

Step 2. Computes $DK = \tilde{C}/(\tilde{e}(C,D)/W)$.

Step 3. Computes $M = Dec_{DK}(E)$.

According to the above, it is obvious that CSP can retrieve the original data after an authorized user modifies the data. This found weakness shows that Huang *et al.*'s scheme cannot ensure data confidentiality.

3.2 Re-encrypting Data with New DK by Executing Data Encryption Phase

Suppose the modified data is re-encrypted with new DK by executing data encryption phase in writing data phase. If a user U_1 has ever modified the data, CSP can obtain D after signing phase is executed. When another authorized user U_2 wants to access the re-encrypted data, CSP can get DK with D to retrieve the data M. The details are as follows:

Step 1. In signing phase, U_1 receives $ST_P = (\tilde{S}_0, S_0, \{S_i, S_i' | i \in Y\})$ from CSP. Then U_1 computes $\tilde{S}_0 = \tilde{S}_0 D$ and $S = S_0$ and sends $ST = (\tilde{S}, S, \{S_i, S_i' | i \in Y\})$ to CSP. Because CSP is aware of $(\tilde{S}_0, S_0, \tilde{S}, S, \{S_i, S_i' | i \in Y\})$, CSP can retrieve D by computing $\tilde{S} \times (\tilde{S}_0)^{-1} = D$.

Step 2. When U_2 accesses the re-encrypted data, U_2 receives $CT_P = (E, \tilde{C}, C, W)$ from CSP in decryption

phase of data decryption phase. U_2 uses parameters $(\tilde{C}, C, W, D)$ to compute $DK = \tilde{C}/(\tilde{e}(C, D)/W)$. Then U_2 can retrieve the data M with DK. It means that CSP is also capable of computing the data encryption key DK because $\tilde{C}, C, W$, and D are all known. Thereupon, CSP can also retrieve the data M with DK.

According to the above, even if the modified data is re-encrypted with new DK by executing data encryption phase, CSP still can decrypt the encrypted data to retrieve the plaintext after another authorized user accesses the re-encrypted data.

4 Conclusions

Huang *et al.* proposed a hierarchical attribute-based encryption scheme to realize data collaboration in cloud computing. In this paper, we explicitly show how Huang *et al.*'s scheme suffers from one weakness. The data M is protected by the key DK, and it is supposed that only users who meet the access policy could obtain the plaintext. However, we find that CSP can retrieve the outsourced data after an authorized user modifies the data because CSP can get the data encryption key DK. Because CSP is semi-trusted, CSP should never know what the data M is. As a result, data confidentiality cannot be ensured in Huang *et al.*'s scheme. According to our findings, how to design a secure and efficient data collaboration scheme in cloud computing is still an urgent and tough issue.

Acknowledgments

This work was supported in part by Ministry of Science and Technology under the Grants MOST 106-2221-E-034-006-, MOST 106-2410-H-025-006-, MOST 106-2622-H-025-001-CC3, and MOST 107-2622-H-025-001-CC3.

References

[1] X. Dong, J. Yu, Y. Luo, Y. Chen, G. Xue and M. Li, "Achieving secure and efficient data collaboration in cloud computing," in *Proceedings of IEEE/ACM 21st International Symposium on Quality of Service (IWQoS'13)*, pp. 195–200, June 2013.

[2] Q. Huang, Z. Ma, Y. Yang, J. Fu and X. Niu, "Secure data sharing and retrieval using attribute-based encryption in cloud-based osns," *Chinese Journal of Electronics*, vol. 23, no. 3, pp. 557–563, 2014.

[3] Q. Huang, Z. Ma, Y. Yang, J. Fu and X. Niu, "Eabds: attribute-based secure data sharing with efficient revocation in cloud computing," *Chinese Journal of Electronics*, vol. 24, no. 4, pp. 862–868, 2015.

[4] Q. Huang, Y. Yang and M. Shen, "Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing," *Future Generation Computer Systems*, vol. 72, pp. 239–249, 2017.

[5] J. Hur and D.K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[6] M. Li, S. Yu, Y. Zheng, K. Ren and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2013.

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

[8] Z. Wan, J. Liu and R. H. Deng, "Hasbe: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743–754, 2012.

[9] K. Yang, X. H. Jia and K. Ren, "Secure and verifiable policy update outsourcing for big data access control in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3461–3470, 2015.

[10] S. Yu, C. Wang, K. Ren and W. Lou, "Achieving secure, scalable and fine-grained data access control in cloud computing," in *Proceedings of IEEE International Conference on Computer Communications (IEEE INFOCOM'10)*, pp. 1–9, Mar. 2010.

Biography

Wei-Liang Tai received the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Taiwan, in 2008. He is currently Associate Professor, Department of Information Communications, Chinese Culture University. His main interests are in information security and forensics and multimedia signal processing. He is currently an Editor of KSII Transactions on Internet and Information Systems.

Ya-Fen Chang is a Professor of Department of Computer Science and Information Engineering at National Taichung University of Science and Technology in Taiwan. She received her BSc degree in computer science and information engineering from National Chiao Tung University and PhD degree in computer science and information engineering from National Chung Cheng University, Taiwan. Her current research interests include electronic commerce, information security, cryptography, mobile communications, image processing, and data hiding.

Wen-Hsin Huang received the MS degree in computer science and information engineering from National Taichung University of Science and Technology in Taiwan in 2018. Her main interests are in electronic commerce and information security.

An Intrusion Detection Model for Wireless Sensor Network Based on Information Gain Ratio and Bagging Algorithm

Rui-Hong Dong, Hou-Hua Yan, and Qiu-Yu Zhang
(*Corresponding author:Qiu-Yu Zhang*)

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

(Received Aug. 5, 2018; Revised and Accepted Mar. 23, 2019; First Online June 11, 2019)

Abstract

Aiming at the problem that the dimension of the traffic data to be processed in the wireless sensor network (WSN) intrusion detection method is too high, which leads to the large amounts of computational complexity of the intrusion detection model and the weak detection performance of the intrusion behavior. Using the principle of ensemble learning algorithm, an intrusion detection model for WSN based on information gain ratio and Bagging algorithm was proposed. Firstly, the information gain ratio method is used to select the feature of sensor node traffic data in this model. Secondly, the Bagging algorithm is used to construct an ensemble classifier so as to train multiple C4.5 decision trees which are improved. The parameters of the ensemble classifier are optimized through 10 iterations, and the dynamic pruning process is introduced. Finally, the classification results of C4.5 decision tree are classified and detected by majority voting mechanism. The experimental results show that compared with the existing intrusion detection methods, the proposed model has higher detection accuracy for Blackhole, Grayhole, Flooding, Scheduling and other intrusion attacks. While ensuring the true positive rate of 99.4%, it can still maintain a low false positive rate and high detection performance for intrusions behavior.

Keywords: Bagging Algorithm; Ensemble Classifier; Intrusion Detection; Information Gain Ratio; Wireless Sensor Network (WSN)

1 Introduction

With the wide application of WSN in smart cities, smart grids, environmental monitoring, medical sensing, industrial and other fields [12], it also brings some security issues such as network attacks and intrusions. Due to the wireless transmission and unattended characteristics of WSN, the sensor node has limited energy, storage capacity and computing power, which makes it vulnerable to various malicious attacks, such as Wormhole, Sinkholes, Greyhole, and Flooding and so on. These typical attacks all cause the network traffic to deviate from the normal network traffic, which will bring great harm to the WSN in a short time. Therefore, as an important technical means of network security, WSN network intrusion detection technology has attracted wide attention from scholars [26].

At present, WSN intrusion detection is mainly divided into anomaly detection, misuse detection, specification-based detection and hybrid system detection [15]. The existing intrusion detection methods mainly include: support vector machine [7, 20], artificial neural network [2, 9], Naive Bayes [10], Bayesian Network [23], decision tree [22], random forest [14], artificial immunity [8], random weight neural network [25] and other methods. For example, in [7], a hybrid method of support vector machine and genetic algorithm was proposed. The genetic algorithm was used to select the feature subset from the original feature set, and SVM was used as the classifier for intrusion detection. The method obtained 97.3% detection rate. However, the detection efficiency of unknown attacks is not efficient.

In [20], an intrusion detection system based on SVM and principal component analysis (PCA) was proposed. For KDDcup99 data, PCA combined with SVM algorithm was used for intrusion detection. This method reduces data analysis time and improves intrusion detection performance, but it cannot identify the different types of attacks. In [9], a back propagation learning algorithm was proposed to optimize the back propagation neural network (BPNN) intrusion detection system. For the KDDcup99 data, it has higher detection rate and lower false detection rate, but the algorithm complexity is higher.

In [23], an intrusion detection method based on ensemble learning was proposed. By Using the KDDcup99 data, the Bayesian network and the random tree were first used

as the base classifier for voting classification, and then identify if an attack has occurred. The algorithm as a whole has high detection efficiency, but the accuracy of U2R attacks was low.

In [14], a lightweight intrusion detection system based on decision tree was established, which improves the detection rate and reduces the complexity of the algorithm, but it does not detect unknown attacks. In [8], it compared the performance of supervised machine learning classifiers, proving that the detection performance of random forests is the best. In [25], an improved clonal selection algorithm was proposed. By selecting the best individual and cloning to detect the intrusion behavior, it was proved that the proposed artificial immune method is better than the artificial neural network.

In [6], a semi-supervised learning method based on fuzziness was proposed. The unlabeled sample was combined with the supervised learning algorithm to optimize the performance of the classifier. The random weight neural network was used as the base classifier to improve the classification ability. However, only two types of tasks can be detected, and multiple attacks cannot be detected.

In [21], considering the characteristics of wireless sensor networks, a detection model based on clustering mutual coordination was proposed. The intrusion detection rate was enhanced and the false detection rate was reduced. However, it is complicated to update the CA-AFSA-BP system during the detection process. And the detection rate of unknown attacks is not high.

In [24], a two-level feature selection method based on SVM was proposed. Fisher and information gain were used to filter noise and irrelevant features respectively in the filtering mode. By reducing the feature dimension, the modeling time and testing time of the system were reduced. However, when the number of training samples increases, the system overhead is large, and the classification detection performance is not high.

In [18], a cluster network intrusion detection system was proposed. Each node calculates the reputation value according to the behavior of observing neighbor nodes. The base station detects the malicious nodes by combining the reputation value and the misuse detection rules. However, because the reputation value calculation method has a great influence on the detection rate, which leads to the excessive dependence on the reputation value calculation method.

In [13], in order to solve the problem of dimension hazard in high-dimensional feature space, a SVM intrusion detection system based on self-encoding network was proposed, which is suitable for high-dimensional spatial information extraction tasks, and it can also reduce the intrusion detection model classification training time and test time. It satisfies the real-time requirements of intrusion detection, but the detection performance of R2L, U2R and other attack behaviors is not high.

In [4], a special WSN data set was developed, and the collected data set is called WSN-DS. It can help researchers better detect and classify WSN's four types of denial of service (DoS) attacks, including Blackhole, Grayhole, Flooding, and Scheduling. The data set is used to train the artificial neural network (ANN) to detect and classify different attacks. By analyzing the above research work, the existing WSN intrusion detection method generally has a large computational load, and the dimension of the traffic data to be processed is too high, which cannot effectively detect multiple attack types and the detection efficiency of unknown attacks is low.

In [17], a novel approach called SCDNN for sensor network intrusion detection was proposed, which combines spectral clustering (SC) and deep neural network (DNN) algorithms. es an effective tool of study. The algorithm has a strong ability of sparse attack classification and effectively improves the detection accuracy of the actual security system. However, the limitations of SCDNN are that its weight parameters and the threshold of each DNN layer need to be optimized, and the k and s parameters of the cluster are determined by experience, rather than by mathematical theory.

In [16],a localization attack recognition method using a deep learning architecture was proposed, by learning the positional and topological feature based on SDA-based deep architecture, the classification accuracy can be significantly improved, but the time complexity and space complexity are relatively large.

In [3], a novel intrusion detection system based on neuro-fuzzy classifier in binary form for packet dropping attack in ad hoc networks was proposed. Simulation results show that efficiently detect the packet dropping attack with high true positive rate and low false positive rate.

Aiming at the shortcomings of the above research, this paper proposes a WSN intrusion detection model based on information gain ratio and Bagging algorithm. The model uses feature gain ratios for feature selection and reduces feature dimensions by removing extraneous features. The Bagging algorithm is used to construct an ensemble classifier to train the improved C4.5 decision tree, and the parameters of the C4.5 decision tree are optimized by multiple iterations to improve the classification accuracy of the classifier. A majority voting mechanism is used for the classification results to detect intrusion behavior. The experimental results show that the model can identify different types of attacks. Compared with the existing intrusion detection methods, the detection accuracy is improved, and many types of attacks can be detected.

The remaining part of this paper is organized as follows. Section 2 introduces related theory, including WSN network topology, feature selection and ensemble theory. Section 3 describes in detail the specific implementation process of the proposed WSN intrusion detection model in this paper. Section 4 gives the experimental results and performance analysis as compared with other related methods. Finally, we conclude our paper in Section 5.

2 Related Theory

2.1 WSN Network Topology

WSN mainly has three kinds of network topologies, which are divided into plane structure, cluster based structure and hierarchical structure [19], as shown in Figure 1. The WSN consists of three parts: Sensor nodes, cluster head nodes and base station. Sensor nodes are used to monitor the target area and collect data from the area. These nodes are arranged in respective clusters, and the sensed data is simply processed and transmitted to the cluster head node. The cluster head nodes collect and process the sensor node data in the cluster and transmit it to the base station. The cluster head nodes in the base station management scope can monitor the behavior of the cluster head nodes in real time, and the intrusion detection model can be deployed to the base station. When the base station receives the traffic data from the cluster head node, each piece of data is processed, and the intrusion detection model is used to determine whether an attack behavior has occurred in the WSN.

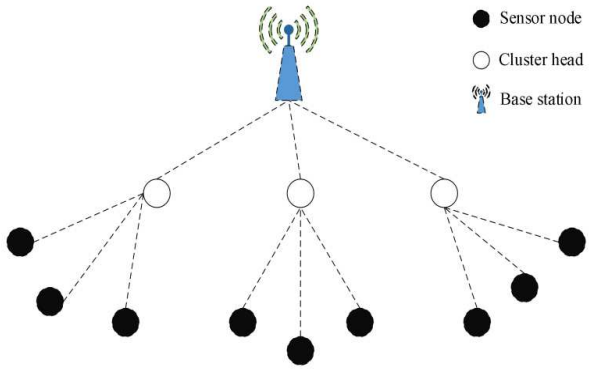


Figure 1: WSN network topology

2.2 Feature Selection

In the WSN, the traffic data dimension is high, some traffic characteristics are not related to the intrusion attack, and the node resources in the WSN are limited. Therefore, the WSN intrusion detection system introduces data preprocessing methods such as feature selection and data dimensionality reduction to remove irrelevant features and reduce the computational load of the intrusion detection method and enhance the intrusion detection efficiency.

The information gain ratio [1] is a feature selection method based on information theory, the specific definitions are as follows:

Definition 1. *Information entropy: The information entropy of a random variable is used to measure the degree of redundancy of the variable. Suppose that in a classification system, C indicates that the category is divided into $c_1, c_2, \ldots, c_n$, n represents the total number of classifications. Then the information entropy $H(C)$ of the classification system is defined as follows:*

$$H(C) = -\sum_{i=1}^{n} P(c_i) log P(c_i), \tag{1}$$

where $P(c_i)$ is the probability of the category $c_i (1 \leq i \leq n)$ at different values.

Definition 2. *Conditional entropy: Conditional entropy can evaluate the uncertainty of the value of a feature, suppose there are X pieces of data in the data set, and each piece of data has s features, which are expressed as $A=\{f_1, f_2, \ldots, f_s\}$. When the overall distribution of feature set A is fixed, the conditional entropy $H(C/A)$ is defined as follows:*

$$H(C) = -\sum_{f \in A} \sum_{c \in C} p(f,c) log p(c/f), \tag{2}$$

where $H(C/A)$ represents the uncertainty of the category C under the condition that the feature set A is different in value, and $P(c/f)$ represents the conditional probability that the category c takes the value under the condition of the feature $A = f$.

Definition 3. *Information gain: The information gain reflects the importance of the feature. The greater the information gain, the more important the features are. Then the information gain IG brought by the feature set A to the system is defined as follows:*

$$IG(A) = H(C) - H(C/A). \tag{3}$$

The information gain tends to select attributes with more branches, which may lead to over-fitting. In order to change the shortcomings of information gain, the information gain ratio is used to judge the partitioning attribute.

Definition 4. *Information gain ratio:*

$$G\_R(A) = IG(A)/H(A), \tag{4}$$

where $G\_R(A)$ is the information gain ratio of feature set A. $H(A)$ is the information entropy when feature A is a random variable according to the Equation (1).

The pseudo code of the information gain ratio feature selection algorithm is defined as Algorithm 1. where $num(S)$ represents the number of features in the selected feature set S, and $max(G\_R(f_i))$ represents the maximum information gain ratio in the feature set $A = f_1, f_2, \ldots, f_s$, The first k selected features are added to the set S, and finally the feature set S is obtained. The algorithm description is detailed in the appendix.

Algorithm 1 Information gain ratio feature selection algorithm

1: Input: Training data_set and feature selection quantity k
2: Output: Selected feature set S
3: Initialize feature sets $S = \varnothing$ /* Initialize feature set S to an empty set */
4: Initialize all feature sets $A = f_1, f_2, \ldots, f_s$ /* s is the number of attribute features */
5: Calculate the information gain ratio of each feature in feature set A from Equation (4)
6: **while** num$(S) < k$ **do**
7: Select $max(G\_R(f_i))$, add the attribute f_i to the feature set S.
8: **end while**
9: The selected feature set S is obtained, and the number of selected features is k.

2.3 Ensemble Theory

2.3.1 Bagging Algorithm

The ensemble classifier is a kind of supervised learning method. As a kind of ensemble classifier, Bagging can avoid the over-fitting of the classifier and can improve the detection efficiency of unknown attacks. The ensemble learning classifier includes m base classifiers, which are trained by Bootstrap sampling method. After m times sampling, the results of m base classifiers are obtained. Finally, the classification results of the ensemble classifier are integrated according to the majority voting principle. Figure 2 shows the specific flow of the Bagging algorithm.

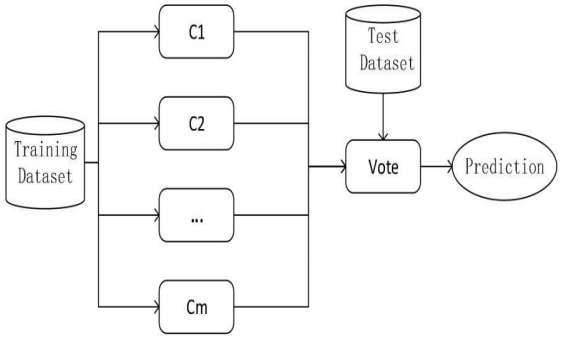


Figure 2: Bagging algorithm

2.3.2 Improved C4.5 Algorithm

The C4.5 algorithm is an algorithm to solve the problem of machine learning classification. The algorithm can find a mapping relationship between feature values and categories, and this mapping relationship can be used to classify unknown intrusion types. The C4.5 algorithm is a tree structure similar to a flow chart. A non-leaf node represents a test on an attribute. Each branch represents a test output, and each leaf node stores a class label. The advantage of this algorithm is that it does not require any domain knowledge, it is suitable for detective knowledge discovery, and it's highly efficient for detecting unknown attack types. For a leaf node, it covers q samples, there are e errors and the penalty factor is 0.5. Assuming that a decision tree has r leaf nodes, the prediction error of the decision tree is ER, which the formula is as follows Equation (5):

$$ER = (\sum_{i=1}^{r} e_i + 0.5 \times r) / \sum_{i-1}^{r} q_i \quad (5)$$

where e_i is the number of samples misclassified in the i-th leaf node of the subtree, and q_i represents the number of samples in the i-th leaf node of the subtree.

The improved C4.5 algorithm pseudo code is defined as follows:

Algorithm 2 demonstrates the process of detecting anomalous intrusions in the WSN by the improved C4.5 classifier. First, if the node satisfies the stop split condition, all records belong to the same category, and it is set as a leaf node; Then the feature with the largest information gain rate is selected for splitting, and the first two steps are repeated until all data classification is completed. Finally, the generated tree needs to be dynamically pruned to reduce the prediction error. The algorithm description is detailed in the appendix.

3 The Proposed Model of WSN Intrusion Detection

WSN Intrusion detection model based on information gain ratio and Bagging algorithm, the shortened form is WI-IGRB, the information gain ratio is used for feature selection, and then the parameters of the ensemble classifier are optimized through 10 iterations, and the dynamic pruning process is introduced. The iteration 10 times is relatively suitable. The parameters of the Bagging algorithm are optimized during the iterative process, and the complexity of the algorithm cannot be too high that may lead to over-fitting of the model. The dynamic pruning process starts from the leaf node of the C4.5 decision tree, calculates the prediction error from the bottom to the node and the prediction error after pruning. If the prediction error after pruning is relatively small, the node is cut off. This process is repeated repeatedly until the prediction error is minimized. Finally, the majority voting system is used to count the type of the most predicted votes in the classifier and use it as the final result of the ensemble classifier. Figure 3 is a flow chart of the proposed WSN intrusion detection model.

As shown in Figure 3, the wireless sensor node collects environmental data and transmits the data to the cluster head node. The cluster head node processes the collected data and transmits it to the base station. The collected traffic data is selected from the base station as a training

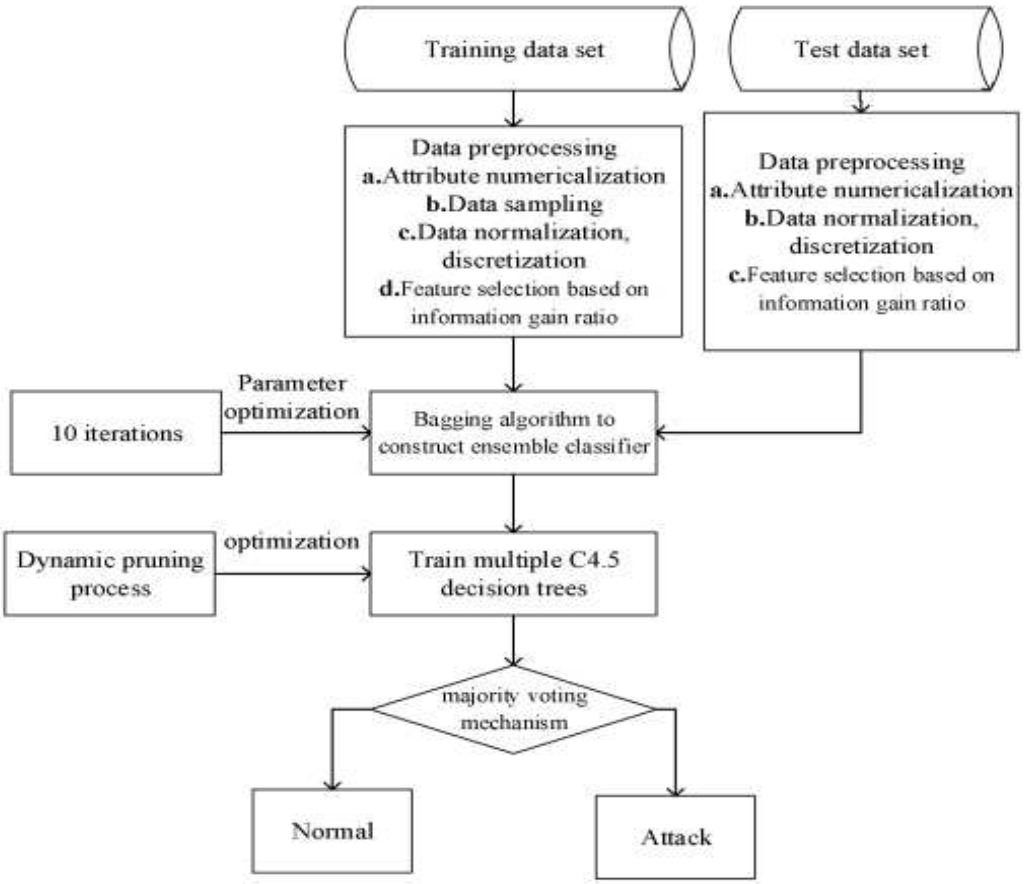


Figure 3: Flow chart of WSN intrusion detection model

data set and a test data set respectively, and the proposed intrusion detection model is trained. It is mainly divided into the following two stages:

1) Model training phase: Preprocessing the training data set, including numeralization, proportional sampling of data, data normalization and discretization operations, and feature selection based on information gain ratio; using Bagging algorithm to construct ensemble classifier, multiple C4.5 decision trees are trained, and the dynamic pruning process is introduced to reduce the prediction error. Finally, the classification prediction is carried out by the majority voting mechanism.

2) Model intrusion detection phase: Preprocessing the collected test data set, including digitizing some features, data normalization and discretization processing, and feature selection based on information gain ratio; using trained integrated detection model Classification; The majority of voting mechanisms are used to integrate classifications to determine whether intrusion has occurred.

Majority voting mechanisms are defined as Equation (6), where m is the number of samples collected by the Bootstrap sampling method, l is the traffic data to be classified, L is the result of the classification, and C^* is used to count the predicted votes in the m classifiers C_i. The most type and use it as the final result of the integrated classifier.

$$C^*(l) = max(\sum_{i}^{m} \alpha(C_i(l) = L). \tag{6}$$

The proposed WSN intrusion detection model algorithm in this paper is defined as follows:

where m is the number of samples collected by the Bootstrap sampling method, and N is the number of iterations of the algorithm. In the model training phase, the Boostrap Sampling sampling method independently trains the decision tree C_i by randomly selecting m sample numbers. Finally, the prediction function is generated in parallel to get the ensemble classifier C^*. In the model intrusion detection phase, the trained ensemble classifier C^* is used to determine whether intrusion behavior occurs in the WSN. The algorithm description is detailed in the appendix.

Algorithm 2 Improved C4.5 algorithm

Input: Data Set B
Output: T-decision tree after dynamic pruning
[x, s] = size(B) /* x is the number of data set B, s is the number of attribute features in data set B */
T={}
if B belongs to the same category or other stopping criteria **then**
  break
end if
while feature set $S = f_1, f_2, \ldots, f_s$ **do**
  Calculate the branch information entropy and conditional entropy of each feature by Equations (1)-(2)
  Calculate the information gain rate $G\_R(f_j)$ of the feature f_j according to the Equation (4)
end while
f_best=Select the maximum information gain rate $max(G\_R(f_j))$
Use f_best as the decision node and join T
Remove f_best from B to get subset B*
if $x > 0$ **then**
  Return to step 3
end if
while B* **do**
  T*=C4.5(B*)
  Attach T* to the corresponding branch of the tree
end while
while T is not NULL **do**
  Calculate the prediction error of the decision tree T according to Equation (5) and the prediction error of the pruning off T leaf node
  if Prediction error of pruning T-leaf nodes < Prediction error of unpruned T-leaf nodes **then**
    Pruning the T-leaf node
  end if
  Pruning upward
end while
Return dynamic pruned T decision tree.

Algorithm 3 WSN intrusion detection model algorithm

Input: Train dataset, test dataset
Output: Intrusion detection result
Model training phase:
Preprocessing the train dataset, the k important features are selected by Algorithm 1
for $n = 1$ *to* N **do**
  for $i = 1$ *to* m **do**
    Sample Rifrom sample train dataset using Bootstrap sampling method
    The improved C4.5 decision tree C_i in Algorithm 2 is trained by the sample R_i
  end for
end for
Using the Equation (6) to get the ensemble classifier C^*
Model intrusion detection phase:
Preprocessing the test dataset, the k important features are selected by Algorithm 1
while test dataset **do**
  Using the ensemble classifier C^* to determine whether an intrusion has occurred.
  Output intrusion detection results
end while

4 Experimental Results and Analysis

4.1 Experimental Data Set Selection

The experiment uses the WSN dataset WSN-DS [4], and the simulator NS-2 was used to simulate the wireless sensor network environment. Based on the LEACH routing protocol, each data has 23 features and simulates four attack types: Blackhole, Grayhole, Flooding, and Scheduling. A total of 374,661 traffic data were collected in the WSN-DS dataset, and 10% of the data were randomly selected as the experimental data set. 60% of the data were used as the training data set, and 40% of the data were used as the test data set. The experimental environment was performed on a 64-bit Windows 7 operating system with 8 GB of RAM and an Intel core i5-3230 CPU. Table 1 illustrates WSN simulation parameters. The data distribution is shown in Table 2.

Table 1: WSN simulation parameters

Parameter	Value
Number of cluters	100
Number of clusters	5
Network area	100m×100m
Base station location	(50,175)
Size of packet header	25 bytes
Size of data packet	500 bytes
Routing protocol	Leach
Simulation time	3600s

Table 2: Distribution of WSN-DS data sets

Data Set	Training set 60%	Testing set 40%
Blackhole	603	402
Grayhole	876	583
Flooding	199	132
Scheduling	398	266
Normal	20404	13603
Sum	22480	14986

The experiment also uses the NSL-KDD dataset, an improved version of the KDD'99 dataset, which removes a large amount of redundant data and maintains the original attack type ratio more suitable for evaluating the actual performance of the intrusion detection algorithm. Each traffic record contains 41-dimensional feature data of various continuous, discrete, and symbol types. The NSL-

KDD includes four attack categories (DoS, Probe, R2L, and U2R) [5]. The NSL-KDD includes a training dataset KDDTrain+_20Percent and a test dataset KDDTest-21. The training data set consists of 21 types of attacks, and 17 new attack types are added to the test set. First, the NSL-KDD data set needs to be preprocessed, and the feature protocol_type, service and attack class is digitized. Then, the data set is divided into five classes, normal, DoS, Probe, U2R, and R2L, mapped to values 1-5 respectively. Finally, normalize the values of the src_bytes and dst_bytes field columns to map the range to [0,1]. The specific data distribution of the NSL-KDD data set is shown in Table 3.

Table 3: Distribution of NSL-KDD data sets

Data Set	KDDTrain+_20Percent	KDDTest+
Normal	13449	9711
DoS	9234	7458
Probe	2289	2421
U2R	11	200
R2L	209	2754
Sum	25192	22544

4.2 Experimental Performance Index

In order to measure the performance of the wireless sensor network intrusion detection model, the true positive rate (TPR), false positive rate (FPR), accuracy (Acc), precision (P) indicators are used for measurement. TP indicates that the true value is a normal sample and is predicted as the number of normal samples. FN indicates that the true value is a normal sample and is predicted as the number of abnormal samples. FP indicates that the true value is an abnormal sample and is predicted as the number of normal samples. TN indicates that the true value is an abnormal sample and is predicted as the number of normal samples. Table 4 shows the definitions of TP, FP, TN and FN.

Table 4: Definition of *TP*, *FP*, *TN* and *FN*

True value	Predicted	
	Normal	Abnormal
Normal	TP	FN
Abnormal	FP	TN

$$
\begin{aligned}
TPR &= TP/(TP+FN) \\
FPR &= FP/(FP+TN) \\
P &= TP/(TP+FP) \\
Acc &= (TP+TN)/(TP+FN+FP+TN)
\end{aligned}
$$

where *TPR* indicates the probability that the true value is normal, the probability of the prediction is positive. *FPR* indicates the probability that the true value is abnormal, and the prediction is positive; *P* indicates the probability that the prediction is normal and the correct prediction is normal. *Acc* represents the accuracy of the prediction result, and the number of normal samples is predicted divided by the total number of samples.

4.3 Feature Selection Method

The existing intrusion detection method adopts data preprocessing methods such as feature selection and data dimensionality reduction to reduce the computational load of the intrusion detection method and enhance the detection efficiency. The main feature selection and dimension reduction methods are: correlation feature selection, linear discriminant analysis, mutual information, information gain, gain ratio, principal component analysis and other methods [11]. In this study, attribute reduction is used for WSN-DS data. First, features that have no or little impact on data types were eliminated, and then common feature selection methods and information gain ratio were selected for comparative analysis. The selected features and performance are shown in Table 5.

It can be seen from Table 3 that the experiment uses the Algorithm 1 information gain ratio to select features, set k=14, select 14-dimensional features from the 23-dimensional features, and select the flow feature set S={3,5,6,7,8,9,10,11,12,13,15,16,17,18}, and use Acc as an evaluation index. Using the information gain feature selection method, if the number of features is much larger than the number of categories, the information gain will become large, and the generalization ability will be reduced without using other more effective classification information. The information gain ratio introduces split information, and the feature splitting information with a large number of values becomes large, which can effectively control the problem of excessive information gain. Through the experiment, the principal component analysis method Acc reached 94.91%, and when using the information gain method, the Acc was 98.52%, and when the information gain ratio feature selection method was used, the Acc was 98.75%. The proposed WSN intrusion detection model has a better classification accuracy when choosing the information gain ratio as the feature selection method. Table 6 lists the selected traffic characteristics and specific description information in the WSN-DS. It includes the number of features in the WSN-DS, the name of features, and the description of corresponding features in the data set.

The information gain ratio method is used to select the important features of the WSN-DS data set traffic characteristics. Comparing the selected features and specific information listed in Table 6, the final selected traffic feature set is S={3,5,6,7,8,9,10,11,12,13,15,16,17,18}.

Table 7 lists the selected traffic characteristics in the NSL-KDD. It includes the number of features in the NSL-KDD, the name of features in the dataset.

The information gain ratio algorithm is used to select the important features in NSL-KDD. The last selected

Table 5: Comparison of feature selection methods

Feature selection method	**Feature selection result**	**Acc (%)**
Principal component analysis	1,2,3,4,5,6,7,8,9,10,11,12,13,14	94.91
Information gain	1,3,4,5,6,7,8,10,11,12,13,15,17,18	98.52
Information gain ratio	3,5,6,7,8,9,10,11,12,13,15,16,17,18	98.75

Table 6: WSN-DS data set selected traffic features

Feature number	**Feature name**	**Description**
1	Node ID	Node ID number
2	Time	Node runtime
3	IS CH	Used to mark whether the node is a cluster head
4	Who CH	Cluster head ID
5	Distance to CH	Distance between node and cluster head
6	Energy consumption	Energy consumed
7	ADV CH send	The number of the advertise CH's broadcast messages sent to the nodes
8	ADV CH receives	The number of advertise CH messages received from CHs
9	Join REQ send	The number of join request messages sent by the nodes to the CH
10	Join REQ receive	The number of join request messages received by the CH from the nodes
11	ADV SCH send	The number of join advertise TDMA schedule broadcast message sent to the nodes
12	ADV SCH receives	The number of scheduled messages received by the CH
13	Rank	Order of node TDMA scheduling
14	Data sent	The number of packets sent from the normal node to its CH
15	Data received	The number of packets received by the node from the CH
16	Data sent to BS	The number of packets sent to the BS
17	Distance CH to BS	Distance between CH and BS
18	Send Code	The cluster sending code
19	Attack Type	Type of the node

Table 7: NSL-KDD dataset features

No.	**Feature**	**No.**	**Features**
1	Duration	22	Is_guest_login
2	Protocol_type	23	Count
3	Service	24	Srv_count
4	Flag	25	Serror_rate
5	Src_bytes	26	Srv_serror_rate
6	Dst_bytes	27	Rerror_rate
7	Land	28	Srv_rerror_rate
8	Wrong_fragment	29	Same_srv_rate
9	Urgent	30	Diff_ser_rate
10	Hot	31	Srv_diff_host_rate
11	Num_failed_logins	32	Dst_host_count
12	Logged_in	33	Dst_host_srv_count
13	Num_compromised	34	Dst_host_same_srv_rate
14	Root_shell	35	Dst_host_diff
15	Su_attempted	36	Dst_host_same_srv_port_ra
16	Num_root	37	Dst_host_serror_rate
17	Num_file_creations	38	Dst_host_serror_rate
18	Num_shells	39	Dst_host_srv_serror_rate
19	Num_access_files	40	Dst_host_serror_rate
20	Num_outbound_cmds_files	41	Dst_host_srv_serror_rate
21	Is_host_login		

feature set is {9, 26, 25, 4, 12, 39, 30, 38, 6, 29, 5, 3, 37, 11, 22, 35, 34, 14}.

4.4 Performance Analysis

Table 8 shows the detection performance of the proposed WSN intrusion detection model for Normal type and attack types based on WSN-DS, such as Blackhole, Grayhole, Flooding, and Scheduling.

As can be seen from Table 8, the detection accuracy of the proposed WSN intrusion detection model for attacks in the WSN, such as Blackhole, Grayhole, Flooding, and Scheduling, is 99.04%, 97.96%, 99.02%, and 96.21%, respectively. The detection accuracy of the normal state is 98.85%. The weighted average results show that the model true positive rate is 99.4%, the false positive rate is 1.9%, the precision is 99.4%, and the classification accuracy rate is 98.75%. The experimental results show that the proposed WSN intrusion detection model has better performance in attack detection in WSN environment and can identify different attack types.

Table 9 shows the WSN intrusion detection model and PCA-SVM [20], Naive Bayes [10], Bayesian Nerwork [23], IG-C4.5 [22], Boosting-C5.0 [14], ANN [4] methods. The specific results of performance comparison were measured and compared using TPR, FPR, Acc, and P index.

As can be seen from Table 9, the TPR of the proposed method reaches 99.4%, which is higher than that of PCA-SVM, Naive Bayes, Bayesian Network, IG-C4.5, and ANN. Among them, the TPR of Naive Bayes method is 95.2%, which is the smallest compared with the above methods. However, the false positive rate FPR of this method is 1.9%, which is higher than Naive Bayes, Bayesian Network and ANN methods. When the detection rate of the WSN intrusion detection model is increased, the data that causes the true value of the attack behavior is incorrectly predicted as the probability of a normal sample increases, and then he false positive rate increases.

The false positive rate of IG-C4.5 method reaches 3.8%, which is the highest false positive rate compared with other methods. The Acc and P are respectively 98.8% and 99.4%, which are higher than PCA-SVM, Naive Bayes, Bayesian Network, IG-C4.5, and ANN methods. Among them, Acc is 0.25% higher than Boosting-C5.0. The reason is that the selection of Boosting training sets is related to the learning results of the previous rounds, which may lead to over-fitting and reduce classification accuracy. In summary, the proposed method performs better than other intrusion detection methods.

The NSL-KDD dataset is used to evaluate the performance of the proposed method and a 10-fold cross validation was performed. In a 10-fold cross validation, the data was divided into 10 replicates of equal size. In each iteration, each part of the data is used for verification and the remaining nine parts are used to train the model. Table 10 shows the performance of the proposed WSN intrusion detection model using the NSL-KDD data set and 10-fold cross validation. Detection performance for different attack types DoS, Probe, U2R, and R2L and Normal appearing in NSL-KDD.

It can be seen from Table 10 that the detection accuracy of the proposed intrusion detection model for NSL-KDD, such as DoS, probe, U2R and R2L, is 99.85%, 99.35%, 59.05%, and 94.0%, respectively. The detection accuracy of the Normal reaches 99.65%. The overall TPR of the model is 99.69%, the FPR is 0.31%, the P is 99.6%, and the Acc is 99.69%.

Table 11 shows the WSN intrusion detection model and PCA-SVM, Naive Bayes, Bayesian Nerwork, IG-C4.5, Boosting-C5.0, ANN methods. The specific results of performance comparison were measured and compared using TPR, FPR, Acc, and P index

As can be seen from Table 11, the performance of the proposed method was evaluated using the NSL-KDD data set and 10-fold cross validation was performed. The TPR of the method in this paper reaches 99.69%, which is higher than that of PCA-SVM, Naive Bayes, Bayesian Nerwork, IG-C4.5, and ANN. The FPR of this method is 0.31%, which is lower than other methods. The reason is that the Bagging ensemble algorithm effectively reduces the variance of the model.

The FPR of Naive Bayes method reaches 11.42%, which is the highest FPR compared with other methods. The Acc of this method is 99.69%, and the Acc is lower than that of Boosting-C5.0, compared with PCA-SVM and Naive Bayes. Bayesian Nerwork, IG-C4.5, and ANN methods are high. Comparing the model building time, we can see that the proposed method has a lower time. The reason is that the selection of Bagging algorithm training set is random, and each round of training sets is independent of each other, while Boosting the selection of each round of training sets is related to the learning results of the previous rounds.

The various predictive functions of Bagging can be generated in parallel, and the various predictive functions of Boosting can only be generated sequentially, such as neural networks, which are extremely time-consuming learning methods. Bagging can save a lot of time overhead through parallel training. At the same time, in the model establishment stage, the information gain ratio method is used to reduce the dimension of the traffic data, and the features with low importance are removed. The 18-dimensional features are selected from the 41-dimensional features. The calculation and time overhead in the detection process are effectively reduced, which is more suitable for the WSN intrusion detection environment.

The experiment also used the training dataset KDDTrain+_20Percent and the test dataset KDDTest-21 to evaluate the performance of the model. Table 12 shows the performance of the WSN intrusion detection model and PCA-SVM, Naive Bayes, Bayesian Nerwork, IG-C4.5, Boosting-C5.0, ANN methods.

Table 8: Performance of the WSN intrusion detection model based on WSN-DS

Performance	Blackhole	Grayhole	Flooding	Scheduling	Normal	Weighted average results
TPR (%)	98.2	96.1	98.2	92.4	99.8	99.4
FPR (%)	0.1	0.2	0.1	0.0	2.1	1.9
P (%)	96.5	96.3	90.2	97.6	99.8	99.4
Acc (%)	99.04	97.96	99.02	96.21	98.85	98.75

Table 9: Comparison of performance of different methods of WSN intrusion detection model

Methods	TPR (%)	FPR (%)	P (%)	Acc (%)
PCA-SVM	96.6	8.6	96.7	94.0
Naive Bayes	95.2	1.0	96.5	97.1
Bayesian Network	96.5	0.9	97.7	97.8
IG-C4.5	97.8	3.8	98.3	97.0
Boosting-C5.0	99.4	2.4	99.4	98.5
ANN	98.5	1.7	98.7	98.4
The proposed method	99.4	1.9	99.4	98.75

Table 10: Performance of intrusion detection methods using NSL-KDD

Performance	Probe	DoS	U2R	R2L	Normal	Sum
TPR (%)	98.8	99.9	18.2	88.0	99.8	99.69
FPR (%)	0.1	0.2	0.1	0.0	0.5	0.31
P(%)	99.4	99.7	98.9	96.8	99.6	99.6
Acc	99.35	99.85	59.05	94.00	99.65	99.69

Table 11: Comparison of performance of different methods using NSL-KDD 10-fold cross validation

Methods	TPR (%)	FPR (%)	P (%)	Acc (%)	Model buildng time(s)
PCA-SVM	93.02	6.97	94.26	93.02	-
Naive Bayes	88.58	11.42	88.67	88.58	-
Bayesian Network	96.69	3.72	96.68	96.48	-
IG-C4.5	96.6	5.25	96.53	95.7	-
Boosting-C5.0	-	0.38	-	99.96	6.38
ANN	99.24	0.83	99.18	99.2	198.67
The proposed method	99.69	0.31	99.60	99.69	5.78

Table 12: Comparison of performance of different methods using NSL-KDD

Methods	TPR (%)	FPR (%)	P (%)	Acc (%)	Model buildng time(s)
PCA-SVM	76.5	31.1	83.4	72.7	-
Naive Bayes	78.6	27.7	82.7	75.45	-
Bayesian Network	76.5	31.1	83.4	72.7	-
IG-C4.5	77.6	27.0	85.0	75.3	-
Boosting-C5.0	98.9	45.19	-	80.56	7.31
ANN	80.1	26.3	85.1	76.9	201.34
The proposed method	81.2	26.2	85.3	77.5	6.01

4.5 Algorithm Analysis

In order to further verify the performance of the proposed method, two performance indicators, time complexity and space complexity are analyzed in detail. The comparison results are shown in Table 13.

As can be seen from Table 13, the number of data sets is X, and the number of flow characteristics is k, m is the number of samples collected by Bootstrap sampling method, and N is the number of algorithm iterations. The time complexity of the PCA-SVM method is $O(5kX)$ and the space complexity is $O(X^2)$. The time complexity of the Naive Bayes method is $O((k+k^2)X)$ and the space complexity is $O(2kX)$. The time complexity of the Bayesian Network method is $O((k+k^2)X)$ and the space complexity is $O(2kX)$. The time complexity of IG-C4.5 method is $O(X+log_2N)$, and the space complexity is $O(kX)$. The time complexity of the Boosting-C5.0 method is $O(X+Nmlog_2X))$, and the space complexity

Table 13: Comparison of time complexity and space complexity performance

Methods	Time complexity	Space complexity
PCA-SVM	$O(5kX)$	$O(X^2)$
Naive Bayes	$O((k+k^2)X)$	$O(2kX)$
Bayesian Network	$O((k+k^2)X)$	$O(2kX)$
IG-C4.5	$O(X+log_2N)$	$O(kX)$
Boosting-C5.0	$O(X+Nmlog_2X)$	$O(kX)$
ANN	$O(kNX)$	$O(kX)$
The proposed method	$O(X+Nmlog_2X)$	$O(mkX)$

is $O(kX)$. The time complexity of the ANN method is $O(kNX)$ and the space complexity is $O(kX)$.The time complexity of the proposed method is $O(X+Nmlog_2X)$, which is higher than that of the IG-C4.5 method. When the number of data sets is very large, $m < k << X$, then $O(X+Nmlog_2X)) < O(kNX)$, It can be seen that the algorithm time complexity of the proposed method is more time complex than that of the ANN method, and the space complexity is relatively constant.

5 Conclusions

A WSN intrusion detection model based on information gain ratio and Bagging algorithm is proposed. In the data preprocessing stage, the feature selection method based on information gain ratio is used to reduce the dimension of the collected WSN traffic data, which reduces the computational complexity of the intrusion detection method and effectively reduces the computation and time overhead in the detection process. In the integrated learning phase, the Bagging algorithm is used to construct the integrated classifier, and several improved C4.5 decision trees are trained. The dynamic pruning process is introduced to reduce the prediction error, and the parameters of the integrated classifier are optimized by 10 iterations. In the intrusion detection phase, the trained integrated classifier is used to classify the data, and the majority voting mechanism is used to judge whether the intrusion behavior occurs. The experimental results show that the detection accuracy of Blackhole, Grayhole, Flooding, Scheduling and Normal are 99.04%, 97.96%, 99.02%, 96.21% and 98.85%, respectively. Compared with other intrusion detection methods, the detection accuracy is improved. A variety of attack types can be effectively detected. Subsequent research focuses on extending other types of attacks in the WSN dataset, using other feature selection techniques combined with deep learning models for wireless sensor network intrusion detection.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61363078), the Research Project in Universities of Education Department of Gansu Province (2017B-16, 2018A-187). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] A. Abduvaliyev, A. S. K. Pathan, R. Roman J. Zhou, and W. C. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013.

[2] H. I. Ahmed, N. A. Elfeshawy, S. F. Elzoghdy, H. S. El-Sayed, and O. S. Faragallah, "A neural network-based learning algorithm for intrusion detection systems," *Wireless Personal Communications*, vol. 97, no. 2, pp. 3097–3112, 2017.

[3] V. N. T. AlkaChaudhary and A. Kumar, "A new intrusion detection system based on soft computing techniques using neuro-fuzzy classier for packet dropping attack in manets," *International Journal of Network Security*, vol. 18, no. 3, pp. 514–522, 2016.

[4] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "Wsn-ds: A dataset for intrusion detection systems in wireless sensor networks," *Journal of Sensors*, vol. 2016, no. 2, pp. 1–16, 2016.

[5] Amrita and K. K. Ravulakollu, "A hybrid intrusion detection system: Integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classiers," *International Journal of Network Security*, vol. 20, no. 1, pp. 41–55, 2018.

[6] R. A. R. Ashfaq, X. Z. Wang, and J. Z. Huang, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484–497, 2017.

[7] B. M. Aslahi-Shahri, R. Rahmani, and M. Chizari, "A hybrid method consisting of ga and svm for intrusion detection system," *Neural Computing and Applications*, vol. 27, no. 6, pp. 1–8, 2016.

[8] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Computer Science*, vol. 89, no. 2016, pp. 117–123, 2016.

[9] Z. Chiba, N. Abghour, and K. Moussaid, "A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection," *Computers and Security*, vol. 75, no. 2018, pp. 36–58, 2018.

[10] D. H. Deshmukh, T. Ghorpade, , and P. Padiya, "Intrusion detection system by improved preprocessing methods and na?ve bayes classifier using nsl-kdd

99 dataset," in *International Conference on International Conference on Electronics and Communication Systems (ICECS'14)*, pp. 1–7, Feb. 2014.

[11] R. H. Dong, D. F. Wu, Q. Y. Zhang, and H. X. Duan, "Mutual information-based intrusion detection model for industrial internet," *International Journal of Network Security*, vol. 20, no. 1, pp. 131–140, 2018.

[12] Z. Feng, J. Fu, and D. Du, "A new approach of anomaly detection in wireless sensor networks using support vector data description," *International Journal of Distributed Sensor Networks*, vol. 13, no. 1, pp. 1–14, 2017.

[13] N. Gao, L. Gao, Y. He, Y. Y. He, and H. Wang, "Lightweight intrusion detection model based on self-encoding network feature dimension reduction," *Chinese Journal of Electronics*, vol. 45, no. 3, pp. 730–739, 2017.

[14] A. Garofalo, C. D. Sarno, and V. Formicola, "Enhancing intrusion detection in wireless sensor networks through decision trees," *Lecture Notes in Computer Science*, vol. 7869, no. 2, pp. 1–15, 2013.

[15] A. Ghosal and S. Halder, "A survey on energy efficient intrusion detection in wireless sensor networks," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 2, pp. 239–261, 2017.

[16] W. Hua, Y. Wen, and D. Zhao, "Identifying localization attacks in wireless sensor networks using deep learning," *Journal of Intelligent and Fuzzy Systems*, vol. 35, no. 2, pp. 1339–1351, 2018.

[17] T. Ma, F. Wang, J. J Cheng, Y. Yu, and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensor*, vol. 16, no. 10, pp. 1701, 2016.

[18] M. M. Ozcelik, E. Irmak, and S.Ozdemir, "A hybrid trust based intrusion detection system for wireless sensor networks," in *International Symposium on Networks, Computers and Communications*, pp. 1–6, Oct. 2017.

[19] N. Rachburee and W. Punlumjeak, "A comparison of feature selection approach between greedy, IG-ratio, Chi-square, and mRMR in educational mining," in *7th International, Conference on International Conference on Information Technology and Electrical Engineering*, pp. 420–424, Oct. 2016.

[20] M. C. Raja and M. M. A. Rabbani, "Combined analysis of support vector machine and principle component analysis for IDS," in *International Conference on Communication and Electronics Systems (ICCES'16)*, pp. 1–5, Oct. 2016.

[21] X. Sun, B. Yan, X. Zhang, and C. Rong, "An integrated intrusion detection model of cluster-based wireless sensor network," *PloS one*, vol. 10, no. 10, pp. e0139513, 2015.

[22] W. Wang, Y. He, J. Liu, and S. Gombault, "Constructing important features from massive network traffic for lightweight intrusion detection," *IET Information Security*, vol. 9, no. 6, pp. 374–379, 2015.

[23] Y. Wang, Y. Shen, and G. Zhang, "Research on intrusion detection model using ensemble learning methods," in *IEEE International Conference on Software Engineering and Service Science*, pp. 422–425, Nov. 2017.

[24] X. Wu, X. Peng, and Y. Yang, "Two-level feature selection method based on svm in intrusion detection," *Transactions of Communications*, vol. 34, no. 4, pp. 19–26, 2015.

[25] C. Yin, L. Ma, and L. Feng, "Towards accurate intrusion detection based on improved clonal selection algorithm," *Multimedia Tools and Applications*, vol. 76, no. 19, pp. 1–14, 2017.

[26] B. Zarpelao, Miani R S, and C. T. Kawakani, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, no. 2017, pp. 25–37, 2017.

Appendix

A 1

Algorithm 1 Information gain ratio feature selection algorithm.

Proof. Information gain ratio is a filter method of feature selection. The Information gain ratio is defined in detail in Section 2.2 of this paper. The algorithm logic steps are described in detail below.

Step1: Initialize the feature set $S = \emptyset$, which will be used to save the selected feature.

Step2: From the original feature set $A = f_1, f_2, \ldots, f_s$, the information gain ratio of each feature is calculated by the formula (4), expressed as $G\_R(f_i)$, $i \in [1, s]$.

Step3: The first k features are sequentially added to the feature set S, and the selected feature set S is output.

From the logic point of view, Information gain ratio feature selection algorithm is correct. After experimental verification, the algorithm finally selects k important features. □

B 2

Algorithm 2 Improved C4.5 algorithm.

Proof. Algorithm 2 demonstrates the process of detecting anomalous intrusions in the WSN by the improved C4.5 classifier. The algorithm logic steps are described in detail below.

Step1: If the node satisfies the stop split condition, all records belong to the same category or the maximum information gain rate is less than the threshold, indicating that the B data set does not need to be classified and break out of the program.

Step2: According to the information entropy formula (1), find the information entropy $H(S)$ of each feature in $S = \{f_1, f_2, \ldots, f_s\}$, calculate the conditional entropy of each feature in feature set S according to the conditional entropy formula (2) and obtain the information gain ratio $G\_R(f_j)$ of each feature in feature set S according to formula (4). The feature fbes with the largest information gain rate($max(G\_R(f_j))$) is selected as the decision node and added to the decision tree T. Repeat the first two steps until all data classifications are complete.

Step3: The generated tree needs to be dynamically pruned to reduce the prediction error. Firstly, delete the subtree rooted at this node, Then, make it a leaf node, the most common classification of training data assigned to the node. Finaly, when the pruned tree is not worse than the original tree for verifying the performance of the set, the node is actually deleted.

From the above, the improved C4.5 algorithm logic is correct, after experimenting, this algorithm can classify each piece of traffic data to generate a decision tree with less prediction error. □

C 3

Algorithm 3 WSN intrusion detection model algorithm.

Proof. Where m is the number of samples collected by the Bootstrap sampling method, and N is the number of iterations of the algorithm. The algorithm logic steps are described in detail below.

Step1: In the model training phase, the Boostrap Sampling sampling method independently trains the decision tree C_i by randomly selecting m sample numbers.

Step2: Using the majority voting mechanism to get the ensemble classifier C^* . Majority voting mechanisms are defined as Equation (6).

Step3: In the model intrusion detection phase, preprocessing the test dataset, as in the training dataset preprocessing. the k important features are selected by Algorithm 1.

Step4: The trained ensemble classifier C^* is used to determine whether intrusion behavior occurs in the WSN.

In summary, the algorithm logic is correct, and it has been proved by experiments. There are detailed experimental results in 4 experimental results and analysis. □

Biography

Rui-Hong Dong. Researcher, worked at school of computer and communication in Lanzhou university of technology. His research interests include network and information security, information hiding and steganalysis analysis, computer network.

Hou-Hua Yan. received the BS degrees in Computer Science and Technology from Taiyuan Institute of Technology, Taiyuan, China, in 2015. Currently, he is studying for his masters degree at Lanzhou University of Technology. His research interests include network and information security, wireless sensor network security, intrusion detection.

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

Intrusion Detection Method Based on Support Vector Machine and Information Gain for Mobile Cloud Computing

Emmanuel Mugabo and Qiu-Yu Zhang
(*Corresponding author: Qiu-Yu Zhang*)

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

(Received Aug. 26, 2018; Revised and Accepted Mar. 23, 2019; First Online June 5, 2019)

Abstract

Intrusion detection system (IDS) has become an important security method that monitors and investigates the network security in mobile cloud computing (MCC). However, in some existing methods, there are still some limitations such as high false positive rates, low classification accuracies, and low true positive rates. To counter these limitations, an intrusion detection method based on support vector machine (SVM) and information gain (IG) for MCC was proposed in this paper. In the proposed method, the SVM classifier is adopted to classify network data into normal and attack behaviors, and due to the irrelevant and redundant features found in KDD datasets, IG is used to select the relevant features and remove unnecessary features. The KDD'99 and NSL-KDD datasets are used to evaluate the effectiveness of the proposed method. Compared with other methods, the experimental results show that the proposed method can detect malicious attacks with high accuracy, true positive rate, low false positive rate and high training speed.

Keywords: Intrusion Detection; Information Gain; Malicious Attacks; Mobile Cloud Computing; Support Vector Machine

1 Introduction

Mobile Cloud Computing (MCC) is an exciting new technology, which integrates cloud computing into the mobile environment [4, 22]. According to Cisco IBSG (Online, 2016), close to 85% of the world's population has access to mobile devices as they bring some convenience, but at the same time, endless security issues also follow. Mobile cloud applications move the computing power and data storage away from mobile devices and into the cloud, which enable users to access network services anywhere and anytime [4, 10]. Furthermore, MCC provides simple and easy infrastructure for mobile applications and services, and it enables users to utilize resources on demand, and take full advantage of cloud computing services. However, due to its distributed nature and easy to use, MCC faces many technical challenges such as privacy, security, and so on.

To counter security issues in MCC like intruders or cyber-attacks, it is necessary to detect those attacks earlier by implementing immediate countermeasures to prevent the harmful risks [8]. Based on the existing security issues solutions, there have been two most useful techniques to defend mobile cloud services against intruders, such as firewall technology and IDS. In the research of cloud environment intrusion detection problems, many researchers have been using mainly four approaches based IDS such as clustering, classification, information theory, and statistical theories to deal with intrusion detection problems [8, 21]. Most recently, researchers have adopted different approaches like deep learning [19], Naïve Bayes [7], neural network [5, 25, 29], SVM [5, 9, 15, 16, 19, 20], genetic algorithm (GA) [13, 15], *etc.*

Meanwhile, The KDD'99 and NSL-KDD datasets have been used by many researchers to survey and evaluate research in intrusion detection. The KDD'99 dataset has not only been the most useful dataset in IDS, but also a benchmark for evaluating the best performance of intrusion detection methods [2, 19]. On the other hand, the NSL-KDD dataset is a new version of the KDD dataset, which has some advantages over the original KDD'99 dataset such as no redundant records in the training set and duplicate records in the test set of the NSL-KDD dataset [3]. After data collection, most of the datasets require feature analysis and dimensionality reduction to extract and select the data that is most likely to produce accurate results and reduce the computing cost and timing cost of the IDS [8,19]. The most recent feature analysis and dimension reduction methods used in cloud computing include principal component analysis (PCA) [19], information gain (IG) [6, 12, 25], genetic algorithm (GA)

based feature selection [15], *etc.*

However, many researchers have used different intrusion detection techniques to provide security for both mobile computing and networks, but still, have the common limitations on low detection accuracy, low true positive rate and high false positive rate. Motivated by this above, this paper proposed an intrusion detection method based on SVM and IG approach, which detects different malicious attacks with high detection accuracy and low false positive rate.

The remainder of this paper is organized as follows: Section 2 presents the recent related works. The problem statement and preliminaries of MCC and other related theories are explained in Section 3. The proposed intrusion detection method of MCC using SVM-IG is presented in Section 4. Section 5 gives the experimental results and performance analysis as compared with other related methods. Finally, we conclude our paper in Section 6.

2 Related Works

Currently, with the rapid development of cloud computing environment, the cloud security has become a serious challenge, and many researchers have adopted different techniques and methods such as machine learning techniques and data mining to improve the capability of IDS [5,29]. Among those techniques, artificial neural network (ANN) and SVM are the most useful methods in the cloud computing area [12, 19, 25, 29].

In [4, 18], the detailed surveys of data security in the MCC are discussed. Li *et al.* [16] proposed an IDS model based on rough set theory (RST) and fuzzy SVM (FSVM), the proposed method uses RST to reduce the dimensions of features, and the experimental results show that the proposed RST-FSVM can do better for IDSs. Hoque *et al.* [13] proposed an IDS model based on GA that filter and reduce the complexity of data; by using KDD'99 dataset, the reasonable detection rate has been achieved but got a slightly high false positive rate. Kannan *et al.* [15] proposed an intrusion detection model that combines genetic based feature selection and FSVM to secure the cloud networks. The proposed genetic based feature selection improved the detection accuracy of the FSVM classifier by selecting the relevant attributes in the KDD'99 dataset.

Zhang *et al.* [27] proposed an intrusion detection method based on cloud model and semi-supervised clustering, and the simulation results show that the performance of intrusion detection method has improved. Hoz *et al.* [14] proposed a network anomaly classification using support vector classifier and non-linear projection techniques; the experimental results show that the reasonable true positive rate has been achieved using NSL-KDD dataset, but has a high false positive rate. Deshmukh *et al.* [7] proposed an IDS model based preprocessing methods and Naïve Bayes classifier; The experimental results show that after applying preprocessing methods including discretization, normalization and feature selection using NSL-KDD dataset, the proposed method effectively improved the performance of IDS.

Pervez *et al.* [20] proposed a feature selection and SVM classifier using NSL-KDD dataset. Eesa *et al.* [9] proposed a new feature selection based on cuttlefish optimization algorithm (CFA) and the decision tree classifier, and by using the KDD'99 dataset, the results show that the proposed approach gives a high accuracy and detection rate with lower false positive rate compared with the results using all 41 features.

Yuan *et al.* [26] proposed a semi-supervised AdaBoost algorithm for network anomaly detection, and the experimental results show that the proposed method can achieve a good result even with a small labeled dataset. Nguyen *et al.* [19] proposed a deep learning approach that detects cyber-attacks in MCC, in this framework, PCA was used for the feature extraction and dimension reduction, and by using KDD'99, NSL-KDD and UNSW-NB-15 datasets, a good accuracy, and detection rate have been achieved, but they do not evaluate the false positive rate.

Ashfaq *et al.* [3] proposed a fuzziness based semi-supervised learning approach using neural network with random weights (NNRw) as a classifier, and through the experiments, NSL-KDDTest+ and NSL-KDDTest-21 are used to test the proposed method. Hammami *et al.* [12] proposed a cloud computing based IDS and human-immune system, the NSL-KDD dataset is used to evaluate the performance of the proposed method.

Zhao *et al.* [29] proposed an intrusion detection approach based SOM neural network in cloud computing, where the particle swarm optimization (PSO) based on simulated annealing was used to optimize the SOM neural network, and the experimental results showed that the PSO algorithm has effectively improved the performance of the system.

In [17, 27], the approaches of network intrusion detection based PCA using SVM were proposed, and PCA was used to reduce the higher dimensional KDD dataset to lower dimensional dataset, and the experimental results showed that PCA has effectively improved the performance of the IDS compared to without using PCA.

Wang [25] proposed an IDS for cloud computing using MLP and K-means algorithm. Information gain, which is a feature selection method, was used to select the most relevant features and remove unneeded features in the KDD'99 dataset. The simulation results show that the proposed approach has good performance compared to each of MLP and K-means respectively.

Aghdam *et al.* [1] proposed a feature selection for IDS using Ant Colony Optimization that identifies important features and improves the performance of IDS. The experimental results on the KDD Cup 99 and NSL-KDD datasets show that the proposed method provides high accuracy and low false positive rate in detecting intrusions.

3 Problem Statement and Preliminaries

3.1 Intrusion Detection System (IDS)

IDS is defined as a software application or a security management tool that controls all events occurring in a computing system or network and analyzing them to find the intrusions or malicious activities either within the system or outside the system [10, 20]. Intrusion is defined as the attempts that are used to compromise the data integrity, confidentiality, and data availability in a computing system [13].

Considering the methods of data collection, there are two distinct types of IDSs: "Host-based IDS" and "Network-based IDS" and considering the detection techniques of intrusions, there are two approaches to detect intrusions: "Misuse or signature detection based IDS" and "Anomaly detection based IDS" [10, 20]. In misuse detection based, the intrusions are detected by searching for activities that correspond to known attacks; While in anomaly detection based, the intrusions are detected by searching for deviations from a normal behavior model.

3.2 Mobile Cloud Computing (MCC)

MCC is a fast-growing architecture, which integrates mobile computing and wireless technology with cloud computing, where the mobile users utilize different cloud services anytime and anywhere based on the pay-as-you-use principle [10, 18]. The cloud computing provides various services such as Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS) to mobile computing in order to tackle the lack of enough storage space and processing power in mobile devices [11]. Although the MCC may seem to be a very exciting technology nowadays, there still remains some technical challenges, more specifically the security of data or information stored in the cloud [4, 11].

Figure 1 shows the architecture of mobile cloud computing.

3.3 Support Vector Machine (SVM)

The SVM is a supervised machine learning method that is used in data mining to analyze data and recognize patterns in the dataset for regression and classification purpose [20]. Nowadays, SVMs are used for linear and nonlinear classifications and support both binary and multiclass classifications. The datasets used in IDS are often high dimensional and heterogeneous. Because the traditional SVMs cannot directly deal with heterogeneous datasets, there is a need to use some kernel functions like Radial Bias Function (RBF), Linear Function, *etc.* in order to extend them on heterogeneous datasets [20]. In Binary classification, the SVM finds for a maximum margin hyperplane to separate the two classes, a class of positive samples and class of negative samples of the training set based on structural risk minimization analysis of statistical learning theory [23].

Let us assume that our binary classification has an n-dimensional feature space of a training set as follows:

$$T_s = \{(x_1, y_1), \ldots, (x_m, y_m)\} \in (R^n \times \{-1, +1\})^m, \quad (1)$$

where $x_i \in ([x_i]_1, [x_i]_2, \ldots, [x_i]_n)^{T_s}$ is the input feature vectors, $y_i \in \{-1, +1\}$ is the binary output of x_i, m is the number of samples in the feature space and R^n an n-dimensional real space.

The classification function of SVM is defined as follows:

$$f(x) = w^{T_s} \cdot x + b, \quad (2)$$

where b is the bias and w is a weight vector.

Thus, the training set should satisfy the following condition:

$$f(x) = \begin{cases} w^{T_s} \cdot x_i + b \geq -1, & \text{for all attack data } x_i \\ w^{T_s} \cdot x_i + b \leq +1, & \text{for all normal data } x_i \end{cases}$$

Figure 2 demonstrates the maximum-margin hyperplane and margins for an SVM classifier trained with samples from two classes either normal or malicious attack.

The main goal of SVM is to find the optimal hyperplane by maximizing the margin between two classes as can be seen in Figure 2. The distance between two hyperplanes is $\frac{2}{\|w\|}$, the optimization objective is just to minimize $||w||$, and this can be obtained by solving the following quadratic optimization problem:

$$\begin{cases} \textit{minimize } (\textit{in } \text{w,b}) \ \frac{1}{2}||w|| \\ \textit{subject to} : y_i(\langle w \cdot x_i \rangle + b) \quad \text{for } i\text{=1,2,...,m} \end{cases} \quad (3)$$

The quadratic optimization problem in Equation (3) can be solved by the sequential minimal optimization (SMO) algorithm using the Lagrange multipliers a_i as follows:

$$\begin{cases} \textit{maximize } L(\alpha) = \\ \qquad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{j=1}^{m} \sum_{i=1}^{m} y_i y_j \alpha_i \alpha_j K(x_i \cdot y_j) \\ \textit{subject to} : \sum_{i=1}^{m} y_i \alpha_i = 0, 0 \leq \alpha_i \leq C, 1 \leq i \leq m \end{cases} \quad (4)$$

where L is the Lagrange function, C is the regularization parameter and K is the kernel function.

After solving Equation (4), we obtain $w = \sum_{i=1}^{m} y_i \alpha_i x_i$, and the decision attack function is defined as follows:

$$f(x, \alpha, b) = \{\pm 1\} = sgn(\sum_{i=1}^{m} y_i \alpha_i K(x, x_i) + b), \quad (5)$$

where b is obtained from Karush-Kuhn-Tucker condition.

To construct SVM classifier, a kernel function and some parameters have to be selected. There are three main types of SVM kernel function: Linear kernel function, Radial Bias kernel function (RBF) and polynomial kernel

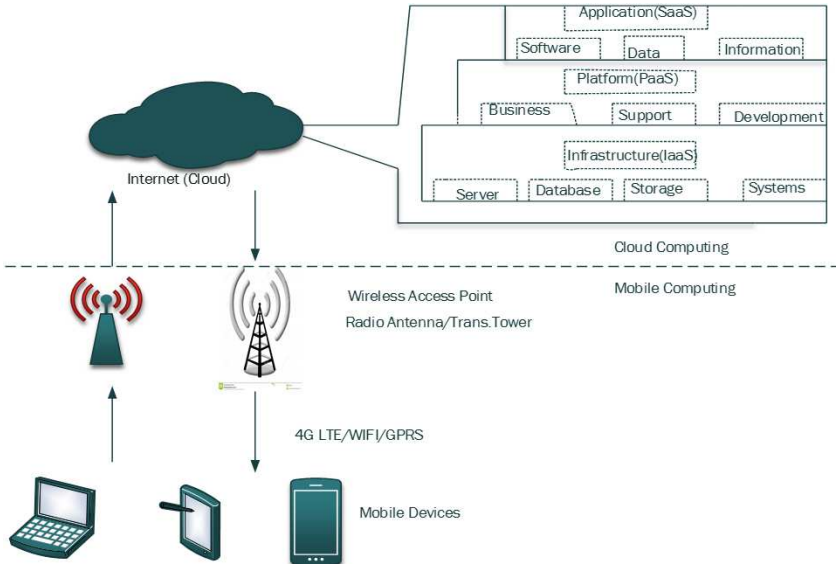


Figure 1: Architecture of mobile cloud computing

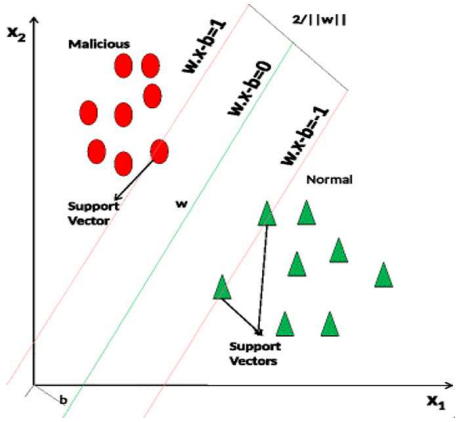


Figure 2: Maximum separating the hyperplane and the margin

function. It is very advantageous to use SVM classifier as it has the ability to give very accurate results, specifically for binary classification, and it is very effective in high dimensional datasets. In addition to that, the SVM classifier is very robust against overfitting and outliers.

3.4 Information Gain Based Feature Selection (IG)

The datasets, which are mostly used for analyzing IDSs like KDD'99 and NSL-KDD datasets, have been facing a serious problem of large amount of records including redundant and irrelevant records, as well as relevant records. To counter this above problem, many researchers have adopted various methods like dimensionality reduction, feature selection and so on. According to [1], feature selection is a task of identifying the relevant features from all features and removing the irrelevant or inappropriate ones in the dataset. In this paper, we have adopted IG based feature selection to reduce the computation complexity of the dataset.

According to [6], IG is a method used to decide which attribute in a given dataset is most important to be used in the machine learning process for classifying data. The IG uses Shannon's entropy to measure the feature set quality.

Let's consider S to be a set of training set samples of any dataset. Suppose that there are n classes and the training set contains s_i samples of class I and s is the total number of samples in the training set. Then the expected information needed to classify a given sample is solved as follows:

$$I(s_1, s_2, \ldots, s_n) = \sum_{i=1}^{n} \frac{s_i}{s} \log 2 \left(\frac{s_i}{s}\right) \tag{6}$$

The training set S can be divided into v subsets S_1, S_2, ..., S_v by an attribute A with values a_1, a_2, ..., a_v, where S_j is the training subset, which has the value a_j for attribute A. Additionally, let S_j contains s_{ij} samples of class I. The Entropy of the attribute A is as follows:

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + s_{2j} + \cdots + s_{nj}}{s} \times I(s_{1j}, \ldots, s_{nj}). \tag{7}$$

Therefore, the IG for attribute or feature A can be calculated as follows:

$$Gain(A) = I(s_1, \ldots, s_n) - E(A). \tag{8}$$

3.5 Dataset Description

The datasets are used to survey and evaluate a research in intrusion detection, some are self-created datasets, and others are publicly available. Over the last years, most researchers have adopted KDD'99 Cup and NSL-KDD datasets, which are publicly available to train and test the performance of an intrusion detection research [7,11].

3.5.1 KDD'99 Cup Dataset

The KDD'99 Cup is a version of DARPA 1998 dataset, which has been provided by MIT Lincoln laboratory in 1999. The complete KDD'99 dataset has up to 5 million input records, and every record represents a TCP/IP connection that consists of 41 features (3 of them are symbolic, and 38 are numeric) and one marked as either normal or attack, and all attacks fall into four major categories: Probe, DoS, U2R and R2L [1, 7, 11]. Because of this large amount of records in the KDD'99 dataset, it has been grouped into three independent subsets: "10% KDD", "Corrected KDD" and "Whole KDD" [11]. Most of the researchers prefer to use 10% KDD for training set and corrected KDD for test set.

3.5.2 NSL-KDD Dataset

The NSL-KDD dataset is considered as a reduced version of KDD'99 Cup, where it overcomes the problem of redundant records existing in KDD'99 Cup training set, the size of the dataset is reduced compared to that of KDD'99, and has no duplicate data in the improved test set [2]. As for the KDD'99 dataset, NSL-KDD dataset also has 41 features and one marked as either normal or attack [11,28]. Apart from having several advantages over KDD'99, the NSL-KDD is still not yet a perfect representative for existing real networks compared to the original KDD'99 dataset, due to the lack of public datasets for network-based IDSs, but it still can be used as an effective benchmark dataset to compare different intrusion detection methods [1, 11].

Table 1 describes the number of records in KDD'99 (10% KDD'99 for training and corrected KDD for testing) and NSL-KDD datasets.

4 The Proposed Method

The flowchart of the intrusion detection method based on SVM and IG for MCC is depicted in Figure 3. The proposed method is divided into two phases namely; data preparing phase and intrusion detection phase.

The data-preparing phase has two main functions, *i.e.*, data collection from the KDD'99 and NSL-KDD datasets and data preprocessing which is divided into three parts: "Data discretization", "Feature selection" and "Data normalization".

In data preparing phase, the packet features collected from KDD'99 and NSL-KDD datasets are first discretized where not all the 41 features of KDD datasets are continuous or discrete values. The features like protocol type (TCP, UDP and ICMP), network service need to be converted into numbers. Among the 41 features, some are irrelevant or redundant leading to a long detection process and degrading the system's performance. Therefore, selecting the most relevant features is an essential way to increase the performance and reduce the computing and timing cost; here the IG based feature selection is used. Thus, the data normalization part follows, in order to scale the data in a specific range; we adopted the min-max normalization method.

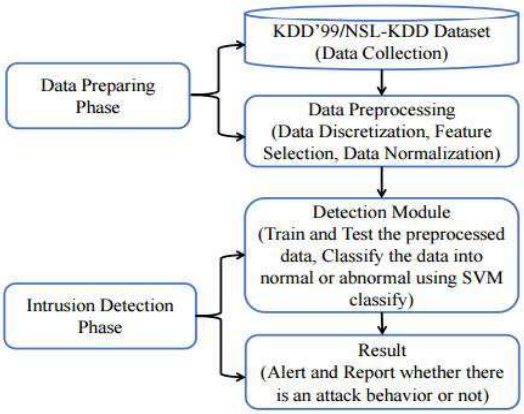


Figure 3: Intrusion detection method of MCC based on SVM and IG

In intrusion detection phase, the selected features from the data preprocessing part are trained, tested and then classified into normal or attack by using the SVM classifier. Therefore, the final result will be reported to the system administrator, and if there is an attack, the system administrator will deal with it accordingly; otherwise, the packet feature will be served as normal.

The proposed SVM-IG algorithm in this paper is defined as follows:

Algorithm 1 SVM-IG

1: Input: The KDD'99 and NSL-KDD training and test data
2: Output: The evaluation metrics of the proposed method (ACC, TPR, PPV, and FPR)
3: Obtain the input data
4: **while** training data **do**
5: Preprocessing of data
6: Consider the RBF kernel function
7: Use the cross-validation to find the best C and gamma (γ) parameters
8: Use the best parameters C and γ to train the whole training set
9: **end while**
10: **while** test data **do**
11: Preprocessing of data
12: Evaluate and predict the output
13: **end while**

5 Experimental Results and Analysis

All the experiments are performed on a Compaq-HP computer with 2.4 GHz Intel (R) Core (TM) i3-3110M

Table 1: KDD'99 and NSL-KDD training and test dataset records

Name	Dataset	Records	Normal	Probe	DoS	U2R	R2L
KDD'99 (10%)	Train	494021	97278	4107	391458	52	1126
Corrected KDD	Test	311029	60593	4166	229853	228	16189
NSL-KDD	$Train+\_20$	25192	13449	2289	9234	11	209
NSL-KDD	Test-21	22544	9711	2421	7458	200	2754

and 10 GB of RAM, and running on windows 10 Enterprise (64bits). The proposed method was implemented in MATLAB R2018b and Weka 3.8.3 data mining tool. The SVM classifier is applied with LibSVM package (MATLAB version 3.23).

5.1 Dataset and Data Preprocessing

In the evaluation process of the proposed method, we have used 10% of the full KDD'99, corrected KDD, NSL-KDD Train+_20 and NSL-KDD Test-21 datasets for training and testing our model as shown in Table 1. All these KDD datasets contain 41 features and one marked as either normal (1) or attack (0). The attacks fall into four major types: Denial of Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R).

In data preprocessing, we adopted the Weka tool to perform the data discretization and feature selection. In data discretization, the continuous features are converted to discrete or nominal features by using the Weka discretization filter and InfoGainAttributeEval with Ranker available in Weka tool is used to select the most important features. Table 2 shows the most relevant features selected by using IG based feature selection.

Through the feature selection analysis, the features {20, 21} for both KDD'99 and NSL-KDD datasets show zero information gain which means they do not contribute to the intrusion detection. The features {5, 6, 7, 9, 10, 11, 13, 14, 15, 16, 17, 18} for NSL-KDD dataset and {5, 6, 7, 9, 13, 14, 15, 16, 17, 18} for KDD'99 dataset have a very small information gain, which has a little effect to the intrusion detection. The stated above features are removed from the datasets due to the small contribution on the intrusion detection. Therefore, by using the IG based feature selection, the most relevant features used in our method for both KDD'99 and NSL-KDD datasets are shown in Table 2 and the dimension of both datasets was reduced as well. Furthermore, the features like protocol type and network service cannot be sent directly to the system, and hence they need to be preprocessed and converted into numerical digits. For example, protocol types like TCP, UDP, and ICMP are converted to number 1, 2 and 3 the same as for other network services. Therefore, the numerical values are scaled within a specified range. In the proposed method, we scaled the numerical values within a range of [0, 1] by using the min-max normalization method.

$$f(x) = \frac{(x - x_{min})}{(x_{max} - x_{min})} \in [0,1]; x \in [x_{min}, x_{max}] \quad (9)$$

where x is an attribute value, x_{min} is the minimum attribute value, x_{max} is the maximum attribute value, and $f(x)$ is the normalized value.

5.2 Performance Metrics

In order to measure the performance of the MCC based IDS, the true positive rate (TPR), false positive rate (FPR), accuracy (ACC), and precision (PPV) indicators are used for measurement. A confusion matrix is used to represent the information related to the actual and predicted classifications performed by the classification system.

The confusion matrix is shown in Table 3. In Table 3, TP indicates that the actual is a normal sample and is predicted as the number of normal samples, FN indicates that the actual is a normal sample and is predicted as the number of abnormal samples, FP indicates that the actual is an abnormal sample and is predicted as the number of normal samples, and TN indicates that the actual is an abnormal sample and is predicted as the number of normal samples.

The ACC, PPV, TPR, and FPR are the four main performance metrics used for the proposed method as described below:

$$\begin{aligned} ACC &= \frac{TP+TN}{TP+FN+FP+TN} \\ PPV &= \frac{TP}{TP+FP} \\ TPR &= \frac{TP}{TP+FN} \\ FPR &= \frac{FP}{FP+TN} \end{aligned}$$

where ACC shows a total number of corrected predictions, PPV indicates that the intrusion predicted by IDS is an actual intrusion, TPR determines the correctly identified positive instances, FPR indicates the normal cases that incorrectly identified as an anomaly.

5.3 Performance Evaluation

After the data preprocessing, the reduced data is fed into the model and processed via LibSVM, an open source for SVM Classifier and Radial Bias Kernel Function (RBF Kernel) which has two hyperparameters C and Gamma (γ), was used to study the effectiveness of the SVM classifier. The parameters C and Gamma (γ) were tuned to find the better cross-validation (Cross Val) accuracy by

Table 2: List of most relevant features in KDD'99 and NSL-KDD dataset using IG

KDD'99 dataset		NSL-KDD dataset	
No.	Selected Features	No.	Selected Features
2	protocol type	3	service
3	service	4	flag
4	flag	12	*logged_in*
23	count	23	count
24	*srv_count*	25	*serror_rate*
25	*serror_rate*	26	*srv_serror_rate*
26	*srv_serror_rate*	29	*same_srv_rate*
29	*same_srv_rate*	32	*dst_host_count*
33	*dst_host_srv_count*	33	*dst_host_srv_count*
34	*dst_host_serror_rate*	34	*dst_host_same_srv_rate*
36	*dst_host_rerror_rate*	38	*dst_host_serror_rate*
38	*dst_host_same_srv_rate*	39	*dst_host_srv_serror_rate*
39	*dst_host_diff_srv_rate*		

Table 3: Confusion Matrix

		Predicted	
		Attack	Normal
Actual	Attack	TP	FN
	Normal	FP	TN

using grid search method and the ones with good Cross Val accuracies were picked and used to train and validate the proposed method as can be seen from the Table 4, Table 5 and Figure 4, Figure 5, Figure 6 and Figure 7 below.

In the proposed method, we have used 10-fold Cross Val to tackle the overfitting problem, which divides the dataset into 10 sub-sets of size N/10 (N is the size number of the dataset) and uses 9 sub-sets for training and 1 remaining sub-set for testing. The practical way to find better parameters of C and γ is to try the exponential growing sequences of them by using coarse and fine grid-search methods. The grid-options such as $log_2 C$ and $log_2 \gamma$ are used to run the SVM classifier for a certain range of C and γ parameters. Figure 4 and Figure 5 shows the coarse grid-search and fine grid-search for the KDD'99 dataset.

For KDD'99 dataset, we conducted the coarse grid-search on $\log_2 C \in [-5, 15]$ and $\log_2 \gamma \in [-14, 2]$, Figure 4 shows the coarse grid-search with an exponential growing sequence of C and γ ($C = 2^{-5}, 2^{-4}, \ldots, 2^{14}, 2^{15}; \gamma = 2^{-14}, 2^{-13}, \ldots, 2^{3}, 2^{2}$), which gives us the best parameters with the Cross Val accuracy of 99%.

In Figure 5, the searching range was reduced to $\log_2 C \in [-4, 8]$ and $\log_2 \gamma \in [-7, 3]$ and the fine grid-search was conducted with an exponential growing sequence of C and γ ($C = 2^{-4}, 2^{-3}, \ldots, 2^{7}, 2^{8}; \gamma = 2^{-7}, 2^{-6}, \ldots, 2^{4}, 2^{3}$), the best parameters were obtained with the Cross Val accuracy of 99%. To find the best parameters of C and γ, several grid-searches were executed, and several high Cross Val accuracies were obtained.

Table 4 shows the best parameters with their Cross Val accuracies, their classification accuracy (ACC), precision (PPV), true positive rate (TPR), and false positive rate

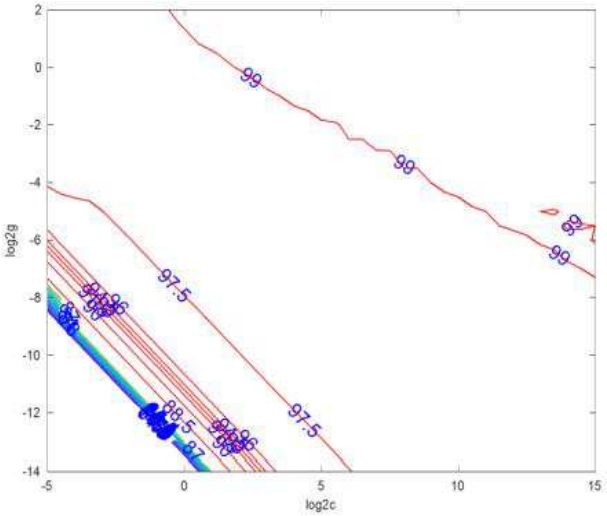


Figure 4: The coarse grid-search on $\log_2 C \in [-5, 15]$ and $\log_2 \gamma \in [-14, 2]$ for the KDD'99 dataset

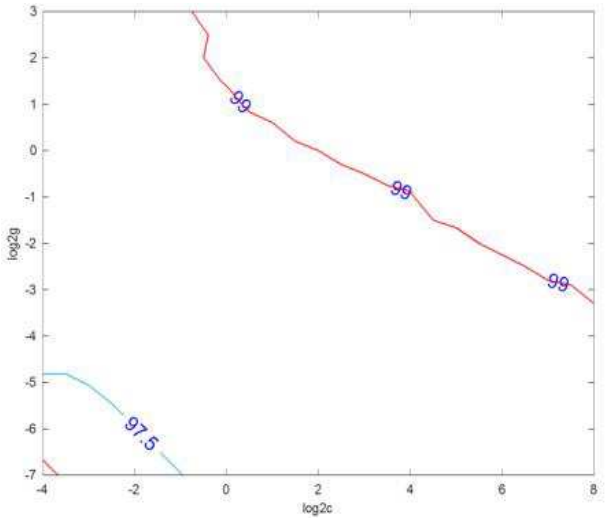


Figure 5: The fine grid-search on $\log_2 C \in [-4, 8]$ and $\log_2 \gamma \in [-7, 3]$ for the KDD'99 dataset

(FPR) for the KDD'99 dataset.

By using grid search methods mentioned above, the C and gamma (γ) parameters of RBF Kernel were tuned to select the ones with high Cross Val accuracy. As can be seen from the Table 5, the row with gamma (γ) =8, C= 1 is selected with Cross Val = 99.025%.

Figure 6 and Figure 7 displays the coarse grid-search and fine grid-search for NSL-KDD dataset.

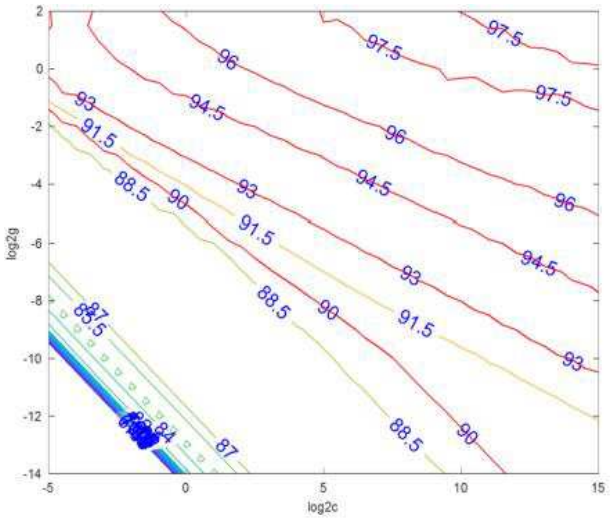


Figure 6: The coarse grid-search on $\log_2 C \in [-5, 15]$ and $\log_2 \gamma \in [-14, 2]$ for the NSL-KDD dataset

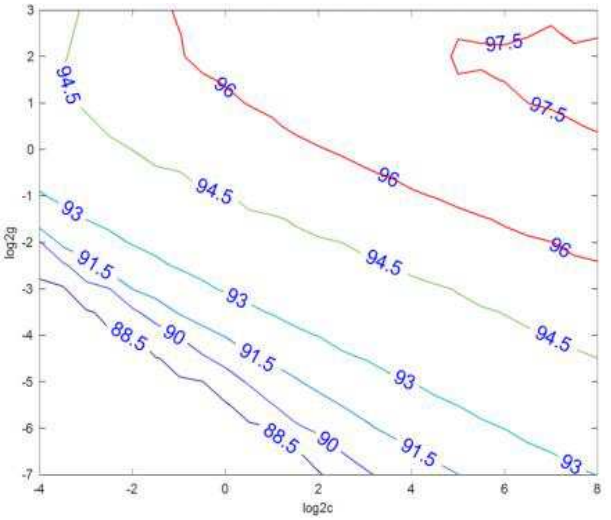


Figure 7: The fine grid-search on $\log_2 C \in [-4, 8]$ and $\log_2 \gamma \in [-7, 3]$ for the NSL-KDD dataset

For NSL-KDD dataset, the coarse grid-search was conducted in the range of $\log_2 C \in [-5, 15]$ and $\log_2 \gamma \in [-14, 2]$ as shown in Figure 6, the best parameters of C and (γ) were obtained with the Cross Val accuracy of 97.5%. Figure 7 shows the fine grid-search in the range of $\log_2 C \in [-4, 8]$ and $\log_2 \gamma \in [-7, 3]$, the best parameters were obtained with the Cross Val accuracy of 97.5%. To find the best parameters of C and (γ), we have conducted several grid-searches, and several high Cross Val accuracies were obtained. Table 5 shows the best parameters with their best Cross Val, their best classification ACC, PPV, TPR, and FPR for the NSL-KDD dataset.

As explained in Table 5, the same grid search method was performed to tune the RBF Kernel parameters, and the one with high Cross Val accuracy is picked, and as can be seen from Table 5, the row with γ=1 and C=2 is the one with the highest Cross Val = 96.6246%.

5.4 Experimental Result and Discussion

The KDD'99 Cup and NSL-KDD datasets are adopted during the experiments to evaluate the performance of the proposed method, and the performance comparison of different intrusion detection methods using KDD'99 and NSL-KDD dataset is shown in Table 6 and Table 7.

As can be seen from Table 6 and Table 7, through comparing to other research methods, the proposed approach has improved with good performance of PPV, TPR, and FPR.

1) With KDD'99 Cup dataset: The proposed method (SVM-IG) has an accuracy, which is smaller than that of Deep learning [19] and GFS-FSVM [15], but greater than the ones for RST-FSVM [16], GA-IDS [13] and SVM [9]. As can be seen from Table 6, Deep learning [19] and GFS-FSVM [15] have good ACC comparing to the proposed method, but the proposed method has good FPR compared to that of GFS-FSVM [15] and has good TPR and PPV compared to that of Deep learning [19].

   The Precision of the proposed method is good compared to other approaches except for SSA [26], but TPR of SSA [26] is slightly small compared to that of the proposed method. In addition to that, the TPR of the proposed method is good compared to other algorithms, except for GA-IDS [13], which has a slightly high FPR and small ACC and PPV compared to that of the proposed method. As can be seen from Table 6, the FPR of the proposed method is also better than the other approaches, except for SSA [26], which has a small TPR compared to the proposed method.

2) With NSL-KDD dataset: As can be seen from Table 7, the accuracy of the proposed method (SVM-IG),ACC=0.8650 is higher than that of SVM [20] and NNRw [3], but smaller than SVC [14], Naïve Bayes [7], and Deep learning [19]. On the other hand, SVC [14] and Naïve Bayes [7] have high FPR compared to that of the proposed method. As mentioned above, Deep learning [19] has good accuracy, but its PPV and TPR are smaller than that of the proposed method.

   The precision of the proposed method is better than that of SVM [19, 20] and Deep learning [19] but

Table 4: Performance of the proposed method using the Grid Search method for the KDD'99 dataset

Parameter Selection		ACC	PPV	TPR	FPR	CrossVal(%)
γ=1	C=32	0.9506	0.9647	0.9460	0.0247	98.925
γ=2	C=8	0.9507	0.9605	0.9464	0.0260	98.9
γ=2	C=16	0.9509	0.9515	0.9466	0.0253	98.95
γ=8	C=1	0.9523	0.9675	0.9489	0.0298	99.025
γ=8	C=2	0.9515	0.9624	0.9483	0.0314	98.925
γ=8	C=4	0.9510	0.9688	0.9478	0.0317	99

Table 5: Performance of the proposed method using the Grid Search method for the NSL-KDD dataset

Parameter Selection		ACC	PPV	TPR	FPR	CrossVal(%)
γ=0.25	C=32	0.8545	0.8878	0.7623	0.0657	96.0384
γ=0.5	C=4	0.8674	0.8813	0.7691	0.0651	96.5267
γ=0.5	C=8	0.8646	0.8877	0.7626	0.0655	96.6056
γ=1	C=1	0.8629	0.8781	0.7620	0.0702	96.3798
γ=1	C=2	0.8676	0.8878	0.7665	0.0646	96.6246
γ=4	C=32	0.8622	0.8757	0.7690	0.0664	96.6021

Table 6: Performance comparison of different intrusion detection methods using KDD'99 Cup

Methods	KDD'99 Cup			
	ACC	PPV	TPR	FPR
RST-FSVM [16]	0.9000	-	0.8576	0.1424
GA-IDS [13]	0.9004	0.9280	0.9500	0.3046
SVM [9]	0.9198	0.7400	0.8200	0.0391
DWIDM-CM SSC [27]	-	-	0.8989	0.0800
SSA [26]	-	0.9863	0.8902	0.0138
Deep Learning [19]	0.9711	0.9443	0.9277	-
GFS-FSVM [15]	0.9857	-	-	$\geq$0.0400
Proposed method (SVM-IG)	0.9523	0.9675	0.9489	0.0298

Table 7: Performance comparison of different intrusion detection methods using NSL-KDD

Methods	NSL-KDD			
	ACC	PPV	TPR	FPR
Naive Bayes [7]	0.9010	0.8900	0.9360	0.1340
SVC [14]	0.8970	-	0.9340	0.1400
SVM [20]	0.8350	0.7400	0.8200	0.1500
SVM [19]	0.8832	0.6470	0.7080	-
Deep Learning [19]	0.09099	0.8195	0.7748	-
NNRw [3]	0.8412	-	-	-
Human Immune System [12]	-	-	0.9860	0.0800
Proposed method (SVM-IG)	0.8676	0.8878	0.7665	0.0646

smaller than that of Naïve Bayes [7], which has a large FPR compared to that of the proposed method. In [7, 12, 14, 20], their TPR are better than that of the proposed method, but not good in terms of FPR compared to the proposed method.

Moreover, the experimental results prove that the proposed method can increase the training speed and shorten the training time cost with the elapsed time of 132.646993s for the KDD'99 dataset and 7.897525s for the NSL-KDD dataset, which is good compared to that of [24].

6 Conclusions

Over the last decades, many artificial intelligence algorithms have been applied to improve the performance of intrusion detection system (IDS). Among these algorithms, SVM is one of the most widely used and has a relatively high performance, and the performance of IDS is highly dependent on the quality of training data. In this paper, we proposed the intrusion detection method based on SVM and IG to detect cyber-attacks in MCC. The SVM classifier is used for binary classification to analyze and classify data in either normal or abnormal behavior, and the IG is used to select the most relevant features in KDD'99 and NSL-KDD dataset. Through the experimental results, we have shown that the proposed method has

good scalability and high training speed, and can detect malicious attacks with high accuracy, high detection rate, and low false positive rate.

For the future research, we will implement this method using multi-class classification and evaluate the performance on a real time basis.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61862041), the Research Project in Universities of Education Department of Gansu Province (2017B-16, 2018A-187). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] M. H. Aghdam and P. Kabiri, "Feature selection for intrusion detection system using ant colony optimization," *International Journal of Network Security*, vol. 18, no. 3, pp. 420–432, 2016.

[2] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[3] R. A. R. Ashfaq, X. Z. Wang, and J. Z. Huang, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484–497, 2017.

[4] T. Bhatia and A. K. Verma, "Data security in mobile cloud computing paradigm: a survey, taxonomy and open research issues," *Journal of Supercomputing*, vol. 73, no. 6, pp. 1–74, 2017.

[5] A. N. Cahyo, R. Hidayat, and D. Adhipta, "Performance comparison of intrusion detection system based anomaly detection using artificial neural network and support vector machine," in *in Advances of Science and Technology for Society: Proceedings of the International Conference on Science and Technology*, vol. 1755, pp. 070011, 2016.

[6] L. Dali, K. Mivule, and H. El-Sayed, "A heuristic attack detection approach using the east weighted attributes for cyber security data," in *IEEE in Intelligent Systems Conference (IntelliSys'17)*, pp. 1067–1073, 2017.

[7] H. D. Deshmukh, T. Ghorpade, and P. Padiya, "Intrusion detection system by improved preprocessing methods and naïve bayes classifier using nsl-kdd 99 dataset," in *International Conference on Electronics and Communication Systems (ICECS'14)*, pp. 1–7, 2014.

[8] R. H. Dong, D. F. Wu, and Q. Y. Zhang, "Mutual information-based intrusion detection model for industrial internet," *International Journal of Network Security*, vol. 20, no. 1, pp. 131–140, 2018.

[9] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2670–2679, 2015.

[10] K. Gai, M. Qiu, and L. Tao, "Intrusion detection techniques for mobile cloud computing in heterogeneous 5G," *Security and Communication Networks*, vol. 9, no. 16, pp. 3049–3058, 2016.

[11] Y. Hamid, V. R. Balasaraswathi, L. Journaux, and M. Sugumaran, "Benchmark datasets for network intrusion detection: A review," *International Journal of Network Security*, vol. 20, no. 4, 2018.

[12] H. Hammami, H. Brahmi, and S. B.Yahia, "Security insurance of cloud computing services through cross roads of human-immune and intrusion-detection systems," in *International Conference on Information Networking (ICOIN'18)*, pp. 174–181, 2018.

[13] M. S. Hoque, M. Mukit, and M. Bikas, "An implementation of intrusion detection system using genetic algorithm," *International Journal of Network Security & Its Applications*, vol. 4, no. 2, pp. 109–120, 2012.

[14] E. D. L. Hoz, A. Ortiz, and J. Ortega, "Network anomaly classification by support vector classi?ers ensemble and non-linear projection techniques," in *International Conference on Hybrid Artificial Intelligence Systems*, pp. 103–111, Sep. 2013.

[15] A. Kannan, G. Q. Maguire, and A. Sharma, "Genetic algorithm based feature selection algorithm for effective intrusion detection in cloud networks," in *IEEE 12th International Conference on Data Mining Workshops (ICDMW'12)*, 2012. (`https://ieeexplore.ieee.org/document/6406470`)

[16] L. Li and K. N. Zhao, "A new intrusion detection system based on rough set theory and fuzzy support vector machine," in *The 3rd International Workshop on Intelligent Systems and Applications (ISA'11)*, 2011. (`https://ieeexplore.ieee.org/document/5873410`)

[17] Q. I. M. Yu, M. Liu, and F. U. Y. Ming, "Research on network intrusion detection using support vector machines based on principal component analysis," *Netinfo Security (in Chinese)*, vol. 2, pp. 15–18, 2015.

[18] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way ahead," *Journal of Network & Computer Applications*, vol. 84, pp. 34–54, 2018.

[19] K. K. Nguyen, D. T. Hoang, and D. Niyato, "Cyberattack detection in mobile cloud computing: A deep learning approach," in *IEEE Wireless Communications and Networking Conference (WCNC'18)*, pp. 1–6, 2018.

[20] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," in *The 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA'14)*, pp. 1–6, 2014.

[21] Q. S. Qassim, A. M. Zin, and M. J. A. Aziz, "Anomalies classification approach for network-based intrusion detection system," *International Journal of Network Security*, vol. 18, no. 6, pp. 1159–1172, 2016.

[22] S. Rezaei, M. Ali Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115–131, 2018.

[23] I. S. Thaseen and C. A. Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017.

[24] H. W. Wang, J. Gu, and S. S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowledge-Based Systems*, vol. 136, pp. 130–139, 2017.

[25] Z. Wang, "Using neural networks in intrusion detection system for cloud computing," *California State University San Marcos*, vol. 24, no. 3, pp. 579–588, 2014.

[26] Y. Yuan, G. Kaklamanos, and D. Hogrefe, "A novel semi-supervised adaboost technique for network anomaly detection," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 111–114, 2016.

[27] J. Zhang and Y. Z. Li, "Dynamic weighted intrusion detection method based on cloud model and semi-supervised clustering," *Journal of Kunming University of Science & Technology*, vol. 2013, no. 4, 2013.

[28] X. Zhang, P. Zhu, and J. Tian, "An effective semi-supervised model for intrusion detection using feature selection based LapSVM," in *International Conference on Computer, Information and Telecommunication Systems (CITS'17)*, pp. 283–286, 2017.

[29] J. Zhao and Y. Zhu, "Research on intrusion detection method based on som neural network in cloud environment," *Computer Science and Application*, vol. 6, no. 8, pp. 505–513, 2016.

Biography

Mugabo Emmanuel. He is currently pursuing his master's degree at Lanzhou University of Technology. He graduated with a bachelor degree in Electronic Science and Communication from University of Dar-es-salaam (Tanzania) in 2014. His main research focuses on the network and information security.

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

A Network Flow Correlation Method Based on Chaos Theory and Principal Component Analysis

Yang Chen and Yonghong Chen
(Corresponding author: Yang Chen)

College of Computer Science, Technology, Huaqiao University
No. 668 Jimei Avenue, Xiamen, Fujian, China 361021
(Email: 13476863090@163.com)

(Received Sept. 18, 2018; Revised and Accepted Mar. 23, 2019; First Online June 9, 2019)

Abstract

Detection of the related incoming and outgoing flows helps to expose the attackers hiding behind stepping stones. Currently, the network flow watermarking scheme is used for the detection of network flow correlation, due to the watermarking schemes introduce large delays to the target flows and often make it impossible to achieve robustness and invisibility. In this paper, we propose a novel flow correlation scheme based on Chaos Theory and Principal Component Analysis. In this method, the network traffic is preprocessed by phase space reconstruction of chaos. Then, traffic traits are extracted by Principal Component Analysis, which are used later to calculate similarity between sender and receiver based on cosine similarity. Experimental results show that the scheme can resist packet insertions, network jitter and losses.

Keywords: Chaos Theory; Flow Correlation; Network Security; Principal Component Analysis

1 Introduction

As the Internet is more and more used in various aspects of everyday life, people are realizing that computer systems are suffering more threats than ever before. Timely response and active defense has become an important guarantee to maintain the continuous dynamic network security. However, most network security mechanisms deal with these network attacks in a passive way. Intrusion detection system is an important part of network security. But current intrusion detection mechanisms are still difficult to effectively track and detect network attack sources [16]. In fact, because network attackers rarely launch attack through their own computers directly, they are more likely to hide their origin by connecting across multiple stepping stones [2, 13, 14, 19] or use anonymous communication systems (such as Tor [11]) before attacking the final targets, it makes intrusion tracing complex and difficult.

Currently, various network flow correlation methods have been proposed efficiently to link packet flows in a network in order to thwart various attacks such as stepping stones intrusion. Traditionally, passive network flow analysis methods [4, 8, 15, 20] have many shortcomings, such as poor real-time, high space costs, poor flexibility, low accuracy, and the inability to handle encrypted traffic, etc. Recently, the network flow watermarking provides a better way to track the intrusion source. However, the robustness and invisibility of network flow watermarks is very important, which are difficult to achieve at the same time [7]. This is because that the robustness requires the injected watermark always robust living in the network flow, while the invisibility prevents the active attackers to see the watermark in network flow. For instance, in the interval-based schemes, the duration of each intercepted flow is partitioned into short time intervals, and all packets within selected intervals are intentionally modulated to form a watermark pattern. Given that a few packets would not greatly affect the pattern created in the entire interval, these schemes are robust against network artifacts such as packet drops and inserts. However, one problem with such schemes is the lack of invisibility. Shifting packets in batches produce noticeable traces of the embedded watermarks, which can expose the watermark positions [5]. This enables the attackers to remove or modify the watermarks embedded in a network stream and even transfer them to another unrelated stream, which will make any linking techniques be meaningless.

In this paper, we proposed a new scheme for linking flows, which aims at designing a similarity degree to effectively link network flows without changing and forwarding the primitive traffic models [3]. Firstly, the embedding dimension and time delay of the network time series are calculated, then the chaotic phase space reconstruction is used to reconstruct the time series to obtain the space characteristics of the network flow. Second, we use the Principal Component Analysis (PCA) algorithm to extract the most important characteristics of the above obtained traffic. In brief, we compare our scheme with the

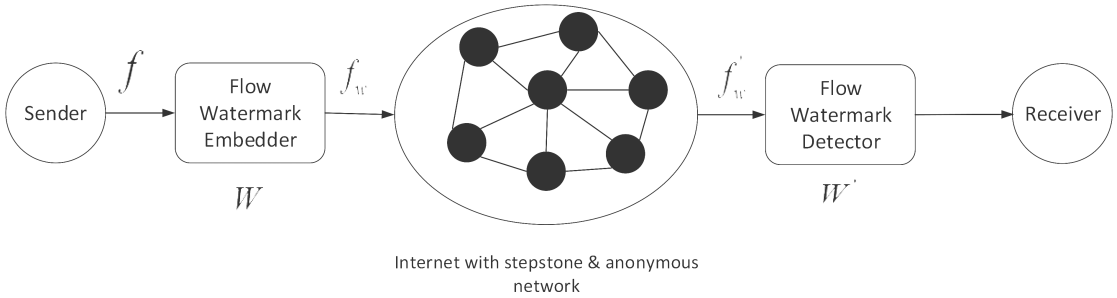


Figure 1: The universal model of network flow watermarking

classical existing methods through experiment and show that our approach can achieve better overall performance under network jitter, packet losses and insertions.

The rest of this paper is arranged as follows: Background on flow watermarking appears in Section II. In Section III we introduce our proposed intrusion detection scheme. Section IV presents the experimental results and discussion. This paper is concluded in Section V along with some future research directions.

2 Background

In this section, we review the problem of detecting stepping stones and then introduce the framework, universal model and typical characteristics of flow watermarks, respectively.

2.1 Stepping Stone Detection

Stepping stone attacks are a common way for network intruders to hide their identity. In a stepping-stone attack, the attacker compromises multiple hosts as relay machines, uses remote login such as Telnet or SSH to construct a chain of connections through these hosts, and then sends attacking commands to the victim through this chain [22]. Because each connection is made through a separate remote login, the next host in the chain can only see the identity of its immediate upstream neighbor, and the victim can only see the identity of the last host. Therefore, we must trace back the chain to find the origin of an attack.

2.2 Universal Model of Network Flow Watermarking

Flow watermarking technologies, embed watermark by changing or modulating traffic characteristics such as inter-packet delay (IPD) and interval centroid, at the sender side, nd the watermark will be identified and extracted at the receiver side to correlate the communication relationship of the sender and the receiver, as shown in Figure 1.

Figure 1 shows the universal model of network flow watermarking. The watermark embedder collects network traffic flow f, then selects a feature of the stream f (such as inter-packet delay (IPD), interval centroid, etc.) as the carrier of watermark w. Watermark detector captures the network traffic flow f_w and extract the watermark $w^{'}$, with watermarking detection algorithm, comparing w with $w^{'}$ to judge whether f_w is correlated with f.

In order to accurately trace the flow, flow watermarking technologies must have the following characteristics [24].

First of all, robustness is needed to ensure that watermark information survive to be correctly detected after malicious attacks or network transmission damages. Secondly, a successful watermark pattern should stay "invisible" to avoid possible attacks, if the intruder found that incoming flow is marked, he might command the stepping stone to take precautionary actions(for example, remove the watermarks).

At present, many flow watermarking approaches have been proposed. Houmansadr *et al.* [7] proposed an interval time-delay based watermarking scheme(RAINBOW), which first calculates the interpacket delays (lPDs) and saves them into the IPD database, and further increases or decreases the value of IPDs to embed the watermark information. In detection process, all IPDs are computed with the IPDs in the database to judge whether the watermark information existed. However, the demand and difficulty of network deployment have become much higher than before. In [21], ICBW method based on interval centroid was proposed, it embeds watermarking signals by adjusting the centroids of the intervals, However, its mechanism is built on a prerequisite that the interval centroid is stable when the count of packets is large enough.

Existing watermarking scheme has more or less modify the network traffic patterns, increasing the possibility of being discovered by the attacker.So this paper aims at designing a similarity degree to efficiently link network flows without forwarding and distorting the original traffic patterns.

3 Correlation Model

The correlation model proposed in this paper can be divided into three parts: Traffic preprocessing, feature extraction, and correlation detection. In the part A of this

section, the network traffic is preprocessed with using chaos theory, which to get chaotic characteristics of network flow. The method restores the hidden characteristics of network flow by reconstructing the time series of network traffic, and can grasp the inherent nature and regularity of chaotic time series. However, the chaotic characteristics obtained in the part A use in practical applications directly will cause a large amount of calculations. So in the part B of this section, we use the Principal Component Analysis (PCA) algorithm to process the chaotic characteristics obtained in the Part A to get more robust traffic characteristics. In the part C of this section, we calculate the similarity between the sender and receiver based on the cosine similarity using the characteristics obtained in the part B.

The algorithm proposed in this paper is as in Table 1.

Table 1: Algorithm steps

Step 1	Collect network flow.
Step 2	Get a time series is $\{x_1, x_2, \cdots, x_n\}$
Step 3	Calculate the time delay and embedding dimension of the time series obtained in the second step using the methods in the part A.
Step 4	The phase space is reconstructed to obtain the space characteristics of the network flow based on the embedding dimension and time delay acquired in the third step.
Step 5	Process the space characteristics obtained in the fourth step to get the final required traffic characteristics using the Principal Component Analysis (PCA) algorithm in the part B.
Step 6	According to the traffic characteristics obtained in the fifth step, we use cosine similarity in the part C to detect the linked flows. If the similarity is within the range of $\eta = \left(\frac{\sqrt{2}}{2}, 1\right]$, the both sides are considered to be correlated and recognized successfully. Otherwise, it is likely that the received flow is uncorrelated.

In the following part of this section, we start with a brief review of phase space reconstruction and Principal Component Analysis (PCA) algorithm and then each component of our scheme is described in details.

3.1 Traffic Preprocessing

Chaos theory is widely used in chemistry, physics, mechanics, mathematics as well as economic system, and it has been proved to be an important and effective theoretical method to solve nonlinear problems. In this paper, chaos theory provides a good means and methods for analyzing network traffic [17]. The collected network traffic time series can be extended from the low dimensional space to high dimensional space through phase space reconstruction technique of the chaos theory. In high dimensional space, it can recover regular characteristic of the network traffic from the seemingly irregular network traffic. In a word, phase space reconstruction technique of the chaos can restore the hidden nature of the original system, analyzing and extracting fixed characteristic value under the original rules and nature of the system accurately.

Given a network flow, its time series $\{x(n), n = 1, 2, \cdots, N\}$, a phase space reconstruction $X_{m,\tau}$ is defined as

$$X_{m,\tau} = (x(n), x(n+\tau), \cdots, x(n+(m-1)\tau)), \quad n = 1, 2, \cdots, N_m \tag{1}$$

Where τ is time delay, the number of vectors in the point set $N_m = N - (m-1)\tau$, and it is the total number of reconstituted by the time sequence of the status point. m is embedding dimension. The embedding dimension refers to the number of variables needed to describe the motion of a system. The appropriate m and τ can deeply explain the space-time characteristics of traffic and reveal the movement rule of the dynamic system.

From the above detailed description of the embedding dimension m and time delay τ, we know that it is important to calculate the two parameters of the network traffic time series. As far as we know, the famous Takens theorem implies that an appropriate time delay τ and a good embedding dimension m play an important role in reconstruction state space, among which the trajectories may maintain the diffeomorphism with original dynamic system. In other words, the dynamic system can be analyzed through phase space reconstruction from certain a time series. As previous work [10] has proved the network traffic is chaotic, we can use the phase space reconstruct technique to get the optimal parameter. The calculation steps are as follows.

Firstly, we get a timestamp of network traffic to get time sequence $\{x_1, x_2, \cdots, x_n\}$ and reconstruct a phase space with the time delay τ and the embedding dimension m describing in above. Secondly, we determine the time delay by using the C-C method [18]. According to the formula (1), we can get the reconstruction of points in space $X_i = \left(x_i, x_{i+1}, \cdots, x_{i+(m-1)\tau}\right)$, the correlation integral of embedding time series is defined as

$$C(m, N, \gamma, t) = \frac{2}{M(M-1)} \sum_{1 \le i \le j \le M} \theta(\gamma - d_{ij}), \gamma > 0 \tag{2}$$

$$\theta(z) = \begin{cases} 0, z < 0 \\ 1, z \ge 0 \end{cases} \tag{3}$$

Where $d_{ij} = \| X_i - Y_j \|$ is the distance between X_i and Y_j, $\| \bullet \|$ denotes maximal norm in this paper for convenience, $\theta(\bullet)$ is Heaviside function. The correlation integral measures the fraction of the pairs of points X_i, whose maximal norm separation is no greater than γ.

For the time series $\{x_i\}, i = 1, 2, \cdots, N$, it will be divided into t subsequence which do not overlap each other. We define each subsequence $S(m, N, \gamma, t)$ as

$$S(m,N,\gamma,t) = \frac{1}{t}\sum_{s=1}^{t}\left[C_s\left(m,\frac{N}{t},\gamma,t\right) - C_s^m\left(1,\frac{N}{t},\gamma,t\right)\right]$$

We choose a value corresponding to maximum (respectively minimum) radius γ, delta is define by

$$\begin{aligned}\Delta S(m,t) &= max\{S(m,\gamma_j,t)\} - min\{S(m,\gamma_j,t)\}\\ \Delta\bar{S}(t) &= \frac{1}{4}\sum_{m=2}^{s}\Delta S(m,t)\end{aligned}$$

The first minimum of $\Delta\overline{S}(t)$ is we need the optimal time delay τ.

In the end, we determine the best embedding dimension by using the Cao method [9]. The relative length of a point in a phase space is defined as

$$L(i,m) = \frac{\left\|X_{m+1}(i) - X_m^{NN}(i)\right\|}{\left\|X_m(i) - X_m^{NN}(i)\right\|}$$

Where $X_m(i)$ and $X_m^{NN}(i)$ are for the m dimension space of the ith vector and its nearest point. Then,

$$\begin{aligned}E(m) &= \frac{1}{N-m\tau}\sum_{i=1}^{N-m\tau}L(i,m)\\ E_1(m) &= {}^{E(m+1)}/_{E(m)}\end{aligned}$$

When $E_1(m)$ changes slowly or even is unchanged, the corresponding m will be the best embedding dimension. After obtaining two important parameters of delay time and embedding dimension, reconstructing phase space of original flow sequence, in m-dimensional space, the trajectory of n points of one-dimensional space can be expressed as

$$\begin{aligned}X &= [X(1), X(2), \cdots, X(N_m)]^T\\ &= \begin{pmatrix} x(1) & x(1+\tau) & \cdots & x(1+(m-1)\tau)\\ x(2) & x(2+\tau) & \cdots & x(2+(m-1)\tau)\\ \vdots & \vdots & \ddots & \vdots\\ x(N_m) & x(N_m+\tau) & \cdots & x(N_m+(m-1)\tau)\end{pmatrix}\end{aligned}$$

3.2 The Extraction of Traffic Traits

In order to reduce the amount of calculations in practical applications, we cannot directly use the chaotic characteristics obtained in the part A to calculate similarity. So in this part, we use Principal Component Analysis (PCA) [6, 12, 23] to analyze this higher dimensional matrix and obtain the major component of the reconstructed the original data. With the proposed approach, the traffic characteristics we obtain is more than an isolated subsequence, but contains all the information about the sequence. In this paper, the Principal Component Analysis (PCA) method is not only used as a tool to reduce the dimension of datasets, but also as a data processing method to extract the most important information from the original data set and use it as the extracted traffic characteristics.

For convenience that (10) will be recorded as

$$\begin{aligned}X &= \begin{pmatrix} a_{11} & \cdots & a_{1m}\\ \vdots & \ddots & \vdots\\ a_{N_m1} & \cdots & a_{N_mm}\end{pmatrix}\\ &= [X(1), X(2), \cdots, X(m)]\end{aligned}$$

In order to balance the weight of each element in each row, the data matrix is normalized and make its mean value of each row is 0. As shown by the following formula:

$$\begin{aligned}G &= \begin{pmatrix} g_{11} & \cdots & g_{1m}\\ \vdots & \ddots & \vdots\\ g_{N_m1} & \cdots & g_{N_mm}\end{pmatrix}\\ g_{ik} &= a_{ik} - \overline{a_i}; \forall i,k; 0 < i \le N_m, 0 < k \le m\\ \overline{a_i} &= \frac{1}{m}\sum_{j=1}^{m}a_{ij}\ ; 0 < i \le N_m\end{aligned}$$

After that, every $g_{ik}\,(0 < i \le N_m + 1; 0 < j \le m + 1)$ is used to calculate the covariance matrix $\boldsymbol{C}\,(m \times m)$ using the following formula:

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1m}\\ \vdots & \ddots & \vdots\\ c_{m1} & \cdots & c_{mm}\end{pmatrix}$$

$$\mathrm{Cov}(g_i, g_j) = c_{ij} = \frac{\sum_{k=1}^{L}(a_{ik}-\overline{a_i})(a_{jk}-\overline{a_j})}{m-1}$$

$$0 < i \le m, 0 < j \le m$$

And then we calculate the eigenvalues $\lambda = (\lambda_1, \lambda_2, \lambda_3, \cdots, \lambda_m)$ of the matrix $\boldsymbol{C}$ through the characteristic polynomial $|\boldsymbol{C} - \lambda\boldsymbol{I}| = 0$ ($\boldsymbol{I}$ denotes identity matrix, $\boldsymbol{C}$ represents covariance matrix mentioned above), and get the corresponding eigenvectors $\boldsymbol{V} = (V_1, V_2, V_3, \cdots, V_m)$.

We rearrange the eigenvalues in a descendant order and the eigenvectors correspondingly. Based on the aggregation of the eigenvalues, the sufficient corresponding eigenvectors are selected to calculate the principle components for the reconstruction of the original data matrix. For example, we choose s eigenvectors to reflect the information of the original data. So the front principle eigenvectors from v_1 to v_s comprise a new matrix $\boldsymbol{V}' = (v_1, v_2, v_3, \cdots, v_s)$. The principle matrix is constructed using the following formula:

$$\begin{aligned}P &= (P_1, P_2, P_3, \cdots, P_s) = G \times \boldsymbol{V}'\\ \boldsymbol{V}' &= (v_1, v_2, v_3, \cdots, v_s)\end{aligned}$$

According to the above formula, the energy of the original data $\boldsymbol{G}$ has been mapped to a new s-dimensional space. Meanwhile, some insignificant information has been ignored through the mapping process. Then we can calculate the principle components information in original data space.

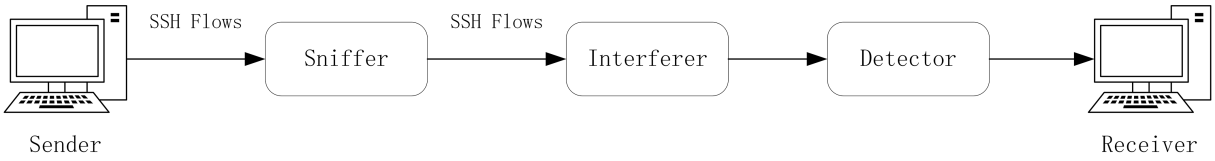


Figure 2: Experimental simulation environment

Finally, the mean value of each column of the new data matrix P is calculated, and the mean sequence is used as the traffic traits extracted from the algorithm:

$$F = \frac{1}{N_m} \sum_{j=1}^{N_m} P_{ij}$$

3.3 Detecting the Correlation Traffic Flows

The stream of packets passes through a noisy channel that may include all kinds of interferences. Finally, the flow arrives at the detector. Assuming that the flow to the detector is disturbed relative to the original flow, it is necessary to processed that based on the above scheme firstly. The detector intercept the received network flow and obtains the digital summary $F^{'}$ of the data stream that may have encountered network noise. Next, the detector reads the digital digest $F^{'}$ stored in a third-party database. Calculate the cosine similarity of F and $F^{'}$:

$$\cos\theta = \frac{\boldsymbol{F} \cdot \boldsymbol{F}^{'}}{\left|\boldsymbol{F}\right| \left|\boldsymbol{F}^{'}\right|} = \frac{\sum_{i=1}^{s}(F_i \times F_i^{'})}{\sqrt{\sum_{i=1}^{s}(F_i)^2} \times \sqrt{\sum_{i=1}^{s}(F_i^{'})^2}}$$

Cosine similarity is a common method of determining the correlation between two n-dimensional vectors. It mapped individual indicator data to vector space and calculates the cosine of the angle between two vectors as a measure of the similarity between two variables. The closer the cosine of the angle to 1, the more similar [1]. In this paper, we pay more attention to the degree of similarity of variation trend between the internal components of the vector, so we use cosine similarity to measure the similarity between the sender and the receiver. If the similarity is within the range of $\eta = \left(\frac{\sqrt{2}}{2}, 1\right]$, the flows are considered to be correlated and recognized successfully. Otherwise, the traffic received is probably not related.

In fact, according to the scheme proposed in this paper, no additional communication overhead is required between the sender and the receiver, except for the shared digital summaries.

4 Simulation Results

The experiment simulation environment design is shown in Figure 2. A network flow passing through the Sniffer gets monitored in real-time, and then generate a digital digest according to traffic traits, and the generated digital digest is passed to the detector secretly through a secure channel, the Interferer implement the potential network jitter and countermeasures intentionally introduced by the adversaries (such as packet insertions and packet losses), once receipt of a disturbed version of the primitive traffic, the detector tries to confirm whether the two flows are linked based on the comparison of digital digest of source and sink ends. In addition, the SSH flows used in the experiment came from CAIDA anonymous network traces, and each SSH stream has a length of 2000. These real SSH streams reflect some of the typical behavior of people in the network.

4.1 Detection Rate of Packet Insertion and Deletion

The detection rate of the presented scheme in this paper is also evaluated in a networked environment where not only packet deletions but also packet insertions occur. Tests have shown that the insertion of chaff packets are conformed to a Pareto distribution [7], and the removal of the packets is independently and randomly. The experimental results shown in Figure 3.

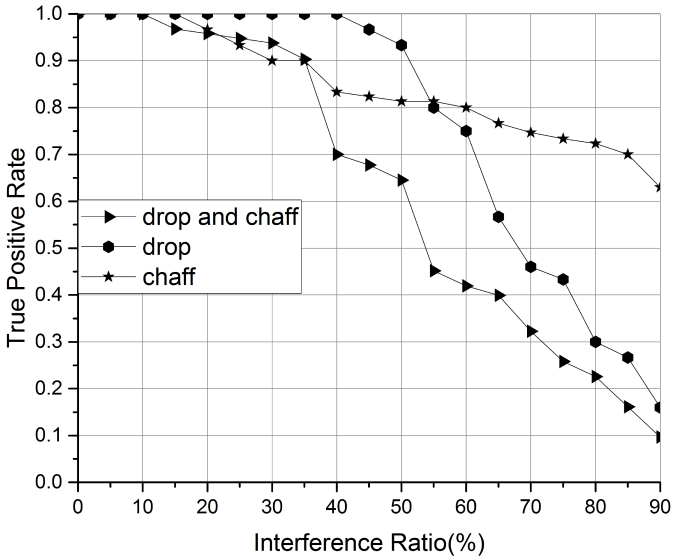


Figure 3: True positive rates comparison under Packet Insertion and deletion

Encouragingly, even if packets injection rate and loss rate are up to 10%, the correlated traffic can still be accurately detected. In addition, from the experimental results, it can be observed that the detection correctness

rate is fewer affected by different number of chaff packets than the packet loss. There is the strong possibility that the digital digest designed in this scheme is based on the inherent characteristics of the flows. Packet deletions will cause the inherent characteristics of the stream to be severely damaged, so that the reconstructed phase space can not fully reflect the original flow characteristics, which leads to the decrease of detection efficiency. However, packet insertions may only influence the inherent characteristics of a portion of the source stream.

4.2 Accuracy Under Various Interference

As far as we know, existing network flow watermark designs can be roughly divided into two categories: Interval-based and IPD-based [3]. Non-blind watermark (RAINBOW) is a good example based on IPD schemes. RAINBOW embeds watermarking by fine-tuning the packet delay in network traffic, after recording the delay sequence information between packets [7]. In interval-based schemes, for example, an interval center-based watermarking (ICBW) presented by Wang *et al.* [21] randomly selects time interval as two different subsets and performs an operation on the centroid of the selected entire time interval pairs to embed a watermark.

In this section, packet deletions and chaff packets are introduced in the case of network jitter. Following the observation of previous work that shows jitter (difference of two delays) is approximately i.i.d. zero-mean Laplace distributed [7], we vary the standard deviation of jitter σ over $\{10, 20, 30, 40\}ms$. Evaluate the performance of three scheme in the same network environment: Our solution, ICBW and RAINBOW. For each solution, 6000 different network flows are used to test their performance under various interference. In addition there are 6,000 network flows acting as control groups that are passed directly to the detector without any interference. Figure 4, Figure 5 and Figure 6 depict the average true positive rates and false positive rates for these 6000 network flows respectively.

As shown in Figure 4 and Table 2, where P_i and P_d denote packet loss interference rate and packet insertions interference rate respectively and σ represents the number of network jitters added. With the increase of the number of available packets, the detection rate will be greatly improved. Compared with RAINBOW and ICBW, actually, the fewer number of packets required by our scheme in achieving the same level of accuracy. Moreover, as the number of available packets increases, the detection rate will be greatly increased. The figure has shown that if the number of packets is the same as the number of packets in the original stream, the detection rate of our solution is over 90% even if as many as 30% packets simultaneously deleted and inserted besides jitter as high as $40ms$, yet ICBW and RAINBOW basically keep smaller than 80% detection rate under these conditions.It may be that, network traffic has self-similarity and chaotic, and reconstructed phase space based on more data packets, which reflects the inherent law of data flow implicitly, so the detection rate will be higher. As shown in Figure 5 and Table 3, the overall detection rate of our program is still higher than that of ICBW and RAINBOW.

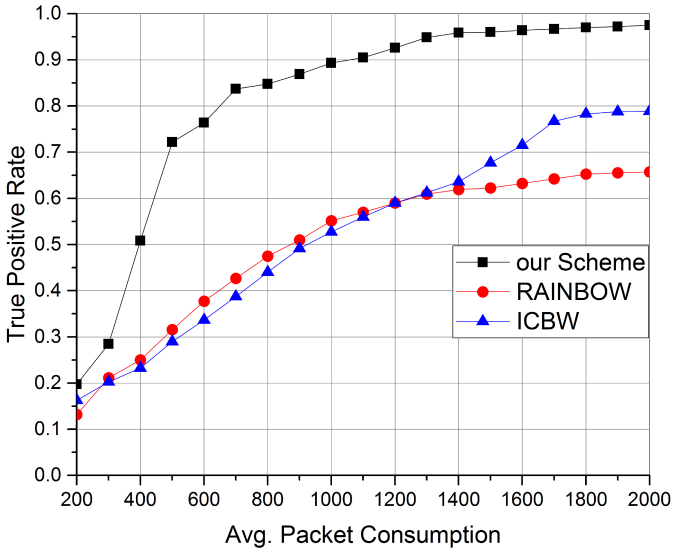


Figure 4: True positive rates comparison under $P_i = 30\%$, $P_d = 30\%$ and $\sigma = 40ms$

Table 2: True detection rate under $P_i = 30\%$, $P_d = 30\%$ and $\sigma = 40ms$

Average Packet Consumption	**400**	**800**	**1200**	**1600**	**2000**
Our Scheme	0.512	0.849	0.922	0.972	0.974
RAINBOW	0.251	0.486	0.598	0.632	0.665
ICBW	0.242	0.442	0.606	0.724	0.778

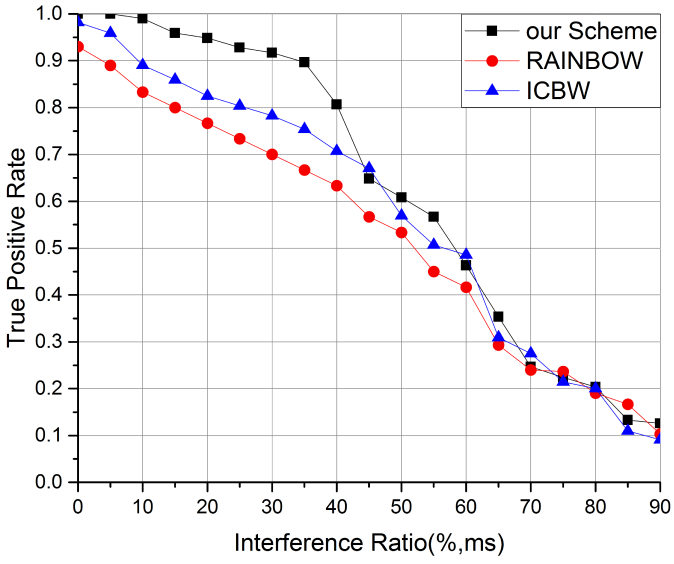


Figure 5: True positive rates comparison under Packet Insertion, Removal and Network Jitter

Table 3: True positive rates under Packet Insertion, Removal and Network Jitter

Interference Rate	10%	20%	30%	40%
Our Scheme	0.992	0.958	0.917	0.816
RAINBOW	0.839	0.787	0.708	0.648
ICBW	0.898	0.823	0.789	0.710

Figure 6 and Table 4 show that since the digital digest designed in this paper is generated according to traffic pattern and there seems to be some inherent similarity between two uncorrelated network flows, when the original traffic is severely disrupted and nine-tenths of packets are not available, the false detection rate for our solution may not be ideal, but it always does not exceed 8%. In addition, as the number of required packets increases, the false positive rate of our scheme has a very significant decrease. The figure also shows that when the number of received packets exceeds 50% of the original flow length, the error rate of our designs is lower than ICBW. When the available packet reaches 80% of the original packet flow length, the false positive rate of our scheme is smaller than RAINBOW, which is basically not exceeding 1%.

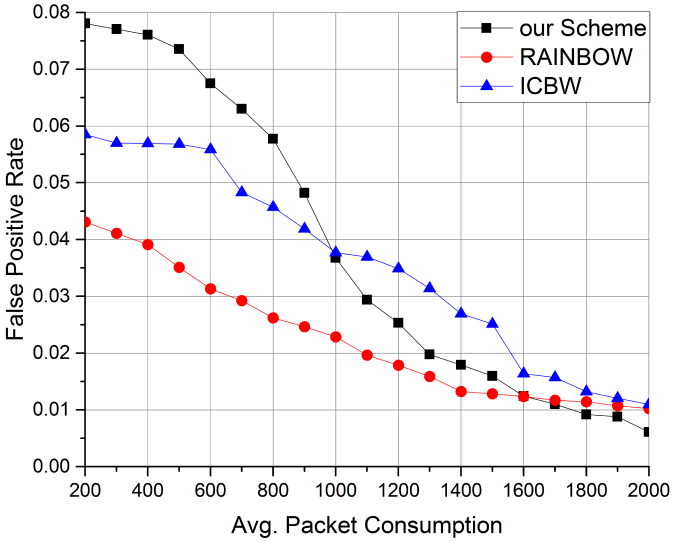


Figure 6: False positive rates comparison under $P_i = 30\%$, $P_d = 30\%$ and $\sigma = 40ms$

Table 4: False positive rates comparison under $P_i = 30\%$, $P_d = 30\%$ and $\sigma = 40ms$

false positive rate	400	800	1200	1600	2000
Our Scheme	0.072	0.058	0.026	0.013	0.007
RAINBOW	0.039	0.027	0.019	0.013	0.011
ICBW	0.057	0.041	0.036	0.016	0.011

5 Conclusion

In this paper, we proposed a novel flow correlation scheme based on Chaos Theory and Principal Component Analysis that does not rely on network watermarking. Only main part of the flow characteristics are utilized without interfering with the communication patterns of the intercepted flows, which prevents attackers from detecting the trace process. The ideal network flow watermarking technology needs to satisfy robustness and invisibility simultaneously, but it can only meet one of them in practical applications, and our solution does not have this concern. And there is no additional communication overhead between the sender and the receiver, except for the shared digital summaries. Finally, theoretical analysis and experimental results confirmed the correctness and the operability of network flow correlation model based on chaos theory and PCA despite the presence of network jitter, packet additions and removals.

There are still some limitations of our proposed method, and its false positive rate is higher than network watermarking, and this is our following work. We may be able to understand more deeply the principles of common coding techniques such as network coding, channel coding, source coding, video coding, etc., and explore the intrinsic connection of these technologies, and seek more robust and adaptable information coding techniques to optimize the feature coding module of this scheme.

Acknowledgments

This study is supported by the National Natural Science Foundation of China(NO. 61370007). The Postgraduate Scientific Research Innovation Ability Training Plan Funding Projects of Huaqiao University(1611314002).

References

[1] S. Amit, *et al.*, "Modern information retrieval: A brief overview," *IEEE Data Engineering Bulletin*, vol. 24, no. 4, pp. 35–43, 2001.

[2] M. Behi, M. Ghasemi, and H. V. Nejad, "A new approach to quantify network security by ranking of security metrics and considering their relationships," *International Journal of Network Security*, vol. 20, no. 1, pp. 141–148, 2018.

[3] Y. Chen, N. Zhang, H. Tian, T. Wang, and Y. Cai, "A novel connection correlation scheme based on threshold secret sharing," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2414–2417, 2016.

[4] G. Danezis, "The traffic analysis of continuous-time mixes," in *International Workshop on Privacy Enhancing Technologies*, pp. 35–50, May 2004.

[5] X. Gong, M. Rodrigues, and N. Kiyavash, "Invisible flow watermarks for channels with dependent substitution, deletion, and bursty insertion errors," *IEEE*

Transactions on Information Forensics and Security, vol. 8, no. 11, pp. 1850–1859, 2013.

[6] H. Harold, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417, 1933.

[7] A. Houmansadr, N. Kiyavash, and N. Borisov, "Non-blind watermarking of network flows," *IEEE/ACM Transactions on Networking*, vol. 22, no. 4, pp. 1232–1244, 2014.

[8] C. T. Li, M. S. Hwang, and S. Chen, "A batch verifying and detecting the illegal signatures," *International Journal of Innovative Computing, Information and Control*, vol. 6, no, 12, pp. 5311–5320, 2010.

[9] Z. Li, H. Zheng, and C. Pei, "A modified cao method with delay embedded," in *The 2nd International Conference on Signal Processing Systems (ICSPS'10)*, pp. V3–458–V3–460, July 2010.

[10] X. Ma and Y. Chen, "Ddos detection method based on chaos analysis of network traffic entropy," *IEEE Communications Letters*, vol. 18, no. 1, pp. 114–117, 2014.

[11] A. Montieri, D. Ciuonzo, G. Aceto, and A. Pescapé, "Anonymity services tor, I2P, JonDonym: Classifying in the dark," in *The 29th International Teletraffic Congress*, pp. 81–89, Sep. 2017.

[12] C. Mu, X. Huang, J. Wu, and Y. Ma, "Network traffic signature generation mechanism using principal component analysis," *China Communications*, vol. 10, no. 11, pp. 95–106, 2013.

[13] F. Nabi and M. M. Nabi, "A process of security assurance properties unification for application logic," *International Journal of Electronics and Information Engineering*, vol. 6, no. 1, pp. 40–48, 2017.

[14] E. U. Opara and O. A. Soluade, "Straddling the next cyber frontier: The empirical analysis on network security, exploits, and vulnerabilities," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 10–18, 2015.

[15] J. Qin, R. Sun, X. Xiang, H. Li, and H. Huang, "Anti-fake digital watermarking algorithm based on QR codes and DWT," *International Journal Network Security*, vol. 18, no. 6, pp. 1102–1108, 2016.

[16] T. Shi, W. Shi, C. Wang, and Z. Wang, "Compressed sensing based intrusion detection system for hybrid wireless mesh networks," in *International Conference on Computing, Networking and Communications (ICNC'18)*, pp. 11–15, Mar. 2018.

[17] J. Song, D. Meng, and Y. Wang, "Analysis of chaotic behavior based on phase space reconstruction methods," in *The Sixth International Symposium on Computational Intelligence and Design (ISCID'13)*, pp. 414–417, Oct. 2013.

[18] L. Tang and J. Liang, "CC method to phase space reconstruction based on multivariate time series," in *The 2nd International Conference on Intelligent Control and Information Processing (ICICIP'11)*, pp. 438–441, July 2011.

[19] J. Wang, Y. Yu, and K. Zhou, "A regular expression matching approach to distributed wireless network security system," *International Journal Network Security*, vol. 16, no. 5, pp. 382–388, 2014.

[20] X. Wang, D. S. Reeves, and S. F. Wu, "Inter-packet delay based correlation for tracing encrypted connections through stepping stones," in *European Symposium on Research in Computer Security*, pp. 244–263, Oct. 2002.

[21] X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems," in *IEEE Symposium on Security and Privacy (SP'07)*, pp. 116–130, May 2007.

[22] X. Xu, J. Zhang, and Q. Li, "Equalized interval centroid based watermarking scheme for stepping stone traceback," in *IEEE International Conference on Data Science in Cyberspace (DSC'16)*, pp. 109–117, June 2016.

[23] K. Yoshiki, F. Kensuke, and S. Toshiharu, "Evaluation of anomaly detection based on sketch and PCA," in *IEEE of Global Telecommunications Conference (GLOBECOM'10)*, pp. 1–5, Dec. 2010.

[24] L. Zhang, Y. Kong, Y. Guo, J. Yan, and Z. Wang, "Survey on network flow watermarking: Model, interferences, applications, technologies and security," *IET Communications*, vol. 12, no. 14, pp. 1639–1648, 2018.

Biography

Yang Chen was born in Chongqing, China in 1994. She received the B.S. Degree from Wuhan Textile University, Hubei, China in 2016. She is currently pursuing the M.S. Degree in Huaqiao University. Her research interests include Digital Watermarking and Property Protection and Blockchain and Application.

Yonghong Chen received the Ph.D. degree from Chongqing University, Chongqing, China, in 2005. He is a Professor in Huaqiao University of China. His current interests include Network and Information Security, Network intrusion detection, Digital Watermarking and Property Protection and Blockchain and Application.

A Secure Authenticated Key Agreement Protocol for Application at Digital Certificat

Javad Saadatmandan and Amirhossein Rahimi
(Corresponding author: Javad Saadatmandan)
Department of Mathematics, Qom Branch, Islamic Azad University
Qom, Iran
jsaadatmandan2014@gmail.com
(Received Mar. 25, 2018; Revised and Accepted Sept. 2, 2019; First Online Feb. 9, 2020)

Abstract

To establish secure channel for network communication in open and distributed environments, authenticated key agreement protocol is an important primitive for establishing session key. So far, a great deals of identity-based protocols have been proposed to provide secure mutual authentication and common session key establishment in two-party setting for secure communications in the open environment. Majority of the existing authenticated key agreement protocols only provide partial forward secrecy. Therefore, such protocols are unsuitable for real-world applications that require a stronger sense of perfect forward secrecy. In this paper, we present a secure two-party identity-based authenticated key agreement protocol with achieves most of the required security attributes. We also show that the scheme achieves the security attributes include known-key secrecy, perfect forward secrecy, PKG forward secrecy, key-compromise impersonation resilience, unknown key-share resilience, no key revelation and known session-specific temporary key information secrecy and also proposed algorithm achieves the shorter run time, lower computation cost, lower communication cost, and a more effective storage method. In addition, the adversary can not compromise the agreed session key.

Keywords: Identity-Based Cryptography; Key Agreement; Perfect Forward Secrecy; PKG Forward Secrecy

1 Introduction

Key agreement protocol is used to provide secure communications in open and distributed environments [4]. Key establishment is a process whereby two (or more) entities can establish a shared secret key (session key) after message interactions. There are two different approaches to key establishment between two entities. In one scenario, one entity generates a session key and securely transmits it to the other entity, this is known as enveloping or key transport [11, 15].

In order to provide authentication for the key agreement protocol, public key certificate is often used in the traditional PKI setting. This require the parties to obtain and verify certificates whenever they want to use a specific public key and the management of public key certificates remains a technically challenging problem. Adi Shamir introduced the identity-based cryptography in 1984 [16]. His idea was to allow parties to use their identities as public keys. With the help of Private Key Generator (PKG), the users attain their private keys and perform cryptographic tasks subsequently. Authentication without the help of public key certificate is the major advantage of identity-based cryptography. Therefore,identity-based key agreement protocols without pairing may be more appealing in practice.

Two-party authenticated key agreement (AK) protocol not only allows parties to compute a session key known only to them but also ensures the authenticity of the parties [12,15]. This secret session key can be used to provide privacy and data integrity during subsequent sessions. A key agreement protocol is said to provide implicit key authentication (of Bob to Alice) if Alice is assured that no other entity besides Bob can possibly ascertain the value of the secret key. A key agreement protocol that provides mutual implicit key authentication is called an authenticated key agreement protocol (or AK protocol) [9]. A key agreement protocol provides key confirmation (of Bob to Alice) if Alice is assured that Bob possesses the secret key. A protocol that provides mutual key authentication as well as mutual key confirmation is called an authenticated key agreement with key confirmation protocol (or an AKC protocol).

In this study, an effective and secure authenticated key agreement (AK) protocol is proposed based on a secure one-way hash function, discrete logarithm problem. By comparing the proposed algorithm with other similar algorithms, we found out that the proposed algorithm had a shorter run time, a lower computation and communication cost, and a more effective storage method. We also investigated the fundamental characteristics of hash

functions by arguing that, as these functions cannot be executed computationally via inverse operators, their application in the proposed algorithm would provide further protection against known cyber attacks.

It is desirable for any authenticated key agreement protocol to possess the following security attributes:

Known-key secrecy. The overture of one secret session key should not compromise other session keys. Therefore key agreement can prevent to compromise session keys and the insider, replay, parallel session, reflection, and man in the middle attacks.

Forward secrecy. If long-term private keys of one or more of the entities are compromised, the secrecy of previously established session keys should not be affected. We say that a system has partial forward secrecy if the compromise of one (or more but not all) of the entities' long-term keys can be corrupted without compromising previously established session keys, and perfect forward secrecy means if the long-term keys of all the entities involved may be corrupted without compromising any session key previously established by these entities. In order to resistance against comprehensive research attack for recovery of secret random number the better way is that the length of the random number should be greater than secret session key. Therefore, random numbers are required for safekeeping confidential information(secret session key). Remember that Leaking the server's secret key can lead to the risk of the session keys being discovered.

PKG Forward Secrecy. The PKG's master key may be corrupted without compromising the security of session keys previously established by any users. It certainly implies the perfect forward secrecy.

Key-compromise impersonation resilience. For an entity called Alice, the compromise of an entity Alice's long-term private key will allow an adversary to impersonate Alice, but it should not enable the adversary to impersonate other entities to Alice.

Unknown key-share resilience. An entity Alice should not be able to be coerced into sharing a key with any entity Eve when in fact entity Alice thinks that he is sharing the key with another entity Bob.

Known session-specific temporary information secrecy [13]. Some random private information is used as an input of the session key generation function. The revelation of this private temporary information should not compromise the secrecy of (other) generated session key. Known session-specific temporary information secrecy was first explored and discussed by Canetti-Krawczyk in [1]. Generally, this important security attribute requires that if the ephemeral secrets of a session are accidentally leaked to the adversary, the secrecy of the specific session key should not be affected. This revelation is reasonably not partial as it may happen in some practical scenarios. In 2009 and 2010, Cao etc. [2, 3] proposed two pairing-free identity-based authenticated key agreement schemes with two or three passes (one round). They all achieved the basic security attributes without pairing operation. However, we find that their protocols do not offer an important security feature, namely known session specific temporary information secrecy, which considers the impact of ephemeral secrets exposure in affecting the secrecy of the session key.

No key control. Neither entity should be able to force the session key to be a preselected value. Key escrow [14] is desirable under certain circumstances especially in certain closed groups applications. For example,escrow is essential in situations where confidentiality as well as survey trail are legal requirements, such as secure communications in the health care profession. So far, some identity-based authenticated key agreement protocols in the escrow mode (e.g. [5, 14, 18, 20, 21]) were proposed. But most of them did not provide perfect forward secrecy attribute. Although Shim [17] proposed a protocol To be claimed to provide such a property, it was later found to be vulnerable to the manin-the-middle attack [19]. In 2006, Gentry proposed an identity-based encryption system [7] that is fully secure in the standard model and has several advantages over previous such systems, Its complexity assumption is called the truncated **q-ABDHE**. Based on the work of Gentry, we present a new two-party identity-based authentication key agreement protocol that can be used in the escrow mode, whilst it achieves the perfect forward secrecy attribute.

The remainder of this paper is organized as follows. Section 2 gives the necessary technical backgrounds and reviews of the identity-based encryption scheme of Gentry and the scheme of Cao *et al.*. In Section 3, we put forward our new proposed scheme. In Section 4, we give the security analysis and efficiency of the proposed protocols, as well as comparisons over comparably protocols. In this paper, we discuss this problem in detail and give an improved one round scheme with efficient computational performance. Finally, we draw some conclusions.

2 Technical Backgrounds

2.1 Bilinear Maps

Let G_1 and G_2 are two (multiplicative) cyclic groups of prime order p, g is a generator of G_1, assume that the discrete logarithm problem (DLP) is hard in both G_1 and G_2. An admissible pairing e is a bilinear map $e : G_1 \times G_1 \longrightarrow G_2$, which satisfies the following three properties [11]:

Table 1: Notations

Symbol	Definition	Symbol	Definition
ID	User ID (User Identiiy)	Not +	Operator XOR
G	Cyclic Additive Group	F_p	Prime Finite Field
G_1 , G_2	Multiplicative Cyclic Group	$H(0)$	Secure Scrambling Function
PKG	Private Key Generator	Z_p^*	Multiplication Group p
AK	Authenticated Key	$\|$	Concatenation Operation
SK	Session key	X (modp)	Remainder of X:p
P_{pub}	Public Key	r_{ID}	Random
e	Bilinear Map	ECC	Elliptic Curve Cryptography
DLP	Discrete Logarithm Problem	CDH	Computional Dffe-Hellman Assumption

- Bilinear: for all $u, v \in G_1$ and $a, b \in Z_p^*$, we have

$$e\left(u^a, v^a\right) = e\left(u, v\right)^{ab};$$

- Non-degenerate: $e\left(g, g\right) \neq 1$;
- Computable: If u, $v \in G_1$, one can compute $e\left(u, v\right) \in G_2$ in polynomial time efficiently.

2.2 Elliptic Curve Groups

$y^2 = \left(x^3 + ax + b\right) \bmod P$ with, $a, b \in Z_P$ and $8a^3 + 81b^2 \bmod p \neq 0$. The points on E/F_p together with an extra point 0 form a group

$$G = \{(x, y) : x, y \in F_p, E(x, y) = 0\} \mathrm{U}, \mathrm{o}.$$

G is a cyclic additive group under the point addition "+" defined as follows: Let $p, q \in G, l$ to be the line containing p and q (tangent line to E/F_p if $p = q$), and R, the third point intersection of l with E/F_p at R, o and $p + q$ Scalar multiplication over E/F_p by an integer is defined by repeating addition, i.e. $kp = p + p + \cdots + p(k)$.

3 The New Proposed Scheme

In this section, we propose an efficient perfect forward secure one-round identity based authenticated key agreement protocol without pairing, which achieves almost all the known security attributes, especially the known session-specific temporary information secrecy. At the same time, it is more computational efficient than the other comparable schemes.The security of the protocol can be reduced to the CDH assumption in the random oracle model. The protocol consists of three phases, *i.e.* **Setup, Key Generation** and **Key Agreement**. These three phases are almost as same as that of Cao's schemes [2] with slight modification, and the generation of the session key is different. we would like an escrowable identity-based key agreement protocol in which the user's session key could be recovered by the PKG whilst the others couldn't recover the user's past session keys even the long term key of user was compromised. The protocol involves three entities: two users called Alice and Bob who wish to establish a shared secret session key, and a PKG that is responsible for the creation and distribution of users' private keys using its master key. The protocol consists of four phases, *i.e.* **Setup, Key Generation** and **Key Agreement** and **Correctness Verification**. In order to keep the integrity of description of the protocol, We give the brief description as below:

Setup: To provide a private key generation service, the private key generator (PKG) first generates the system parameters and its public/private key pairs as follows. Given a security system parameter k, the private key generator (PKG) chooses the tuple $\{E/F_p, G, p\}$ as defined in Section 2, choose the master private key $x \in Z_p^*$, calculates the public key of PKG as $P_{pub} = kp^x$. And then choose two groups of prime order p, three secure cryptographic hash functions and one the bilinear map, *i.e.* G, G_t: groups of prime order p; $e : G \times G \longrightarrow G_t$: The bilinear map:

$$H_1 : \{0,1\}^* \times G \longrightarrow Z_p^*$$
$$H_2 : \{0,1\}^* \times \{0,1\}^* \times G \times G \times G \longrightarrow \{0,1\}^k$$
$$H_3 : \{0,1\}^* \times \{0,1\}^* \times G \times G \times G \times G \longrightarrow \{0,1\}^k$$

Let $g, p, t \in G$, $t = g^x \bmod p$, $g_T = e(g, t) \in G_t$. The public key of the PKG Then the PKG publishes the system public parameters as $< E/F_p, G, P_{pub}, t, g_T, H_1, H_2, H_3 >$ and the master private key of the PKG is x.

Key Generation: To generate a private key for the identity Z_p^*, the PKG generates a long-term private key for user identity as bellow. For a user whose identity $r_{ID} \in Z_P^*, r_{ID} \neq x$, the PKG generates a random $r_{ID} \in Z_P^*$, it always assigns identical r_{ID} for a given identity ID and computes $R_{ID} = (kp^{-r_{ID}})^{\frac{1}{(x - ID)}}, h_{ID} = H_1(ID \| R_{ID})$ and outputs the private key as $d_{ID} =< R_{ID}, s_{ID} >$, where $s_{ID} = r_{ID} + h_{ID}x$. The long-term private key of user with identity ID is transmitted to him via a

secure out-of-bound channel. The user with identity ID can verify his long-term private key by checking the equation $s_{ID}P = R_{ID} + H_1(ID \parallel R_{ID})P_{pub}$. The long-term private key is valid if the equation holds and vice versa. Suppose there are two entities called Alice (act as the initiator) and Bob (act as the responder) who want to establish the session key.

Key Agreement:

Alice and Bob are two entities who want to establish a shared session key with implicit key authentication by running the following protocol. We use ID_A and ID_B to demonstrate the identification strings of Alice and Bob (It could be E-mail address or any other strings). The protocol is a 2-pass procedure, the details are as follows.

Scheme 1.

1) $A \longrightarrow B : \{ID_A, R_A\}$. B chooses $b \in Z_p^*$ and computes the message

$$T_B = g_B^{b(R_A + H_1(ID_A \| R_A)P_{pub})} \bmod p.$$

2) $B \longrightarrow A : \{ID_B, R_B, T_B\}$. A chooses $a \in Z_p^*$ and computes the message

$$T_A = g_A^{b(R_B + H_1(ID_B \| R_B)P_{pub})} \bmod p.$$

3) $A \longrightarrow B : \{T_B\}$. B computes

$$K_{BA} = (b+1)\, s_B^{-1} T_A + H_1\left((ID_A \| R_A)\, p_{pub}\right) + bp$$

and

$$SK_{BA} = H_2\left(ID_A \| ID_B \| T_A \| T_B \| K_{BA}\right).$$

Finally A computes

$$K_{AB} = (a+1)\, s_A^{-1} T_B + H_1\left((ID_B \| R_B)\, p_{pub}\right) + ap$$

and

$$SK_{BA} = H_2\left(ID_A \| ID_B \| T_A \| T_B \| K_{BA}\right).$$

Correctness Verification:

At the end of the protocol execution, Alice and Bob will agree on the same session key. We can easily verify that $sk = SK_{BA} = SK_{AB}$ From the form of and SK_{BA} and SK_{AB}, we can know that if the adversary acquired the session-specific ephemeral secrets a and b, he can not learns the session key SK_{BA} or SK_{AB}, because he can not compute $H_1((ID_A \| R_A)p_{pub})$, T_A, T_B and too s_A^{-1}, s_B^{-1}. Because hash function inverse is computationally infeasible without knowing server 's secret key. So this scheme does to gain the additional security attribute - Known session-specific temporary information secrecy. It is easy to validate that

$$\begin{aligned}
K_{AB} &= (a+1)\, s_A^{-1} T_B + H_1\left((ID_A \| R_A)\, p_{pub}\right) + ap \\
&= as_A^{-1} T_B + s_A^{-1} T_B + H_1\left((ID_A \| R_A)\, p_{pub}\right) + ap \\
&= abp + ap + bp + s_B p \\
&= bap + bp + ap + s_A p \\
&= bs_B^{-1} T_A + s_B^{-1} T_A + H_1\left((ID_A \| R_A)\, p_{pub}\right) + bp \\
&= (b+1)\, s_B^{-1} T_A + H_1\left((ID_A \| R_A)\, p_{pub}\right) + bp \\
&= K_{AB}.
\end{aligned}$$

So we get the same agreed session key with $sk = SK_{BA} = SK_{AB}$.

Scheme 2.

1) $A \longrightarrow B : \{ID_A, R_A, T_A\}$.
The initiator A chooses a random ephemeral key $a \in Z_p^*$ and compute the message $T_A = ap$;
2) $B \longrightarrow A : \{ID_B, R_B, T_B\}$
On receiving the message from A, The responder B chooses a random ephemeral key and compute the message $T_B = bp$; Finally, A computes

$$\begin{aligned}
K_{AB} &= (T_B + R_B + H_1((ID_B \| R_B)P_{pub}) \cdot \\
&\qquad (a + s_A); \\
SK_{AB} &= H_3(ID_A, ID_B, T_A, T_B, K_{AB}).
\end{aligned}$$

B computes

$$\begin{aligned}
K_{AB} &= (T_A + R_A + H_1((ID_A \| R_A)P_{pub}) \cdot \\
&\qquad (b + s_B); \\
SK_{BA} &= H_2(ID_A, ID_B, T_A, T_B, K_{BA});
\end{aligned}$$

It is easy to validate that

$$\begin{aligned}
K_{AB} &= (T_B + R_B + H_1((ID_B \| R_B)P_{pub}) \cdot \\
&\qquad (a + s_A); \\
&= (bP + s_B P)(a + s_A) \\
&= (aP + s_A P)(b + s_B) \\
&= (T_B + R_B + H_1((ID_B \| R_B)P_{pub}) \cdot \\
&\qquad (b + s_B) \\
&= K_{BA} \\
&= (a + s_A)(b + s_B)P \\
&= abP + as_B P + bs_A P + s_A s_B P.
\end{aligned}$$

We can verify that $sk = SK_{BA} = SK_{AB}$.

4 Analysis of Security and Efficiency

In this section, we give the general analysis of security and efficiency, as well as comparisons over comparable protocols.

A. Security Analysis.

We informally declare that our new proposed scheme has several desirable security attributes, such as known-key secrecy, PKG forward secrecy, key-compromise impersonation resilience, unknown key-share resilience, and no key control. Especially, this scheme achieves the perfect forward secrecy attribute.

1) Known-key secrecy.
If one session key is compromised, this does not mean that any other session keys are compromised. The fact is that each run of the protocol computes a different session key which depends on the ephemeral private keys x and y. While x and y were selected randomly by Alice and Bob independently.

2) Key-compromise impersonation resilience.
Suppose an adversary called Eve who knows Alice's long term private key wishes to masquerade as Bob to Alice. Although Eve could declare with Bob's identity and send T_B to Alice, but without knowing the private key of Bob, he couldn't use K_{AB} to compute the identical session key as same as that of Alice.

3) Unknown key-share resilience.
In order to attack this protocol, the adversary is required to learn the private key of some entity. In fact, Chen and Kudla [5] has pointed out that the unknown key-share resilience attribute is implied by the implicit key authentication.

4) No key control.
In this protocol, x and y are selected by Alice and Bob randomly, neither entity is able to force the session key to be a preselected value. If the adversary Eve modified the exchanged message with such purpose, Alice and Bob can hardly compute the same session key.

5) Key agreement secrecy.
The overture of one secret session key should not compromise other session keys. Therefore key agreement can prevent to compromise session keys and the insider, replay, parallel session, reflection, server spoofing, and man in the middle attacks. In order to resistance against comprehensive research attack for recovery secret random number is better the length of the random number to be greater than secret session key.Therefore, random numbers and session keys are required for safekeeping confidential information(secret session key).Remember that Leaking the server's secret key can lead to the risk of the session keys being discovered.

6) Perfect forward secrecy.
If the long term keys of two parties involved were compromised, one (except the PKG) could compute K_{AB_1}, K_{BA_1} and $<as_BP$, bs_AP, $s_As_BP, T_A, T_B, s_A^{-1}, s_B^{-1}, H_1\left((ID_B \| R_B)\, P_{pub}\right) >$ but he couldn't compute K_{AB_2} and K_{BA_2} without knowing same agreed session key with $SK_{BA} = SK_{AB}$.
In order to compute K_{AB_1}, K_{BA_1}, K_{AB_2} and K_{BA_2} at two proposed schemes , one should solve the Computational Diffie-Hellman hard problem and inverse one-way hash function.

7) PKG forward secrecy.
If the adversary acquired the system master key of PKG, it means that the adversary can also acquire the private key of both Alice and Bob. It still couldn't compute . In order to compute , one should solve the Computational Diffie-Hellman problem and other inverse one-way hash function.

8) Known session-specific temporary information secrecy.
If the adversary knew the ephemeral session secrets a and b but not the long-term key of both, then he could only compute $< abP$, as_BP, $bs_AP, R_{ID} >$ but not s_As_BP, s_{ID}.
In order to compute s_As_BP, one need to acquire at least one of the long term private key of Alice and Bob. It is still a Computational Diffie- Hellman hard problem. In this scheme, the computation of K_{AB} or K_{BA} needs only two scalar addition and two scalar multiplication operations. If we consider the preprocessing of computation of $R_{ID} + H_1\left((ID_{ID} \| R_{ID})\, P_{pub}\right)$, then the computation cost is only one scalar addition and one scalar multiplication. It is more efficient than that of Cao' s scheme.

9) Resistance to the Modification Attack.
In the proposed protocol, each authentication message is supported via a new secret randomized number and accompanied by a one-way hash function. Without this randomized number, the attacker is unable to calculate the correct hash function value for authenticating the ID message. For this reason, it is very difficult to generate a manipulated message from n valid message.

10) Resistance to Disclosure Server's Secret Key Attack.

Proof. Even if the server's secret key x is disclosed, the attacker would not be able to retrieve ID_{ID} and h_{ID} from $R_{ID} + H_1\left((ID_{ID} \| R_{ID})\, P_{pub}\right)$. Since, due to using only one H(0) function method, the server can easily change/modify the secret key x and return it to the smart card. Remember that Leaking the server's secret key can lead to the risk of the session keys being discovered. □

11) Resistance to the Server spoofing Attack.

Proof. In this type of attack, a attacker cannot masquerade as a legal server since he cannot calculate $s_A s_B P, s_{ID}$ and R_{ID} without first identifying ID_{ID}, r_{ID} and x.

Therefore, the server would not be able to compute $sk = SK_{BA} = SK_{AB}$ without identifying ID_{ID}. In addition, the session key is different for the same user at different sign-in sessions. As a result, the proposed scheme is secured against the server deception attack. □

12) Resistance to the Parallel Session Attack.

Proof. Assuming that the attackers can, through replaying the sign-in request message $\{ID_A, R_A\}$, $\{ID_B, R_B, T_B\}$ turn themselves into an authorized user (U_i) within the valid time frame. However, in such a case, they would not be able to calculate $sk = SK_{BA} = SK_{AB}$ in the next step since the confirmation message does not contain all the data required for establishing the next steps. Because, the security of the proposed scheme authentication message against the parallel attack would depend on the complexity of the logarithmic calculations over $GF(p)$, one-way hash function, Elliptic Curve Groups and the Diffie–Hellman key agreement protocol. □

13) Resistance to the Insider Attack.

Proof. If an immune insider server obtains the confidential information $< abP, as_B P, bs_A P, R_{ID} >$, he would not be able to extract similar sensitive information $s_A s_B P, s_{ID}$ and $R_{ID} + H_1((ID_{ID} \| R_{ID}) P_{pub})$. Because it is computationally infeasible to invert the one-way hash function $h(0)$. In addition, solving a discrete logarithm problem has been a difficult task. The session key agreement also acts against the insider attack procedures. □

14) Resistance to the Replay Attack (Re-execution Attack).

Proof. We can assume the attackers have managed to impersonate the sign-in request message to replay the same sign-in message $\{ID_A, R_A\}$, $\{ID_B, R_B, T_B\}$ to the server. However, it would not be easy for the server to discover the replay attack through examining the protocol combines with the random numbers and timestamp. In this case, if the attacker re-executes an old message on the part of the server, then the server can easily discover the re-execution attack by comparing sign-in message with the current random number and timestamp. Therefore the proposed scheme is protected from the replay attack. □

B. Comparison with Existing Protocols.

One example of an identity-based authenticated key agreement protocol in the escrow mode is the protocol proposed by Chen and Kudla [5]. A drawback with this protocol (and also of Smart's identity-based authenticated key agreement protocol [18]) is that it does not provide perfect forward secrecy attribute. Although Shim [17] proposed a protocol that is claimed to provide such an attribute, it was later found to be vulnerable to the man-in-the-middle attack [19]. In 2005, Wang [21] proposed an identity-based authenticated key agreement protocol which achieves perfect forward secrecy in the escrow mode, it needs to do 3 exponentiation in G, one multiplication in G, and one pairing. Our protocol needs to do one exponentiation in G, 4 exponentiation in G_T, and one pairing. The computational efficiency of two schemes is almost the same. It is more efficient than that of Cao's scheme, because it can prevent to compromise session keys and the insider, replay, parallel session, reflection, server spoofing, and man in the middle attacks.

5 Conclusions

Perfect session-specific temporary information secrecy is an important security attribute for authenticated key agreement protocols (in both escrow and escrowless modes). We presented an identity-based authenticated key agreement protocol that is secure in the escrow mode. We demonstrated that our proposed protocol provides almost all of the known security attributes, especially the perfect session specific temporary information secrecy attribute with nice computational efficiency than reported other schemes.

References

[1] R. Canetti, H. Krawczyk, "Analysis of key exchange protocols and their use for building secure channels," in *Proceedings of the Advances in Cryptology (EUROCRYPT'01)*, LNCS 2045, Springer-Verlag, pp. 453-474, 2001.

[2] X. F. Cao, W. D. Kou, K. Fan, J. Zhang, "An identity-based authenticated key agreement protocol without bilinear pairing," *Chinese Journal of Electronics & Information Technology*, vol. 31. no. 5, pp. 1241-1244, 2009.

[3] X. F. Cao, W. D. Kou, X. N. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895-2903, 2010.

[4] T. Y. Chang, W. P. Yang, M. S. Hwang, "Simple authenticated key agreement and protected password

change protocol", *Computers & Mathematics with Applications*, vol. 49, pp. 703–714, 2005.

[5] L. Chen, and C. Kudla, "Identity based key agreement protocols from pairings," in *Proceedings of the 16th IEEE Computer Security Foundations Workshop*, pp. 219-213, 2002.

[6] A. Cilardo, L. Coppolino, N. Mazzocca, L. Romano, "Elliptic curve cryptography engineering," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 395-406, 2006.

[7] C. Gentry, "Practical identity-based encryption without random oracles," in *Proceedings of the EUROCRYPTO'06*, LNCS 4004, Springer-Verlag, pp. 445-464, 2006.

[8] 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, Jan. 2013.

[9] M. S. Hwang, S. Y. Hsiao, W. P. Yang, "Security on improvement of modified authenticated key agreement protocol," *Information - An International Interdisciplinary Journal*, vol. 17, no. 4, pp.1173–1178, Apr. 2014.

[10] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves", *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 73–84, 2004.

[11] 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.

[12] 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.

[13] T. K. Mandt, C. H. Tan, "Certificateless authenticated two-party key agreement protocols," in *Proceedings of the 11th Annual Asian Computing Science Conference (ASIAN'06)*, Secure Software and Related Issues, LNCS 4435, Springer-Verlag, pp. 37-44, 2008.

[14] N. McCullagh, and P. S. L. M. Barreto, "A new two-party identity-based authenticated key agreement," in *Proceedings of CT-RSA'05*, LNCS 3376, Springer-Verlag, pp. 262-274, 2005.

[15] H. H. Ou, M. S. Hwang, "Double delegation-based authentication and key agreement protocol for PCSs," *Wireless Personal Communications*, vol. 72, no. 1, pp. 437–446, Sep. 2013.

[16] A. Shamir, "Identity-based cryptosystems and signature schemes," in *CRYPTO'84*, LNCS 196, Springer, pp. 47–53, 1984.

[17] K. Shim, "Efficient ID-based authenticated key agreement protocol based on the Weil pairing," *Electronics Letters*, vol. 9, no. 8, pp. 653-654, 2003.

[18] N. P. Smart, "An identity based authenticated key agreement protocol based on the Weil pairing," *Electronics Letters*, vol. 38, no. 13, pp. 630-632, 2002.

[19] H. Sun and B. Hsieh, "Security analysis of Shim's authenticated key agreement protocols from pairings," *Cryptology ePrint Archive*, Report 2003/113, 2003. (`http://eprint.iacr.org/2003/113`)

[20] S. B. Wang, Z. F. Cao, and X. L. Dong, "Provably secure identity-based authenticated key agreement protocols in the standard model," *Chinese Journal of Computers*, vol. 30, no. 10, pp. 1842-1854, 2007.

[21] Y. Wang, "Efficient identity-based and authenticated key agreement protocol," *Cryptology ePrint Archive*, Report 2005/108, 2005.

Biography

Javad Saadatmandan received his Ph.D from the Department of Mathematics and Computer Sciences at Qom University in Qom, Iran. He is presently an assistant professor of mathematics at IAU, Qom, Iran. His research interest include cryptographic protocols and wavelet transforms.

Amir Hossein Rahimi received the B.Sc. degree in 2009 in applied mathematics from the University of Arak, Iran, and the M.Sc. degrees in 2014 in cryptography engineering from Malek-Ashtar University of Technology Isfehan. His current research interests include areas of communication theory, information security, cryptography, smart grid, steganography, digital signature and authentication protocols. He has published more than 10 papers in the fields mentioned. Also, he have been teaching mathematical sciences in universities of Qom province, Iran form 2015 until now. Email:Amir.Rahimi361@Gmail.Com

Identity Based Key-Insulated Encryption with Outsourced Equality Test

Seth Alornyo, Yanan Zhao, Guobin Zhu, and Hu Xiong
(Corresponding author: Yanan Zhao)
School of Information and Software Engineering, University of Electronic Science and Technology of China
Sichuan-Chengdu
(Email: zynbyxz@gmail.com)
(Received Oct. 2, 2018; Revised and Accepted Mar. 23, 2019; First Online June 11, 2019)

Abstract

We firstly combine the concepts of key-insulated encryption (KIE) and identity-based encryption with the equality test (IBE-ET) to obtain identity-based key-insulated encryption with equality test (IB-KIEET). The scheme inherits the advantages of identity-based encryption (IBE), which simplifying certificate management for public key encryption. Furthermore, the key-insulated mechanism was added in our scheme, which perfectly reduced the possibility of key exposure. Our scheme achieves weak indistinquishable identity chosen ciphertext (W-IND-ID-CCA) security in the random oracle model. Meanwhile, it is indicated that our scheme is feasible and practical through the experimental simulation and theoretical analysis.

Keywords: Identity Based Encryption; Key-Insulated; Outsourced Equality Test

1 Introduction

Due to the rapid popularity of cloud computing, storing data in the cloud (such as photos, videos, emails, and instant messages) has become a trend for individuals and organizations [5, 20]. However, the cloud server cannot be fully trusted to ensure the confidentiality of user data uploaded to the cloud [16]. For this reason, user's data should be encrypted before sending it to the cloud server. Public key encryption seems to be suitable for encryption [1]. But it is unrealistic for users to download all the data from the cloud server each time. Therefore, it is desirable to design a scheme that supports the search function stored on the ciphertext in the cloud server without revealing any information related to these ciphertexts.

Boneh *et al.* [3] proposed the first public key encryption using keyword search (PKE-KS). In the PKE-KS scheme, the user can encrypt the keyword and corresponding data under the user's public key, meanwhile, the user creates a target keyword trapdoor by using his/her private key and then uploads it to the cloud server. Nonetheless, the cloud server can only compare keywords with trapdoors under the same public key. This has become the bottleneck for the development of keyword search. To address this problem, Yang *et al.* [28] proposed the concept of public key encryption scheme (PKE-ET) with equality test based on bilinear pairing. Compared to PKE-KS, the equality test in PKE-ET can be performed between two ciphertexts encrypted in the same public key and different public keys.

Following the works of Yang *et al.* [28], some well-designed schemes with equality test have been constructed [11, 15, 21, 26]. Recently, Sha Ma [18] proposed the notion of identity based encryption with outsourced equality test(IBE-ET) in cloud computing. The above-mentioned scheme is the first time to integrate identity-based cryptosystem into public key encryption with equality test, thus it inherits the advantages of both primitives. However, the problem caused by key exposure can't be resisted in this scheme. There is no doubt that key exposure will lead to the destructive consequence, for which Dodis *et al.* [6] proposed the primitive of key-insulated. In their scheme, the secret keys consist of two parts which named user secret key and helper key. The user secret key has been constantly changing, so the possibility of key exposure is significantly reduced. Therefore, a scheme need to be devised that satisfies both the equality test and the key-insulated encryption.

1.1 Related Work

1.1.1 Key-insulated Encryption

In order to reduce the damage which is caused by private key-exposure, Dodis *et al.* [6] firstly introduced the key-insulated encryption. Nevertheless, in this scheme, the total time period number should be determined in advance. Since then, many research results, about key-insulated encryption have been put forward. By introducing the concept of proxy re-encryption, Wang *et al.* [22] processed a key-insulated proxy re-encryption scheme (KIPRE). He *et al.* [8] combined key-insulated encryption with certificate-less public key encryption (CL-PKE) and present a con-

crete paradigm which is called certificateless key-insulated encryption scheme (CLKIE). Hanaoka *et al.* [7] combined identity-based encryption with key-insulated encryption and proposed the first identity-based key-insulated encryption scheme. Later, Bellare and Palacio [2] proposed a new key-insulated encryption scheme. In this scheme, the total time period number doesn't need to be given in advance. Benoît *et al.* [13] processed a identity-based key-insulated encryption scheme without random oracles.

1.1.2 Equality Test

Boneh *et al.* [4] proposed the first public key encryption with keyword search (PKE-KS) scheme. In this scheme, user is able to test the equvalance between two ciphertexts which are encrypted with the same public key. Later, some well-designed PKE-KS schemes were put foward [9, 27, 29]. However, it is unable for user to conduct search functionality for ciphertexts under different public keys. In order to solve this problem, Yang *et al.* [28] presented public key encryption with equality test (PKE-ET). This scheme allows user to search the ciphertexts in different public keys. After that a large amount of schemes corresponding to PKE-ET have been put forward [4,14,19,30]. Although PKE-ET has excellent performance, there are still some problem on key certificate management, which seriously constrain the efficient in practice. To solve this problem, Ma [18] combined PKE-ET and (identity-based encryption) IBE [3, 23] and proposed the first identity-based encryption with equality test (IBE-ET). Different from PKE-ET, IBE-ET solved the problem of key certificate management. In recent year, a series of schemes which focus on IBE-ET have been published. Wu *et al.* [24] presented a dual server IBE-ET which can resist the inner keywords guessing attack. Recently, in order to provide a scheme which achieves IND-ID-CCA security, Lee *et al.* [10] proposed a semi-generic construction of IBE-ET. Unfortunately, IBE-ET can not reduce the damage caused by private key-exposure. So far, there has not been any scheme which can solve private key-exposure problem.

1.2 Our Contribution

To resolve these challenges, we propose identity based key-insulated encryption with equality test (IB-KIEET) in this paper. To summarize, our contribution to this paper consist of three points:

1) We first incoporate the idea of identity-based key-insulated encryption into IBE-ET to propose the IB-KIEET scheme. Specifically, IB-KIEET enables the cloud server to conduct an equivalence test on ciphertext. Meanwhile, IB-KIEET can resist private key exposure;

2) Our scheme achieves Weak-IND-ID-CCA (W-IND-ID-CCA) security, which can prevent an insider attack.

3) Finally, we give the experimental simulation and theoretical analysis which can indicate the feasibility and practicability of our scheme.

1.3 Organization

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

2 Preliminaries

2.1 Billinear map

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

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

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

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

2.2 Bilinear Diffie-Hellman (BDH) problem

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

$$ADV_{\mathbb{A}}^{BDH} Pr[\mathbb{A}(g, g^a, g^b, g^c) = e(g, g)^{abc}]$$

We say that the BDH assumption holds if for any polynomial-time algorithm $\mathbb{A}$, its advantage $Adv_{\mathbb{A}}^{BDH}$ is negligible.

2.3 Definitions

In this section, we give formal definitions of our scheme. A physically secured helper device is employed in our model to help update user secret key at a time i,we assume our helper device is secured. Our scheme achieves weak chosen ciphertext security (*i.e.* W-IND-ID-CCA) under the defined security model.

Identity based key-insulated encryption with outsourced equality test (IB-KIEET): In identity based encryption with equality test against outsider attack scheme, we specify nine algorithms: Setup, Extract,

UserKeyGeneration, DeviceKeyUpdate, UserKeyUpdate, Trapdoor,Encrypt, Decrypt, Test, where $\mathbb{M}$ and $\mathbb{C}$ are its plaintext space and ciphertext space, respectively:

1) **Setup**(λ): It takes as input a security parameter λ , total number of time period $T = N$ and returns the public system parameter K and the master key msk.
2) **Extract**(msk,ID): It takes as input, msk, an arbitrary ID$\in \{0,1\}^*$, system parameter K and returns a secret key dk_{ID} to the user with identity ID. This algorithm is also performed by a PKG. After the algorithm is performed, PKG sends to the user with identity ID via a secure channel.
3) **UserKeyGeneration**(K, N, dk_{ID}): The user key generation algorithm takes the received secret key dk_{ID} and the total number of time periods N.The algorithm outputs user's master private key dk^*_{ID} and set user's initial secret key dk^0_{ID}
4) **DeviceKeyUpdate**(i, j, dk^*_{ID}): The physically secure device takes as input indices i, j for the time periods ($1 \leq i, j \leq N$) and a master private key dk^*_{ID}.It outputs a partial secret key $dk'^{i,j}_{ID}$.
5) **UserKeyUpdate**($i, j, dk^i_{ID}, dk'^{i,j}_{ID}$): It takes as input indices i, j, a secret key dk^i_{ID}, and a partial secret key $dk'^{i,j}_{ID}$.It returns the secret key dk^j_{ID} for time period j.
6) **Trapdoor** (msk,ID,): It takes as input msk and an arbitrary $ID \in \{0,1\}^*$ and returns a trapdoor td for that identity.
7) **Encrypt**(K,i,ID,m): It takes as input K, the index i of the current time period N, an identity ID $\in \{0,1\}^*$ and a plaintext $m \in M$, and returns a ciphertext c as $c = (i, c)$, where $c \in C$.
8) **Decryption**(dk^i_{ID}, i, c): It takes a current private secret key dk^i_{ID} and a ciphertext (i, c) as inputs and returns a plaintext $m \in M$ or a symbol $\perp$ if the ciphertext is invalid.
9) **Test**(C_A, C_B)**:** It takes ciphertext C_A and C_B produced by user A and user B respectively. It output 1 if message associated with C_A and C_B are equal. It outputs 0 otherwise.

Correctness: The algorithm must satisfy the following conditions:

1) When dk^i_{ID} is updated secret decryption key generated by the physically secure DeviceKeyUpdate algorithm given ID as the public key, then

$$\forall m \in M : Decrypt(C, dk^i_{ID}) = M,$$

where $C = Encrypt(ID, M)$ and $C = (i, c)$.

2) When td_A and td_B are trapdoors generated by trapdoor algorithm given ID_A and ID_B as the public keys, then

$$\forall M \in M : Test(C_A, td_A, C_B, td_B) = 1,$$

where C_A= Encrypt(ID_A, M) and C_B= Encrypt(ID_B, M).

3) When td_A and td_B are trapdoors generated by trapdoor algorithm given ID_A and ID_B as the public keys, then

$$\forall M, M^{'} \in M \text{ and } M \neq M^{'},$$

$$Pr[Test(C_A, td_A, C_B, td_B) = 1]$$

is negligible where C_A = Encrypt (ID_A, M) and C_B = Encrypt $(ID_B, M^{'})$.

Security Models:

1) **Setup**: The challenger takes a security parameter λ as input and runs the setup algorithm. It gives the system parameters K to the adversary $\mathbb{A}$ and keeps the master key msk by itself.
2) **Phase 1**: Private decryption key queries (ID_a): The challenger runs the Extract algorithm to generate the private decryption key dk^i_a corresponding to the public key ID_a. It sends dk^i_a to $\mathbb{A}$.
3) **Trapdoor queries** ID_a. The challenger runs the above private decryption key queries on ID_a to get $dk_{ID,a}$ and then generates the trapdoor td_a using $dk_{ID,a}$ via Trapdoor algorithm. Finally, it sends (td_a) to $\mathbb{A}$.
4) **Decryption queries** $(ID_a, (i, C))$**:** The challenger runs the Decryption algorithm to decrypt the ciphertext (i, C_a) by running Extract algorithm to obtain the private secret key $dk^i_{ID,a}$ corresponding to the public key ID_a. Finally, it sends the plaintext M_a to $\mathbb{A}$.
5) **Challenge**: $\mathbb{A}$ submits an identity ID_{ch} on which it wishes to be challenged. The only constraint is that ID_{ch} did not appear in private decryption key queries in Phase 1 but ID_{ch} may appear in trapdoor queries in Phase 1 or in decryption query ID_{ch}. The challenger randomly chooses a plaintext m $\in$ M and sets C^*= Encrypt(ID_{ch}, m, tok^*_{ID}). Finally, it sends C^* to $\mathbb{A}$ as its challenge ciphertext.
6) **Phase 2**: Private decryption key queries ID_a where $ID_a \neq ID_{ch}$. The challenger responds in the same way as in Phase 1.
7) **Trapdoor queries** ID_a. The challenger responds in the same way as in phase 1.
8) **Decryption queries** $(ID_a, C_i) \neq (ID_{ch}, C^*)$. The challenger responds in the same way as in Phase 1.

9) **Guess**: $\mathbb{A}$ submits a guess $m^{'} \in$ M.

Definition 1. *The scheme is W-ID-CCA secure if for all W-IND-ID-CCA adversaries,* $\boldsymbol{Adv}_{IB-KIEET,A}^{W-ID-CCA}(K) = Pr[m = m^{'}]$ *is negligible.*

3 Construction

We provide a detailed construction for the IB-KIEET in this section as follows:

1) **Setup**(l^{λ},N): Initially, the system takes a security parameter λ, a time period N and returns public system parameters K, the master secret key msk.

   - The system generates two multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ with the same orime order p of λ length bits and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The system selects an arbitrary generator $g \in \mathbb{G}$.
   - The algorithm exploit a keyed permutation $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ for a positive integers, K=$k(\lambda)$ and n=$n(\lambda)$. Set a random value k_1 from $\{0,1\}$. Generate a MAC scheme MAC = (GSV) and obtain k_2 by running $G(\lambda)$. Set the master token key $MTK = (k_1, k_2)$. We adopted Lee et al.'s work to resist against insider attack.
   - The system chooses three hash functions: $H_1 : \{0,1\}^t \to Z_p^*, H_2 : \{0,1\}^* \to \mathbb{G}, H_3 : T \times \mathbb{G} \times \mathbb{G}_T \to [0,1]^{t+l}$, where l is the length of random numbers and t is the length of messages. The algorithm randomly picks (α, β) and set g_1=g^{α}, g_2=g^{β}.Next,picks random elements $\{g_3, h, h_1, \ldots, h_N\} \in \mathbb{G}$. It publishes public parameter K=$(T, p, \mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, g_3, h_{N-1}, MAC, H_1, H_2, H_3)$ and MSK =(α, β).T is referred to as a MAC Tag.

2) **Extract** (K,MSK,ID): For a given string ID $\in \{0,1\}^*$, public parameter K and MSK,the algorithm compute $h_{ID} = H_1(ID) \in \mathbb{G}$, set master decryption key $mdk_{ID} = (h_{ID}^{\alpha}, h_{ID}^{\beta})$ where (α, β) is the master secret key.

3) **UserKeyGen**(K,mdk,ID): On input mdk_{ID}, the algorithm set $dk_{ID} = (h_{ID}^{\alpha^{(1)}}, h_{ID}^{\alpha^{(2)}})$ where $(\alpha^{(1)}, \alpha^{(2)}) \in Z_p^*$ and parse it as $dk_{ID} = (dk_{ID}^{(1)}, dk_{ID}^{(2)})$, chooses a random elements $\eta \in \mathbb{G}$ and set $dk_{ID}^* = (dk_{ID}^{(1)}/\eta, dk_{ID}^{(2)})$, and set user's initial decryption key as $dk_{ID}^0 = (\eta, \phi, \phi, \phi)$.

4) **DevKeyUpdate**(i, j, dk_{ID}^*): On input two indices i, j and dk_{ID}^*, parse dk_{ID}^* as $(dk_{ID}^{*(1)}, dk_{ID}^{*(2)})$, choose $t \in \mathbb{Z}_q^*$ and return a partial secret key $dk_{ID}^{\prime i,j} = (dk_{ID}^{*(1)}.h_j^t, dk_{ID}^{*(2)}, g^t)$.

5) **UserKeyUpdate**(i, j, dk_{ID}^*): The algorithm on input indices i, j, a secret key dk_{ID}^i and a partial secret key $dk_{ID}^{\prime i,j} = (x, y, z)$ parse $dk_{ID}^i = (dk_{ID}^{i(1)}, dk_{ID}^{i(2)}, dk_{ID}^{i(3)}, dk_{ID}^{i(4)})$. The algorithm output $dk_{ID}^j = (dk_{ID}^{j(1)}, dk_{ID}^{j(2)}, dk_{ID}^{j(3)}, dk_{ID}^{j(4)})$ where $dk_{ID}^{j(1)} = dk_{ID}^{i(1)}$ and $dk_{ID}^{i(1)} = \eta$ for all i. Therefore $dk_{ID}^{j(2)} = dk_{ID}^{i(1)}.x$,$dk_{ID}^{j(3)} = y$,$dk_{ID}^{j(4)}$=z. The algorithm send $(dk_{ID}^i, dk_{ID}^{\prime i,j})$ via a secure channel to the user. A new secret key computed at a period i is used to decrypt a specific ciphertext corresponding to period i. If $(dk_{ID}^i, dk_{ID}^{\prime i,j})$ is deleted as a result of the key update, then ciphertext stored on the cloud server at a period i could not be decrypted by the user. Other similar key-insulated schemes deleted previous secret keys when the current key was updated to a new secret key.

6) **Trapdoor** (ID): For a given string ID $\in \{0,1\}^*$ the algorithm computes $h_{ID} = H_1(ID) \in \mathbb{G}$ and set the trapdoor $td_{ID} = h_{ID}^{\beta}$, td_{ID} is the second element of mdk_{ID}.

7) **Encrypt**(K, ID,m): To encrypt m with a public ID, algorithm selects two random numbers $r_1, r_2 \in Z_p^*$.Then it computes:

$$\begin{aligned} C_1 &= g^{r_1}, \\ C_2 &= W^{r_1}.H_2(e(g_2, h_{ID})^{r_1}) \end{aligned}$$

where

$$\begin{aligned} W^{r_1} &= F(k_1, H(m)), \\ C_3 &= g^{r_2} \\ C_4 &= (m \parallel r_1) \oplus H_3(C_1 \parallel C_2 \parallel P \parallel e(g_1, h_{ID})^{r_2}). \end{aligned}$$

Finally it returns $C = (C_1, C_2, C_3, C_4)$, where $P \leftarrow S(k_2, C_3)$ for the signing algorithm S of the employed MAC, the corresponding tag P is used to verify C_3. The function F is assumed to be a strong pseudo-random permutation and the MAC is existentially unforgeable under chosen message attack.

8) **Decrypt**(C, dk_{ID}, tok_{ID}):On input the ciphertext C, updated secret key dk_{ID}^i and a token $tok_{ID} = (k_1, k_2)$, the algorithm computes:

$$\begin{aligned} m' \parallel r' &= C_4 \oplus H_3(C_1 \parallel C_2 \parallel P \parallel e(C_3, dk_{ID}^i)), \\ m' \parallel r' &= H_3(e(C_3, dk_{ID}^i)). \end{aligned}$$

Given $P \leftarrow S(k_2, C_3)$ where $P = MAC_{k_2}(C_3)$, the algorithm verify:

$$P' = MAC_{k_2}(C_3) \text{ if } P' = P.$$

Then it checks whether $C_1 = g^{r_1'}$ and C_2=$W^{r_1'}.H_2(e(C_1, h_{ID}^{\beta}))$ where $W^{r_1'}$=$F(k_1, H(m'))$. If both holds,the algorithm return m'. Otherwise, return $\perp$.

9) **Test**$(C_A, td_{ID_A}, C_B, td_{ID_A})$: On input a ciphertext C_A, trapdoor td_A and a given senders' ciphertext C_B. The algorithm test whether M_A=M_B by computing:

$$T_A = \frac{C_{2,A}}{H_2(e(C_{1,A}, td_{ID,A}))},$$
$$T_B = \frac{C_{2,B}}{H_2(e(C_{1,B}, td_{ID,B}))}$$

the algorithm outputs 1 if the equation holds, outputs 0 otherwise.

Correctness: The conditions that satisfies the above definitions are shown below:

1) Assuming a well-formed ciphertext for ID_A and ID_B. Given the following:

$$\begin{aligned} T_A &= \frac{C_{2,A}}{H_2(C_{1,A}, td_{ID,A})}, \\ &= \frac{W_A^{r_{1,A}}.H_2(e(g_A^{r_1}, h_{ID,A}^{\beta})}{H_2(e(g_A^{r_1}, h_{ID,A}^{\beta})}, \\ &= W_A^{r_{1,A}} \\ T_B &= \frac{C_{2,B}}{H_2(C_{1,B}, td_{ID,B})} \\ &= \frac{W_B^{r_{1,B}}.H_2(e(g_B^{r_1}, h_{ID,B}^{\beta})}{H_2(e(g_B^{r_1}, h_{ID,B}^{\beta})} \\ &= W_B^{r_{1,B}} \end{aligned}$$

It output 1 if the following equation holds. Otherwise output 0.

$$e(C_{1,A}, T_B) = e(C_{1,A}, T_A).$$

Therefore,

$$\begin{aligned} e(C_{1,A}, T_B) &= e(g^{r_{1,A}}, W_B^{r_{1,B}}) = e(g, W_B)^{r_{1,A} r_{1,B}} \\ e(C_{1,B}, T_A) &= e(g^{r_{1,B}}, W_A^{r_{1,A}}) = e(g, W_A)^{r_{1,A} r_{1,B}} \end{aligned}$$

Where $W_A^{r_1} = F(k_1, m_A)$ and $W_B^{r_1} = F(k_1, m_B)$, given token tok_{ID}=k_1, the function outputs M_A and M_B.If W_A=W_B, then $e(C_{1,A}, T_B) = e(C_{1,B}, T_A)$. Test $(C_A, td_{ID,A}, C_B, td_{ID,B})$ outputs 1.

2) For any $M_A \neq M_B$, Test $(C_A, td_{ID,A}, C_B, td_{ID,B})$= 1, this implies that $e(g, W_A)^{r_{1,A}} = e(g, W_B)^{r_{1,B}}$. Hence $Pr[e(g, W_A)=(g, W_B)] = \frac{1}{P}$. Therefore, we assume that $Pr[Test(C_A, td_{ID,A}, C_B, td_{ID,B}) =1]$ is negligible.

4 Security Analysis

Theorem 1. *The Above IB-KIEET Scheme is W-IND-ID-CCA Secure in the Random Oracle Model Assuming BDHP is negligible.*

Proof. Let $\mathcal{A}$ be a PPT adversary attacking the W-IND-CCA security of the above scheme. Suppose that $\mathbb{A}$ runs in time T and makes at most q_H hash queries and q_D decryption queries. Let $Adv_A^{W-IND-CCA}(t, q_H, q_D)$ denote the advantage of $\mathbb{A}$ in the W-IND-ID-CCA experiment. The security proof is done through a sequence of games by [28]. The preliminaries of the original game is considered as follows:

Game G_0 $\alpha \leftarrow Z_q^*$, y=g^{α}, T=N, R=$\emptyset$;

$m \leftarrow G_1, r \leftarrow Z_p^*, U^*$=$g^r, V^*$=$m^r$, W^* =$H(T, U^*, V^*, y^r) \oplus (m \parallel r)$;

$m \leftarrow A^{o^{H,o_2}}(T, U^*, V^*, W^*)$, where the oracle works as follows:

O_H: On input a triple $(T, U, V, Y) \in G_1^4$,where a same random value is returned, if the same input is asked multiple times, the same answer will be returned.

O_2: On input a ciphertext (T,U,V,W), it returns the decryption algorithm to decrypt it using the secret key α given within a time N.

Let X_o be the event that $m^{'}$=m in Game G_0. However the probability in Game G_0 is Pr $[S_o]$. Hence we modify Game G_0 and obtain the following game.

Game G_1 $\alpha \leftarrow Z_q^*$, y=g^{α}, T=N, R=$\emptyset$;

$m \leftarrow G_1, r \leftarrow Z_p^*, U^*$=$g^r, V^*$=$m^r, R^* \rightarrow [0,1]^{t+i}, W^*$= $H(T, U^*, V^*, y^r) \oplus (m \parallel r), R$ =$R \cup (T, U^*, V^*(U^*)^{\alpha}, R^*)$;

$m \leftarrow A^{O_H, O_2}(y, T, U^*, V^*, W^*)$, where the oracle works as follows:

O_H: On input a triple $(T, U, V, Y) \in G_1^4$ where if there is an entry (T, U, V, Y, h) in the hash table R, h is returned, otherwise a random value h is selected and returned, and (T, U, V, Y, h) is added to R.

O_2: On input a ciphertext (T, U, V, W), a hash query on (T, U, V, U^{α}) is issued. Suppose the answer is $h \in [0,1]^{t+i}$, then $m \parallel r$ is computed as $h \oplus W$,then a validity check on whether U=g^r and V=m^r is performed. If the check fails, $\perp$ is returned: otherwise, m is returned. The event that $Game_1$ occurs is denoted by S_1. However its observed that G_0=G_1, hence we deduce the probability of the random oracle as:

$$Pr[S_1] = Pr[S_0].$$

In the next game, we further modify the simulation game in an indistinguishable way:

Game G_2 $\alpha \leftarrow Z_q^*$, y=g^{α}, T=N, R=$\emptyset$;

$m \leftarrow G_1, r \leftarrow Z_p^*, U^*$=$g^r, V^* = m^r, W^* \rightarrow [0,1]^{t+i}, R = R \cup (t, U^*, V^*(U^*)^{\alpha}, W^*)$;

$m \leftarrow A^{O_H,O_2}(y,T,U^*,V^*,W^*)$. The oracle response to queries as follows:

O_H: Game G_2 is identical to Game G_1. However if Adversary queries for $(U*,.,(U*)^\alpha)$, then the game is aborted. Let ε be this event.

O_2: This is also the same as Game G_1, however if Adversary ask for decryption of (U^*,V^*W), where $W^{'} \neq W^*$, $\perp$ is retuned.

Chosen Ciphertext security (CCA) secure is paramount in this game because W^* is a random value in both Games, however the random oracle responds are unique and probabilistic because W^* is dependent on U and V^*. The probability of $\perp$ occurring is negligible.

In the next game, we further modify the simulation game in a time T based indistinguishable way.

Game G_3 $\alpha \leftarrow Z_q^*, y = g^\alpha, T = N, R = \emptyset$;

$m \leftarrow G_1, r \leftarrow Z_p^*, U^*{=}g^r, V^*{=}m^r, W^* \rightarrow [0,1]^{t+i}, R{=}\ R \cup (T,U^*,V^*(U^*)^\alpha,W^*)$;

$m \leftarrow A^{O_H,O_2}(y,T,U^*,V^*,W^*)$:

O_H: Game G_3 is identical to Game G_2 . However if Adversary queries for $(U^*,T,U^*,.,(U^*)^\alpha)$, then the game is aborted. Let ε_1 be this event.

O_2: This is also the same as Game G_2, however if Adversary ask for decryption of (U^*,V^*,T), where $T^{'} \neq T$, $\perp$ is retuned.

The timestamp associated with the ciphertext improve the security of this game. T is a tampstamp value associated with the ciphertext in both Games, however the random oracle responds are unique and probabilistic because decrption queries are dependent on T, U^* and V^*. The probability of $\perp$ occurring is negligible.

The challenge ciphertext generated in this game is identically distributed to that in Game G_2 and G_3 as W^* is a random value in both Game G_2 and Game G_3. The simulation of O_2 is secure since W^* is uniquely determined by U^* and V^* in Game G_2 and U^*, V^*, T in Game G_3. Therefore, if event ε_1 does not occur, Game G_3 is identical to Game G_1. However, we show below that event ε_1 occurs with negligible probability.

We further simulates decryption queries in indistinquishable way from Game G_3. The decryption queries are separated into two types which includes:

Type 1: (T,U,V,U^α) has been queried to O_H before a decryption query (T,U,V,W) is issued. In this case, W is uniquely determined after (T,U,V,U^α) is queried to O_H. So the decryption oracle is simulated perfectly.

Type 2: (U,V,U^α) has never been queried to O_H when a decryption query (U,V,W) is issued. In this case, $\perp$ is returned by the decryption oracle. The simulation fails if (U,V,W) is a valid ciphertext. However, this happens with negligible probability.

□

5 Comparison

In this section, we compare the efficiency of algorithms and time consumption among the proposed scheme, Ma's [18] scheme, which combined the concepts of public key encryption with equality test and identity-based encryption, Wu *et al.*'s [25] scheme,which solved the problem of the insider attack, and Li *et al.*'s [12] scheme,in which a key-insulation cryptosystem was proposed in order to minimize the damage of secret key exposure. The comparison result of efficiency is shown in Table 1, which includes Outsider Attack(OA), Insider Attach(IA), encryption(Enc), decryption(Dec), Test and Security. The above comparison shows that our scheme can resist both OA and IA, whereas others' don't have this ability. In addition, the scheme in [18, 25] as well as our scheme implement chosen ciphertext security, which is stronger than chosen plaintext security achieved in [12].

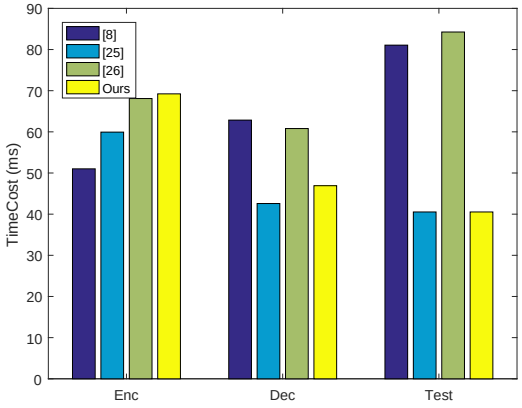


Figure 1: Computation overhead of different schemes

In order to evaluate the computation efficiency of these schemes, the Pairing-Based Cryptography(PBC) Library [17] is used to quantify the time consumption of encryption, decryption and test operations. This experiment is executed on windows 7 OS equipped with an i5-4460 CPU @3.2 GHz and 4G bytes memory. The time consumptions, which are obtained by repeat simulations, are shown in Figure 1. From Figure 1 we can observe that the computation cost of decryption and test of our scheme is comparable with other existing works, whereas our encryption computational cost seems higher. This is forgivable due to the additional computation overheads required to prevent both insider and outsider attacks, which, however, is not the case in other works. In the aspect of the computation cost of decryption and test, our scheme is better than schemes in [12, 25]. Although time consumption of decryption and test operations of our

Table 1: Comparing the efficiency of algorithm of variant PKE-ETs with our scheme

SCHEME	OA	IA	Enc	Dec	Test	Security
[18]	N	N	$4Exp_1 + 2Exp_2$	$2P+2Exp_1$	4P	OW-ID-CCA
[25]	N	Y	$1P+3Exp_1 + 1Exp_2$	$1P+2Exp_1$	2P	W-IND-ID-CCA
[12]	Y	N	$1P+4Exp_1 + 1Exp_2$	3P	$4P+1Exp_2$	IND-ID-CPA
Ours	Y	Y	$2P+2Exp_1 + 2Exp_2$	$2P+2Exp_1$	2P	W-IND-ID-CCA

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

scheme is slightly high than scheme proposed in [25], it provides additional security for outsider attack.

6 Conclusions

Inspired by the notion of scheme in [18], we put forward identity-based key-insulated encryption with outsourced equality test scheme. In this paper, the mechanism of key-insulated is used to reduce the damage to private key exposure. Besides, our scheme also has the ability to resist insider attack from HBC server, which makes it is practical and suitable in cloud computing. Finally, our scheme security is proved in the random oracle. Theoretical analysis and experiment simulation both demonstrate that our scheme is secure and efficient.

7 Acknowledgements

This work was supported in part by the 13th Five-Year Plan of National Cryptography Development Fund for Cryptographic Theory of China under Grant MMJJ20170204, in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2016J091, the Guangxi Colleges and Universities Key Laboratory of Cloud Computing and Complex Systems, and in part by the Natural Science Foundation of China under Grants U1401257, 61472064 and 61602096.

References

[1] D. S. AbdElminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40–48, 2018.

[2] M. Bellare and A. Palacio, "Protecting against key-exposure: Strongly key-insulated encryption with optimal threshold," *Applicable Algebra in Engineering, Communication and Computing*, vol. 16, no. 6, pp. 379–396, 2006.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Annual International Cryptology Conference*, pp. 213–229, 2001.

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

[5] X. F. Cao, H. Li, L. J. Dang, and Y. Lin, "A two-party privacy preserving set intersection protocol against malicious users in cloud computing," *Computer Standards & Interfaces*, vol. 54, pp. 41–45, 2017.

[6] Y. Dodis, J. Katz, S. H. Xu, and M. Yung, "Key-insulated public key cryptosystems," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 65–82, 2002.

[7] Y. Hanaoka, G. Hanaoka, J. Shikata, and H. Imai, "Identity-based hierarchical strongly key-insulated encryption and its application," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 495–514, 2005.

[8] L. B. He, C. Yuan, H. Xiong, and Z. G. Qin, "An efficient and provably secure certificateless key insulated encryption with applications to mobile internet," *Internationl Journal Network Security*, vol. 19, no. 6, pp. 940–949, 2017.

[9] S. T. Hsu, C. C. Yang, and M. S. Hwang, "A study of public key encryption with keyword search," *Internationl Journal Network Security*, vol. 15, no. 2, pp. 71–79, 2013.

[10] H. T. Lee, S. Ling, J. H. Seo, and H. X. Wang, "Semi-generic construction of public key encryption and identity-based encryption with equality test," *Information Sciences*, vol. 373, pp. 419–440, 2016.

[11] H. T. Lee, H. X. Wang, and K. Zhang, "Security analysis and modification of id-based encryption with equality test from acisp 2017," in *Australasian Conference on Information Security and Privacy*, pp. 780–786, 2018.

[12] J. Li, F. G. Zhang, and T. M. Wang, "A strong identity based key-insulated cryptosystem," in *In-*

ternational Conference on Embedded and Ubiquitous Computing, pp. 352–361, 2006.

[13] B. Libert, J. J. Quisquater, and M. Yung, "Parallel key-insulated public key encryption without random oracles," in *International Workshop on Public Key Cryptography*, pp. 298–314, 2007.

[14] X. J. Lin, L. Sun, and H. P. Qu, "Generic construction of public key encryption, identity-based encryption and signcryption with equality test," *Information Sciences*, vol. 453, pp. 111–126, 2018.

[15] H. Lipmaa, "Verifiable homomorphic oblivious transfer and private equality test," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 416–433, 2003.

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

[17] B. Lynn, "The stanford pairing based crypto library," *Privacy Preservation Scheme for Multicast Communications in Smart Buildings of the Smart Grid*, 2013. (`https://blog.csdn.net/vingstar/article/details/17113155`)

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

[19] H. P. Qu, Z. Yan, J. L. Lin, Q. Zhang, and L. Sun, "Certificateless public key encryption with equality test," *Information Sciences*, vol. 462, no. 76–92, 2018.

[20] S. Rezaei, M. Ali Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115–131, 2018.

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

[22] Y. L. Wang, D. J. Yan, F. G. Li, and H. Xiong, "A key-insulated proxy re-encryption scheme for data sharing in a cloud environment," *Internationl Journal Network Security*, vol. 19, no. 4, pp. 623–630, 2017.

[23] B. Waters, "Efficient identity-based encryption without random oracles," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 114–127, 2005.

[24] L. B. Wu, Y. B. Zhang, K. K. R. Choo, and D. B. He, "Efficient and secure identity-based encryption scheme with equality test in cloud computing," *Future Generation Computer Systems*, vol. 73, pp. 22–31, 2017.

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

[26] L. B. Wu, Y. B. Zhang, K. K. R. Choo, and D. B. He, "Efficient identity-based encryption scheme with equality test in smart city," *IEEE Transactions on Sustainable Computing*, vol. 3, no. 1, pp. 44–55, 2018.

[27] P. Xu, H. Jin, Q. H. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," *IEEE Transactions on computers*, vol. 62, no. 11, pp. 2266–2277, 2013.

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

[29] Y. Yu, J. B. Ni, H. M. Yang, Y. Mu, and W. Susilo, "Efficient public key encryption with revocable keyword search," *Security and Communication Networks*, vol. 7, no. 2, pp. 466–472, 2014.

[30] K. Zhang, J. Chen, H. T. Lee, H. F. Qian, and H. X. Wang, "Efficient public key encryption with equality test in the standard model," *Theoretical Computer Science*, vol. 755, pp. 65-80, 2019.

Biography

Seth Alornyo received his Master of Philosophy(M.Phil) degree from Kwame Nkrumah University of Science and Technology in 2014. Currently, he is pursuing his Ph.D. in Software Engineering at University of Electronic Science and Technology of China. His research interests lie in the area of Public Key Encryption and Network Security.

Yanan Zhao is currently pursuing her M.S. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China. She received her B.S. degree from Jiangxi University of Science and Technology in 2017. Her research interests include identity-based public key cryptography.

Guobin Zhu is an assistant professor at University of Electronic Science and Technology of China (UESTC). He received his Ph.D degree from UESTC in 2014. His research interests include: network security and applied cryptography.

Hu Xiong received his Ph.D. degree from University of Electronic Science and Technology of China (UESTC) in 2009.He is now a professor in the UESTC.His research interests include Cryptography and ad hoc network security

Using Parametric t-Distributed Stochastic Neighbor Embedding Combined with Hierarchical Neural Network for Network Intrusion Detectione

Huijun Yao[1], Chaopeng Li[2], and Peng Sun[3]
(Corresponding author: Chaopeng Li)
Jiangsu Cable Technology Research Institute Co.Ltd[1]
Nanjing, Jiangsu Province, 210001, China
National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences[2]
Beijing, 100190, China
(Email: licp@dsp.ac.cn)
(Received July 29, 2019; Revised and Accepted Dec. 6, 2019; First Online Feb. 9, 2020)

Abstract

Parametric t-distributed stochastic neighbor embedding (t-SNE) algorithm is a kind of unsupervised dimensionality reduction method which is widely used and effectively. However, current research rarely involves the application of parametric t-SNE in network attack detection. Simultaneously, it is rare to apply a reasonable model for parametric t-SNE. Therefore, we propose a novel unsupervised dimensionality reduction algorithm to detect attack behaviors, which uses t-SNE combined with a hierarchical neural network. This algorithm maps a high-dimensional network data space into a low-dimensional latent space. Furthermore, we evaluate the performance of the parametric t-SNE method in experiments using two public network intrusion datasets and a self-collected network dataset. In experiments, several unsupervised dimensionality reduction algorithms are discussed and compared with the algorithm we proposed. This comparison shows that parametric t-SNE based on hierarchical neural network gets excellent dimensionality effect, which achieved a maximum of 99% accuracy for 1-nearest neighbor.

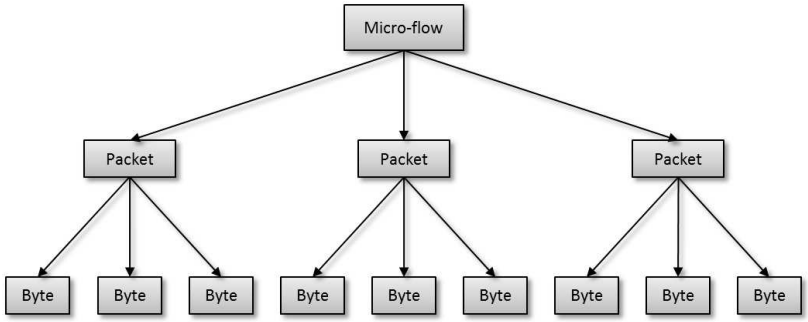
Keywords: Hierarchical Neural Network; Network Intrusion Detection; Parametric T-SNE

1 Introduction

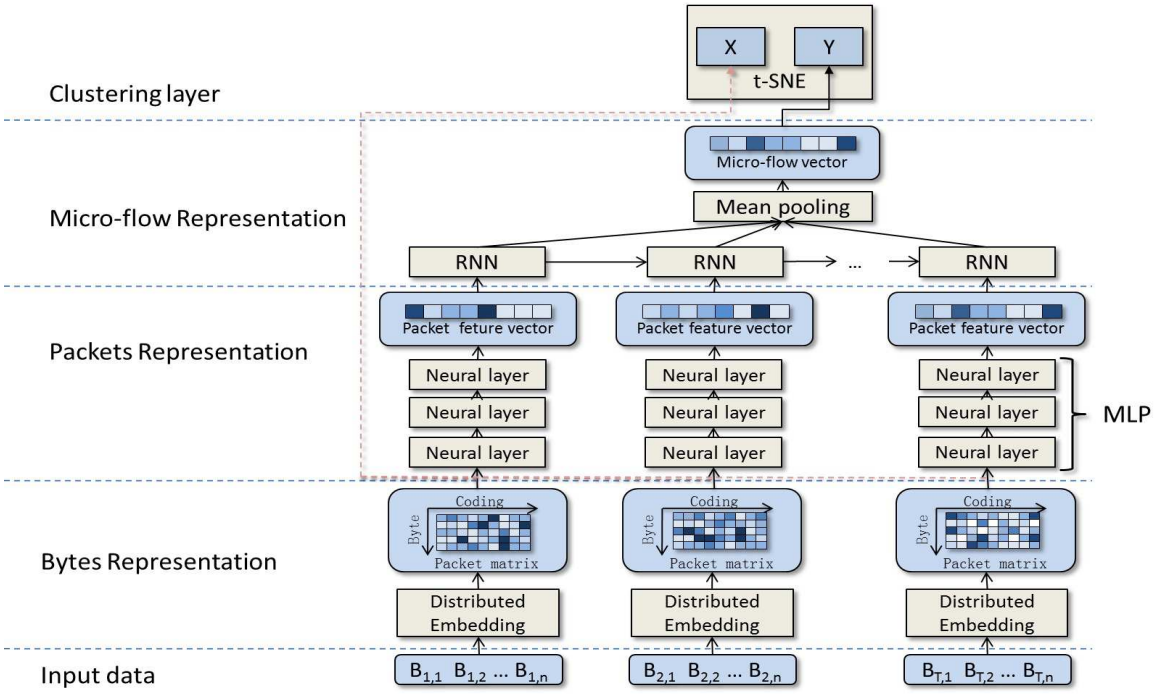
Network data possesses high-dimensional characteristics, which hinders a machine learning model from achieving good performance. Therefore, dimensional reduction is commonly used for a large amount of high-dimensional network data. Traditional reduction algorithms, such as principal component analysis (PCA) and neighborhood components analysis [22], are the commonly used linear reduction techniques. However, these linear reduction algorithms are not ideal when dealing with nonlinear data in a high-dimensional space. In addition, auto-encoders proposed by Hinton [6] can map high-dimensional data by maximizing the variances in latent space. Manifold learning is another such reduction algorithm. Various algorithms, such as Isomap [8], Locally Linear Embedding (LLE) [9], and Maximum Variance Unfolding (MVU) [15], focus more on the local structure of the high-dimensional data. Unfortunately, these algorithms are non-parametric and cannot map the out-of-sample data. A typical t-distributed stochastic neighbor embedding (t-SNE) algorithm [20] is another such non-parametric manifold learning algorithm. Furthermore, Maaten *et al.* [3] presented a parametric t-SNE model based on stacked restricted Boltzmann machine models [12] and solved this problem of out-of-sample data.

However, for network data, it cannot simply build a stacked and fully connected neural network model because of the hierarchical structure of network data. A network streaming data consists of two layers, *i.e.*, packet and micro-flow layers. The micro-flow layer is a set of IP packets that contain the same source IP address, destination IP address, source port, and destination port; and are from the same time window. In this study, micro-flow (defined by five tuple) data is considered to be a sequence of network packets, and these packets can be considered as limited-length data blocks. Thus, a packet layer means byte-level data and the streaming layer is a sequence of packets. To detect network attacks by modeling both packet and streaming layers, a hierarchical model is designed in this study.

This study aims at investigating and proposing a new



(a) The structure of network traffics



(b) The RNN-MLP model

Figure 1: The structure of network traffics and RNN-MLP model

parametric t-SNE model, which is adapted to the hierarchical structure of network data and performs unsupervised dimensionality reduction because of the lack of labels for malicious network behaviors. Unlike typical t-SNE, the new algorithm should solve the problem of out-of-sample data. We performed dimensional reduction and visualization based on the Defense Advanced Research Projects Agency (DARPA) 1998 dataset [4] and the Information Security Centre of Excellence (ISCX)-2012 dataset [11]. The effects of various hyper-parameters, such as perplexity, learning rate, packet length, and flow length, on the results of dimensionality reduction are discussed. The experiments show that the parameterized t-SNE method has about four to five percent absolute improvement as compared with other classical dimensionality reduction methods, such as using auto-encoders and PCA.

The remainder of this study is organized as follows. In the second section, the parametric t-SNE algorithm combined with a hierarchical deep neural network is described. Furthermore, in the third section, the experimental setup and results, and performance of various influencing factors are discussed. Finally, conclusions and future work are described in the fourth section.

2 Parametric T-SNE Based On Hierarchical Deep Neural Network

In this section, we introduce parametric t-SNE based on the recurrent neural network (RNN)-multilayer perceptron (MLP) model [5, 17]. First, a typical t-SNE algorithm is described, which is a global dimensional reduction method. However, the out-of-the-sample extension is invalid. Further, we introduce the parametric t-SNE algorithm. As an improvement, the algorithm can train the RNN-MLP model while performing global dimensionality reduction, thereby making the model effective for external

samples. In addition, we also discuss the preprocessing method of network traffics.

2.1 Structure Of Micro-flow And Hierarchical Neural Network

In this study, the micro-flow is the intrusion detecting object. The micro-flow sequence is divided into two layers, *i.e.*, flow and packet layers. The network data is temporal sequence and hierarchical. In Figure 1(a), the structure of micro-flow is shown. One micro-flow is composed of an ordered set of network packets and one packet is composed of bytes. Therefore, it is necessary to design a hierarchical model corresponding for the special data structure.

Inspired by the special structure of network traffics, we design a hierarchical deep neural network model, named RNN-MLP. The structure of this model is shown in Figure 1(b). The model consists of 4 parts. The first layer from the bottom is the byte representation. The second layer is the packet representation, and the third layer is RNN model, which is the flow representation. The RNN model is a deep neural network, which is suitable for modeling temporal sequences, such as speech recognition [1], language models [10] and micro-flow [14]. The top layer is a t-SNE clustering model.

1) Byte representation. In this paper, we adopt distributed embedding for byte representation. The network packets are composed of bytes,which are presented as $packet = \{b_1, b_2, \ldots, b_n\}$, where n is the number of bytes in a packet. As the input of an embedding function f_{emb}, each byte is mapped to a k-dimensional byte-embedding vector, and each element of the vector follows a uniform distribution from 0 to 1. The mapping packet is contracted presented as:

$$v_p = \{f_{emb}(b_1), f_{emb}(b_2), \ldots, f_{emb}(b_n)\}$$

where, v_p is the packet vector which is concatenated by byte vectors and is taken as the input for the following RBM model.

2) Packet representation. The packet representation refers to the whole MLP because raw data are recommended as inputs in the deep neural network commonly. The packets of byte-level data are directly considered as model inputs. The output of MLP is presented as follows:

$$o_{mlp} = \theta(W_{m\_o} \cdot \theta(W_{m\_h} \cdot x + b_{m\_h}) + b_{m\_o}),$$

where x refers to input data that equals to v_p; W_{m_o} and W_{m_h} are the weights of the output and hidden layers, respectively; b_{m_h} and b_{m_o} are the biases of the hidden and output layers, respectively; the function $\theta(\cdot)$ is the activation function; o_{mlp} is the packet feature vector.

3) Flow representation. The flow representation refers to the entire recurrent neural network [19] There are two aspects of the inputs to a recurrent model. One part is the output of MLP, o_{mlp}, and the other is the output of the recurrent layer from the last time step. The recurrent network is presented as follows:

$$o_{rnn,t} = \theta(W_{r\_i} \cdot o_{mlp} + W_{r\_h} \cdot o_{r,t-1} + b),$$

where W_{r_i} and W_{r_h} are the weights of the input layer and the recurrent layer, and the symbol b is the bias; $o_{r,t}$ is the output of the recurrent layer at the tth step.

4) Clustering layer. After obtaining the output $o_{rnn,t}$ from the RNN model, a parametric t-SNE method is adopted to cluster whose detail is discussed in Section 2.2.

2.2 Parametric T-SNE Model and Backward Propagation

In this part, we firstly introduce the t-SHE algorithm and obtain a gradient of cost function. Then the parametric t-SNE algorithm based on hierarchical neural network is discussed. The t-SNE algorithm consists of two steps. The first step is probability distribution in a high-dimensional space is performed. Accordingly, the more similar a pair of objects in the space are, the easier it is to be selected. Conversely, the probability of selecting two dissimilar objects is reduced. Further, the probability in a low-dimensional space is constructed, and the high-dimensional probability distribution is similar to the low-dimensional probability distribution. Different algorithms use different criteria to measure similarity distances (such as k-means using Euclidean distance). The t-SNE algorithm uses conditional probability to present the similarity distances of two objects. Specifically, when given a set $X = \{x1, x2, \ldots, xN\}$ containing N samples (objects), between any two samples xi and xj, the distance is defined as follows:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}$$

The definition of the probability condition between two samples is as follows:

$$p_{j|i} = \frac{\exp(-\frac{||x_i - x_j||^2}{2\sigma_i^2})}{\sum_{k \neq i} \exp(-\frac{||x_i - x_k||^2}{2\sigma_i^2})}$$

where σ_i^2 is the standard deviation of the Gaussian distribution of the data.

After dimensional reduction via t-SNE, the samples' set is presented as$Y = \{y1, y2, \ldots, yN\}$, which is the mapping from a high-dimensional space into a low-dimensional space. The distance q_{ij} between two samples in the low-dimensional space is presented as follows:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}}$$

The final optimization of the t-SNE algorithm is minimizing Kullback-Leibler (KL) divergence, which is presented as follows:

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Generally, the values of pii and qii are 0. The minimization of the KL divergence is non-convex optimization; thus, the technique of mini-batch gradient descent is adopted, and the gradient is presented as follows:

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \quad (1)$$

So far the partial derivative about y_i are obtained and the method of t-SNE algorithm has been introduced.

In the model of parametric t-SNE, the symbol y_i refers to the output of the hierarchical neural network. The weights of neural networks are updated by back propagation. In this case, the weights of the model are presented as $W = \{w1, w2, \ldots, wK\}$, where K refers to the amount of weights from the neural network, and the gradient is presented as follows:

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial Y} \frac{\partial Y}{\partial w_j}, j = 1, 2, \ldots, K,$$

where $\frac{\partial C}{\partial Y}$ can be calculated by Equation (1) and $\frac{\partial Y}{\partial w_j}$ can be calculated by back propagation.

2.3 Preprocessing

The micro-flow sequence is preprocessed. A micro-flow sequence, f, contains many ordered network packets, which can be presented as $f = \{p1, p2, \ldots, pm\}$. The length of a micro-flow sequence refers to the number of the packets it contains. Furthermore, the length of packets is also different between any two packets. Thus, the dimensions of the samples are inconsistent and not suitable as the input for the t-SNE algorithm. The preprocessing is described as follows:

1) Cutting and padding. To construct an equal-length micro-flow sequence, cutting and padding are involved. Preset each micro-flow sequence to contain m packets and each packet contains t bytes. Under known m and t conditions, the truncated network packets or micro-flow sequence can be cut. For a network packet or micro-flow that is too short, it needs to be padding with zero. The details of cutting and padding are presented in Algorithm 1.

2) Ignoring the address information. In network intrusion detection, the IP and Mac addresses are usually shielded to avoid interference from these messages in the detecting model.

3 Parametric T-SNE Based On Hierarchical Deep Neural Network

In this section, the experimental setups, evaluation metrics, effects of hyper parameters, and performance comparison of different algorithms are discussed.

3.1 Experimental Setup

In the experimental setup, two public datasets, DARPA 1998 [4] and ISCX-2012 [11], are involved. Additionally, a self-collected real network dataset without label is also employed in the experiment.

DARPA 1998 is a public dataset, which was sponsored by DARPA for the first realistic and systematic evaluation of research intrusion detection system, published by the MIT Lincoln Laboratory in the United State in 1998. This dataset contains a seven-week training set and a two-week test set. In this dataset, the traffic data contains four types of attacks, *i.e.*, DoS, Probe, U2R, and R2L. The percentage of attacks in the training set of DARPA 1998 is about 65.54%, while the proportion of attacks in DARPA is 63.29%, 1.99%, 0.26%, and 0.01%. The proportion of the test set is similar to that of the training set.

ISCX-2012 is a public network dataset published by the Information Security Centre of Excellence (ISCX) of the University of New Brunswick in Canada in 2012. This dataset contains the full network traffic data of seven days. All traffic data are normal on the first day, while four types of malicious traffics occurred in the following six days. The different typesof malicious traffic were BF-SSH, infiltrating, DDoS, and HttpDoS. The percentage of normality in ISCX-2012 is about 97.27%, while the proportion of attacks in ISCX-2012 is 0.46%, 0.66%, 0.23%, and 1.38% respectively.

The self-collected dataset contains data of seven days full traffic, which is collected by our self-developed network data acquisition equipment from a Chinese telecommunication operator. This dataset is without labels and plays a validated role in the experiments.

The experimental platform is Dell R720, which consists of a CPU of 16 cores with 2.7 GHz, 96 GB memory, and Nvidia Grid K2 GPU. The OS used is Ubuntu 14.04.

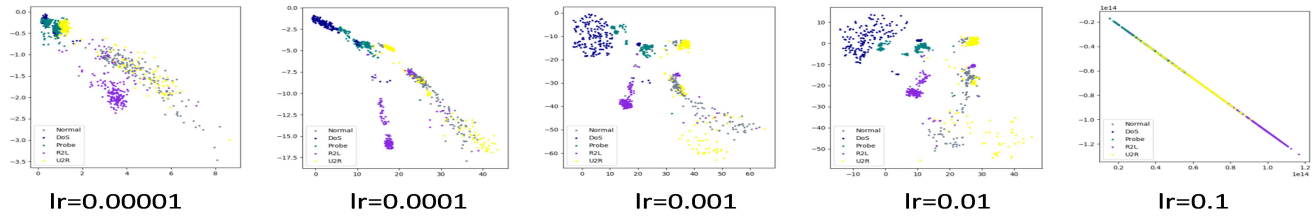
3.2 Evaluation Metrics

In this study, the 1-nearest neighbor (1-NN) algorithm is adopted, and the metrics for this are accuracy and recall. Accuracy is a description of systematic errors, a measure of statistical bias that represents the reliability of a rule, usually represented by the proportion of correct classifications. However, if some attacks are more important, the recall, which is the fraction of relevant instances that have been retrieved over the total amount of relevant instances,

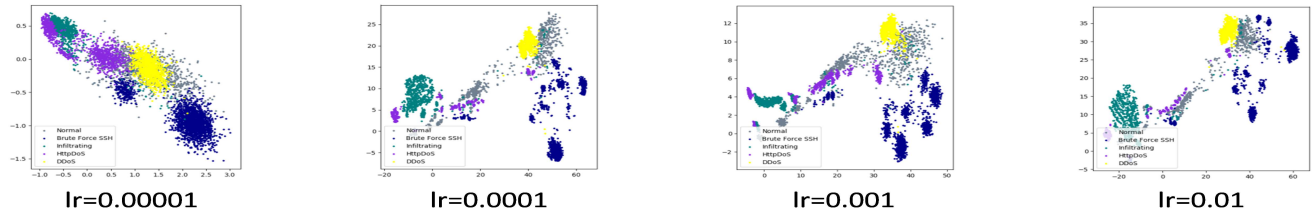
Algorithm 1 Preprocessing the network flow

Input: micro-flow, presetting m and t
Output: preprocessed micro-flow

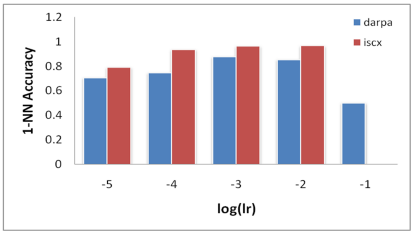
```
cnt ← 0
if length_flow ¡ t then
    padding with (t-length_flow) packets, where the packets are filled by zero values
end if
for each packet in flow: do
    if cnt ≥ t then
        break
    end if
    if length_packet ¡ m: then
        padding with zero value at the end of the packet
    else
        if length_packet ¿ m: then
            cutting and only reserve the first m bytes
        end if
    end if
    cnt ← cnt + 1
end for
```



(a) DARPA: learning rate from 0.00001 to 0.1



(b) ISCX-2012: learning rate from 0.00001 to 0.1



(c) The accuracy with learning rate

Figure 2: Two-dimensional (2D) dimensionality reduction effect under different learning rates

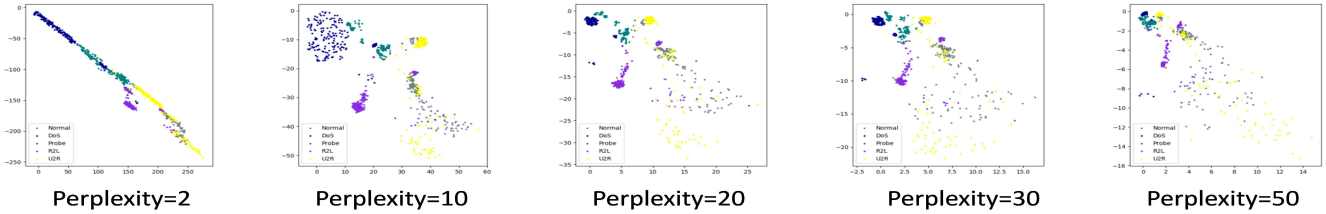
should be given more attention.

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

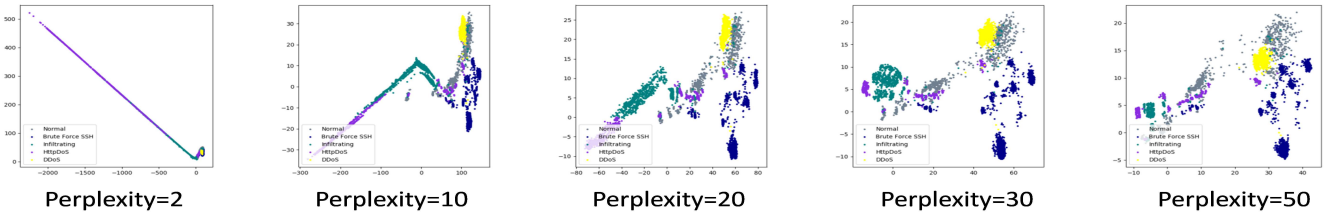
$$Recall = \frac{TP}{TP+FN}$$

where TN is the number of instances correctly predicted as a non-attack instance. FN is the number of instances wrongly predicted as a non-attack instance. FP is the number of instances wrongly predicted as an attack. TP is the number of instances correctly predicted as an attack.

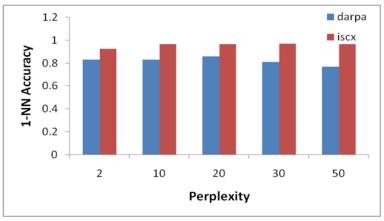
In this study, the 2D dimensionality reduction renderings are also involved. The visual renderings are not numerical indicators; however, they enable us to visually determine the effect of dimensionality reduction.



(a) DARPA: perplexity from 2 to 50



(b) ISCX-2012: perplexity from 2 to 50



(c) The accuracy with perplexity

Figure 3: 2D dimensionality reduction effect under different perplexity

3.3 Influence Of Hyper Parameters

There are two main types of hyper parameters that affect the performance of the parametric t- SNE algorithm. One consists of the inherent parameters of the algorithm, such as learning rate and perplexity. The other type consists of parameters of the preprocessing network data (the length of micro-flow and the size of packets).

1) Learning rate. The learning rate of the parametric t-SNE algorithm affects the speed at which the model converges in back propagation. There is an argument [20]that if the learning rate is too low, the distribution of samples tends to be spherical; Conversely, the model cannot converge.

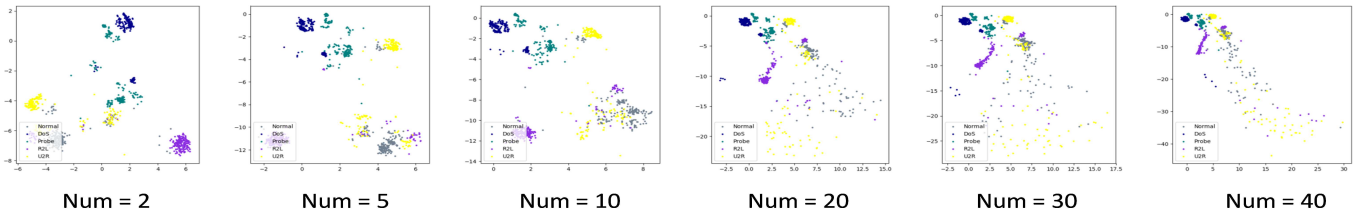
   In the experiments of learning rate, the range is from 0.0001 to 0.1. In Figure 2, the 2D dimensionality reduction renderings are shown. In the DARPA data set, the 1-NN error rate is the lowest between 0.001 and 0.01 learning rates, and too large or too small learning rates will increase the error rate. A similar phenomenon exists in the ISCX-2012 data set. However, the difference is that when the learning rate reaches 0.1, this model cannot converge, and the learning rate cannot be calculated.

2) Perplexity. The degree of perplexity is the number of nearest neighbors selected during the iterative update process. Generally, a larger sample set requires a higher degree of perplexity. It is highlighted in the literature [20]that the non-parametric t-SNE algorithm is less sensitive to the confusion parameter. However, it has been found through experiments that the parametric t-SNE algorithm is more sensitive to perplexity than the non-parametric t-SNE algorithm.

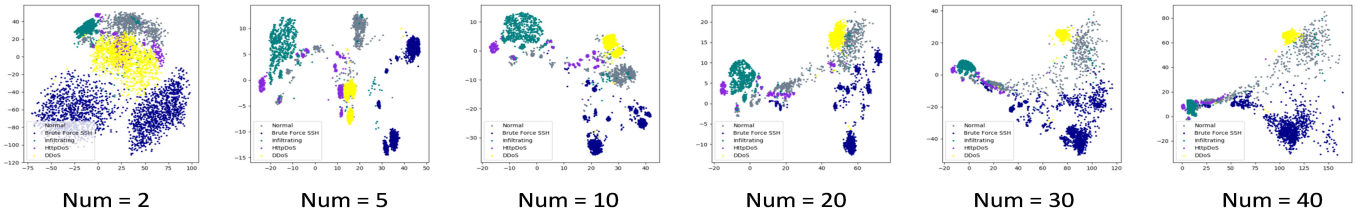
   In the experiments of perplexity, the range is from 2 to 50. It can be seen that the degree of perplexity is data sensitive. For the DARPA 1998 dataset, the degree of perplexity has a greater impact on the 1-NN accuracy; however, for the ISCX-2012 dataset, the degree of perplexity is less affected. The preliminary assumption is that in different datasets the manifold characteristics are not identical in high-dimensional data space. The 1-NN accuracy rate does not change too much; however, the effect of data's 2D reduction is significant. It can be clearly seen that when the perplexity is 2 or 10, the high-dimensional data is not effectively mapped into the latent space.

3) Length of micro-flow. The length of micro-flow refers to the number of packets that one flow contains. We use a five tuple (source IP, destination IP, source port, destination port, and time window) sequence of packets that indicate the micro-flow. The number of network packets included in each flow is not uniform; thus, cutting and padding is required.

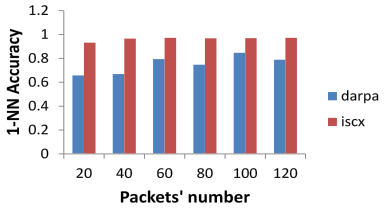
   As shown in Figure 4, in the experiments of the length of micro-flow, the range is from 2 to 50. There is a phenomenon that the 1-NN error rate with short-



(a) DARPA: flow number from 2 to 40



(b) ISCX-2012: flow number from 2 to 40



(c) The accuracy with flow number

Figure 4: 2D dimensionality reduction effect under different length of micro-flow

length flow is lower than that of the longer one. A simple explanation is that irrespective of a normal or attack traffic, the first few network packets in a micro-flow are all connected packets, which cannot be detected as attacks.

4) length of packets. The size of a packet is the number of bytes a sampled IP packet contains. The short IP packets are padding. Conversely, long packets are cutting. Considering the importance of the header data of the IP packet, at least 60 bytes are reserved (The IP packet contains at least a 20 byte header. The TCP layer also contains at least a 20 byte header. Other application layer protocol data reserved 20 byte header).

The experiment of the length of packets is illustrated in Figure 5. The range of length is from 20 to 120 bytes. The best performance is achieved at the 100 byte length in both DARPA 1998 and ISCX-2012 datasets. In most cases, packets need to be 100 bytes long to contain enough information to be detected; however, packets with more than 100 bytes may cause excessive padding, and it involves too much noise.

3.4 Results Of Comparison Experiments

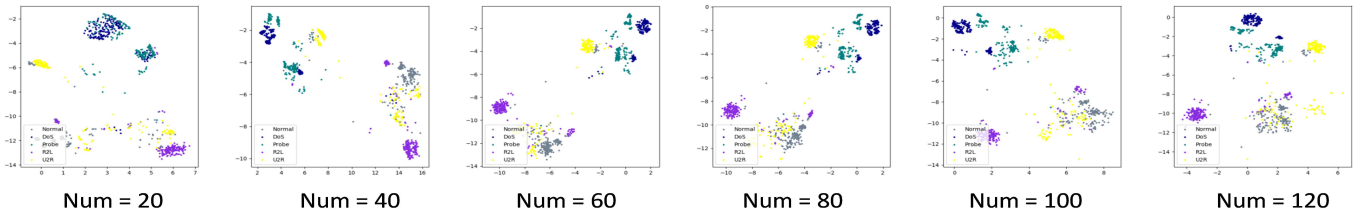
We set up a control experiment choosing PCA and auto-encoder as the control group.

The PCA algorithm is a typical linear dimensional reduction algorithm. The main idea of PCA is mapping data along the maximum direction of the variance makes the data easier to distinguish.

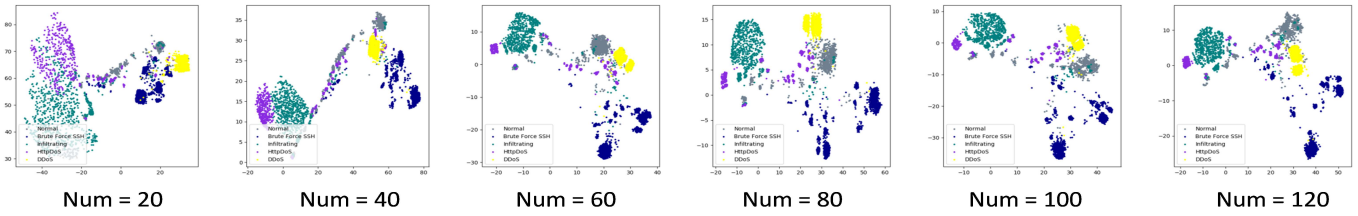
The auto-encoder was proposed by Hinton in 2006 [6], which is a deep neural network unsupervised algorithm. The core idea is that using multi-layer neural network makes input vectors closed to the output vectors. The structure of the auto-encoder is shown in Figure 6. The structure of the auto-encoder contains encoder and decoder, which are multi-layer neural network. The dimensional reduction is the output of the encoder.

The 2D dimensionality reduction renderings of DARAP1998, ISCX-2012 and the self-collected dataset are shown in Figure 6. In the subfigures of PCA, the sample points are scattered and have a certain clustering effect; however, the distinction between U2R, normal, and DoS is not obvious. In the subfigures of the auto-encoder, the sample points are scattered, and only R2L can be distinguished from other types. In the subfigures of parametric t-SNE, the effects of reduction are obvious.

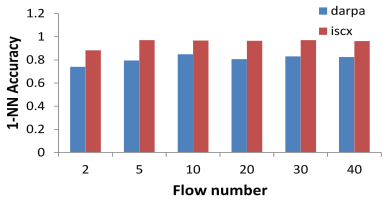
In Figure 6(c), the performance of each reduction algorithms are shown based on the self-collected dataset. Differently from the other datasets, the data is untagged here, so the picture is with only one color to mark the network flow. It shows that by PCA and auto-encoder algorithms, samples have not been distinguished or reduce the dimension sensibly. However, sample points are effectively divided into 5 clusters via t-SNE algorithm.



(a) DARPA: packets' number from 20 to 120

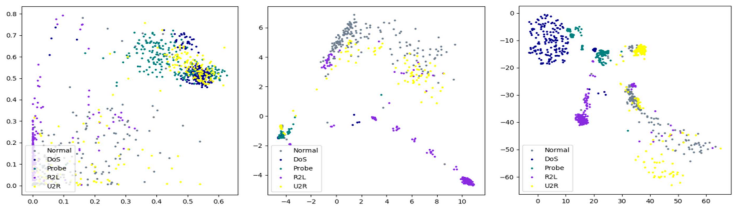


(b) ISCX-2012: packets' number from 20 to 120

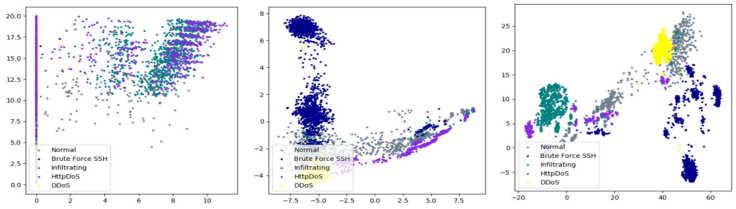


(c) The accuracy with packets'number

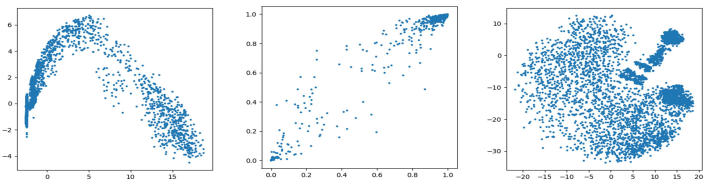
Figure 5: 2D dimensionality reduction effect under different length of packets



(a) DARPA: 2-D dimensionality reduction



(b) ISCX-2012 : 2-D dimensionality reduction



(c) self-collected dataset: 2-D dimensionality reduction

Figure 6: 2D dimensionality reduction effect of PCA (left), auto-encoder (middle) and parametric t-SNE algorithm (right)

In Table 1, the accuracies and recalls of 2-D, 5-D, and 10-D dimensional reductions are presented. According

Table 1: 1-NN accuracy and recall rate for PCA, auto-encoder, and parametric t-SNE

Dimensions	2-D				5-D				10-D			
Datasets	DARPA		ISCX		DARPA		ISCX		DARPA		ISCX	
Metrics	Acc	Rc	Acc	Rc	Acc	Rc	Acc	Rc	Acc	Rc	Acc	Rc
PCA	0.784	0.83	0.831	0.854	0.823	0.819	0.988	0.986	0.832	0.831	0.985	0.982
Auto-encoder	0.411	0.654	0.361	0.517	0.684	0.774	0.678	0.698	0.826	0.856	0.875	0.863
t-SNE (RNN)	0.85	0.871	0.97	0.967	0.773	0.819	0.986	0.983	0.842	0.874	0.99	0.988
t-SNE (MLP)	0.819	0.862	0.975	0.972	0.808	0.858	0.962	0.957	0.819	0.869	0.949	0.945
t-SNE (RNN-MLP)	0.848	0.872	0.981	0.978	0.791	0.858	0.989	0.987	0.871	0.897	0.99	0.988

Table 2: The performance of different algorithms based on DARPA1998 dataset and ISCX-2012 dataset

Dataset	DARPA1998		ISCX-2012	
Algorithm	Accuracy	Avg-Rc	Accuracy	Avg-Rc
SVM [21]	79.4	47.6	N/A	N/A
Random forest [7]	91.4	78.23	N/A	N/A
Bayes network [13]	90.6	53.47	N/A	N/A
PLSSVM [2]	99.8	68.25	N/A	N/A
ALL-AGL [16]	N/A	N/A	95.4	93.2
AMGA2-NB [18]	N/A	N/A	94.5	92.7
t-SNE(RNN-MLP)	87.1	89.7	99.0	98.8

to the table, the parametric t-SNE method based on the RNN-MLP model is better than other algorithms. By comparing different dimensions, we can observe that as the dimension increases, both the 1-NN accuracy and 1-NN recall rate of the algorithm increase.

3.5 Algorrithm And Implementation Comparison

In this paper we also compare some algorithms based on the KDD99 dataset and ISCX-2012 dataset. It is noteworthy that most of the current studies are based on supervised method, and the t-SNE method we proposed are unsupervised. The methods of KNN, SVM, Tree and random forests, and Bayes are involved in Table 2.

As can be seen, for the DARPA1998 dataset, the t-SNE model performed well in terms of the accuracy rate and obtained a higher average recall than any other algorithms in Table 2. Our model were constructed via a recurrent neural network and t-SNE which takes the byte-level data (raw data) as inputs. It could be inferred that the recurrent model is suitable for streaming-type data.

According to the performance of difference algorithms based on ISCX-2012 dataset, we compared the t-SNE method with three supervised methods. In spite of the t-SNE algorithm is unsupervised, t-SNE method got a best accuracy rate and a second good recall rate.

4 Conclusions

- In this study, we are committed to developing an unsupervised dimensionality reduction method for facing network attacks and have proposed a parametric t-SNE method based on a hierarchical neural network. Furthermore, a data preprocessing method adapted to the parametric t-SNE algorithm for the indefinite-length network data is discussed.

- In the experiments, the proposed method achieves better results of dimensional reduction than other algorithms.

- In future works, we aim to investigate how to introduce geographic information, such as IP addresses, as input data. Furthermore, we aim to investigate other unsupervised reduction or clustering methods for network attacks.

References

[1] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *National Aerospace and Electronics Conference (NAECON'15)*, pp. 339–344, 2015.

[2] F. Amiri, M. R. Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1184–1199, 2011.

[3] L. V. Der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[4] A. Fischer and C. Igel, "An introduction to restricted boltzmann machines," in *Iberoamerican Congress on Pattern Recognition*, pp. 14–36, 2012.

[5] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in Neural Information Processing Systems*, pp. 1019–1027, 2016.

[6] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems*, pp. 513–520, 2005.

[7] M. A. M. Hasan, M. Nasser, B. Pal, and S. Ahmad, "Support vector machine and random forest modeling for intrusion detection system (IDS)," *Journal of Intelligent Learning Systems and Applications*, vol. 6, no. 1, pp. 45, 2014.

[8] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[9] T. Iwata, K. Saito, N. Ueda, S. Stromsten, T. L. Griffiths, and J. B. Tenenbaum, "Parametric embedding for class visualization," in *Advances in Neural Information Processing Systems*, pp. 617–624, 2005.

[10] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *Computation and Language*, 2016. arXiv Preprint arXiv:1602.02410

[11] R. Lippmann, R. K. Cunningham, D. J. Fried, I. Graf, K. R. Kendall, S. E. Webster, and M. A. Zissman, "Results of the darpa 1998 offline intrusion detection evaluation," in *Recent Advances in Intrusion Detection*, vol. 99, pp. 829–835, 1999.

[12] L. V. D. Maaten, "Learning a parametric embedding by preserving local structure," in *Artificial Intelligence and Statistics*, pp. 384–391, 2009.

[13] H. A. Nguyen and D. Choi, "Application of data mining to network intrusion detection: Classifier selection model," in *Asia-Pacific Network Operations and Management Symposium*, pp. 399–408, 2008.

[14] T. L. Pao, W. Y. Liao, Y. T. Chen, and T. N. Wu, "Mandarin audio-visual speech recognition with effects to the noise and emotion," *International Journal of Innovative Computing, Information and Control (IJICIC'10)*, vol. 6, no. 2, pp. 711–724, 2010.

[15] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[16] H. Sallay, A. Ammar, M. B. Saad, and S. Bourouis, "A real time adaptive intrusion detection alert classifier for high speed networks," in *IEEE 12th International Symposium on Network Computing and Applications*, pp. 73–80, 2013.

[17] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, 2012.

[18] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of denial-of-service attacks based on computer vision techniques," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2519–2533, 2014.

[19] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2017.

[20] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proceedings of the Twenty-first International Conference on Machine Learning*, pp. 106, 2004.

[21] X. Xu, "Adaptive intrusion detection based on machine learning: Feature extraction, classifier construction and sequential pattern prediction," *International Journal of Web Services Practices*, vol. 2, no. 1-2, pp. 49–58, 2006.

[22] E. Zyad, C. Khalid, and B. Mohammed, "Improving network intrusion detection using geometric mean LDA," *International Journal of Network Security*, vol. 20, no. 5, pp. 820–826, 2018.

Biography

Huijun Yao received his master's degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2007. He began to work in Jiangsu broadcasting cable information network, in 2007. He is now an experienced researcher of technology Institute of Jiangsu broadcasting cable information network, Nanjing, China. His research interests are in the fields of cable television, digital Set-top Box, 5th-Generation.

Peng Sun received his master's degree from Northwestern Polytechnical University, Xi'an, China, in 2011 and his doctoral degree from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, in 2014. He began to work in Institute of Acoustics, in 2014. He is currently a researcher of National Network New Media Engineering Research Center, Chinese Academy of Sciences, Beijing, China. His research interests are in network media, digital signal processing, smart terminal..

Chaopeng Li received his PhD degree in National Network New Media Engineering Research Center, Chinese Academy of Sciences, Beijing, China, in 2019 and received his Bachelor's degree from Beijing University of Post and Telecommunications, Beijing, China, in 2013. He currently works at Jimei University. His current research interests are in deep learning, network intrusion detection and information processing.

A Method of Constructing Arc Edge Anonymous Area Based on LBS Privacy Protection in the Internet of Vehicles

Peng-Shou Xie, Xue-Ming Han, Tao Feng, Yan Yan, and Guo-Qiang Ma
(Corresponding author: Xue-ming Han)

School of Computer and Communications, Lanzhou University of Technology
No.287 Lan-gong-ping Road, Lanzhou, Gansu 730050, China
(Email: hxmhan@163.com)

(Received Oct. 9, 2019; Revised and Accepted Jan. 6, 2020; First Online Feb. 9, 2020)

Abstract

For the users of Internet of Vehicle, a larger value of the privacy protection factor K means the privacy can be better protected. However, too excessive value of safety factor will cause the decrease of query quality and the accuracy of location information in the Internet of Vehicles. In order to balance the contradiction between privacy protection security and query service quality caused by the accuracy of location information, an arc-edge anonymous area constructing method is proposed based on the location k-anonymity principle, which is used to optimize and improve the boundary-based polygonal anonymous region. Experiment results show that the generalization area can effectively reduce the anonymous region and the relative anonymity, which improves the quality of service on the basis of satisfying the privacy of the Internet of Vehicles.

Keywords: Anonymity; Anonymous Regions Constructed; Arc Edge Anonymous Area; Internet of Vehicles; Location Privacy

1 Introduction

With the continuous development of the Internet of Things, and the increasing popularity of various networks, such as Wifi and the rapid spread of 4G cellular networks, people are increasingly relying on location-based services. For example, check out nearby supermarkets and restaurants, how to find the nearest subway station, *etc.* The Internet of Vehicles comes from the Internet of Things, which can be used in multiple areas [3]. The Internet of Vehicles is also called the vehicular ad hoc networks (VANETs) [5], which is designed to provide vehicle-to-vehicle (V2V) communication and vehicle to infrastructure (V2I). The system of Internet of Vehicles mainly includes onboard unit (OBU), application unit (AU) and road side unit [4] (RSU). OBU is mainly used to exchange information with OBUs in other vehicles or with RSUs; AUs are mainly devices that use OBU's communication capabilities to implement communication functions; RSUs are wireless access devices along both sides of the road or dedicated fixed locations. In the Internet of Vehicles, data transmission and sharing operations are realized through wireless access technology. Location-Based Service (LBS) is a location-based server that provides value-added services to users based on their own location information provided by mobile users.

GPS and location-based services bring great convenience to people's travel and can be used in many industries. But at the same time, personal privacy leaks have become a serious problem [7, 8]. Personal occupation, hobbies, and health conditions are easily leaked, and personal identity information may be fraudulently used. It's no exception for the Internet of Vehicles. A series of information such as the location and trajectory of the vehicle are easily leaked, and these leaks provide convenience to the attacker [1]. So how to protect LBS-based location services is a hot issue studied by many scholars today.

Dummy position and pseudonym [18], cryptography [19] and fuzzy generalization [2] are common methods for location privacy protection. The earliest example of applying K-anonymity to location privacy protection is to generalize an area containing K users into a rectangular plane on the plane.The user send a request to the LBS anonymous server by the area, so that the probability of the user being attacked is 1/K. Rectangle area generalization is considered as an effective k-anonymity privacy protection model [10], which has certain security performance. And many subsequent studies are basically based on this idea. Later scholars adopted a quad-tree structure and indexed blocks to calculate anonymity [12]. Later the P2PSC algorithm was proposed [11], which forms an anonymous region through P2P and multi-hop communication. Lin Ying used Hilbert method to further optimize

the anonymous region, and some invalid grids were reduced [9]. But this approach puts itself in the center of the anonymous zone which is easy to be attacked. In addition to rectangular anonymous areas, circularly divided anonymous areas are also used for privacy protection [14]. Jia Zongqi proposes a privacy protection method based on fan-shaped anonymous region aiming at the problem of weak anti-central attack ability [6], Experiments show that anonymous performance and energy consumption can achieve good results under different user densities. Later scholars proposed the SCABGE algorithm [13], they divide the plane into multiple grid planes, and continually doubles to find the anonymous areas that meet the requirements, and caches the results. But this method will also leak the user's location. In addition to the regular graphics being used for anonymous areas, boundary-based polygons are also used in anonymous areas [15].

In the Vento-based location privacy nearest neighbor query method [20], In order to solve the problem that spatial anonymous regions are vulnerable to multiple queries and inference attacks, Zhou Yihua constructed a random k-hidden set to satisfy the location k-anonymity and l-diversity. The grid distance between the random k-hidden sets is greater than the threshold S. Using private information retrieval protocol to ensure the privacy of query results in the retrieval process, and the service provider provides the location-based services to the users without knowing the accurate query results of the users. Pei Zhuoxiong considered the query service area of location service providers [17], Introduce it into anonymous region construction. Generate subdomain areas and merge them according to the size of service providers finally the quality of the query service is improved. Aiming at the problems of high communication overhead, low anonymity efficiency and low success rate in the formation of location privacy protection anonymous area under mobile point-to-point (P2P) architecture, Xu Mingyan proposed a distributed user awareness scheme [16], it Recommends privacy parameters and search radius for candidate users according to user distribution characteristics and help users quickly form anonymous areas. The simulation results show that the algorithm has low communication overhead and high success rate. However, some of the above methods have a large anonymity area, some do not consider the center attack, some reduce the quality of the query, Some increase the amount of calculation and thus lower the quality of service.

From these aspects, We improve the arc-edge anonymity area based on location-based service privacy protection in Internet of Vehicles. The system structure is based on a central server structure. The algorithm is based on k-anonymity, starting from the initiator, forming a gradually expanding polygon, Searching the networking object from the counterclockwise direction. As the polygon expands, until the K target objects form an anonymous polygon area. And finally the arc polygon is constructed, the initiator of this algorithm is a virtual object, the real object is located at a random position of the arc-edge polygon, Central attack is effectively avoided. At the same time, since the anonymous area is an arc-side polygon, the anonymous area is reduced and the quality of service is improved.

2 Process of Constructing Arc Edge Polygon

2.1 Architecture

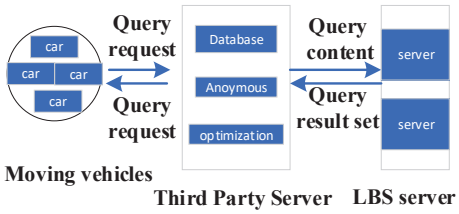


Figure 1: Architecture of location-based privacy protection anonymous system for Internet of Vehicle

Limited by bandwidth between vehicles and LBS server, The anonymous system structure of the privacy protection of the Internet of Vehicle consists of three parts to meet the real-time anonymity process. As shown Figure 1. The first part is vehicle network, the second part is the central anonymous server, It's used for anonymous related processes. The third part is the LBS server, which is used to process data and return or receive requests for location services. Data is transmitted between the vehicle and a third-party anonymous server using a secure channel, such as SSL. And the third-party anonymous server hides vehicle IP, identity and geographic information. Each vehicle is equivalent to a node, which has the ability of communication and data processing, and it can receive GPS signals with positioning function. Vehicle Nodes can communicate with other intelligent terminal nodes by single or multiple hops. Nodes self-organize through 3G/4G/Wi-Fi network and communicate with base stations and roadside units. Finally, the location request is sent to the third party central server. The central server carries out the process of vehicle anonymity and generalizes the network node anonymity to form a K-anonymous node set. Then the third party server sends the anonymous set to the LBS location server. The LBS location server performs location query after authentication. Finally, the query results are returned to the third-party server, and the third-party server returns the final results to the proxy point vehicle. The proxy point broadcasts the results to the vehicle network node in the anonymous area. After the request node is broadcasted by the query results, it calculates the location, and finally obtains the required information. This is the service process of the Anonymous Structure Diagram of Internet of Vehicle.

The query process is showed in Figure 2.

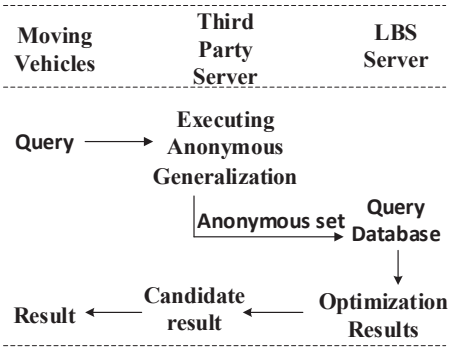


Figure 2: Query process

The paper is based on certain assumptions:

1) Moving vehicles and Third Parity server are trusted, and LBS is semi-trusted, providing services to users. And users' privacy may be leaked.

2) Attackers can obtain some prior knowledge through public databases. At the same time, they have the ability to analyze and reason, and can obtain delayed knowledge through anonymous information and prior knowledge.

3) The distribution of vehicles has certain rules, but has different distribution density.

2.2 Related Instructions

1) K-Anonymity: The k-anonymity mechanism requires that each record in the table be at least consistent with the quasi-identifier of the k-1 records in the table. In short, it means that a user cannot be distinguished from another k-1 users at least. The algorithm is also based on this basic principle, K users are placed in a region of a specific generalized area.

2) The Anonymous demand parameter Q: $\{(x_q,y_q),d_q,\text{k},s_{min},s_{max},t_q\}$, Different node privacy requirements are not necessarily the same. Q represents the set,(x_q, y_q) Represents the coordinates containing the request point, and d_q represents the distance between the request point and the proxy point. K is a privacy requirement. The larger the K, the higher the privacy, indicating that more privacy protection is needed.s_{min} represents the minimum area of anonymity, the smallest anonymous area that required to satisfy privacy requirements.s_{max} represents the maximum anonymous area required for anonymity. Once this area is exceeded, It shows that the request scope is too large and the anonymity condition cannot be met, so the anonymous process will fail. The Internet of Vehicles is a dynamic network, and every moment the vehicle is in a dynamic change. Points added in the anonymous area will exit at the next moment.Therefore, we set a certain delay t_q for it, and it is necessary to reanonymize when beyond this delay. Timeliness is a problem that must be considered. It is worth noting that the anonymous area S is not the bigger the better. Under normal circumstances, the larger S, the larger K, and the probability of being attacked is q=1/k, but the excessively large anonymous area will cause the resource consumption of the server. So set the interval for S. In this way, while satisfying the location privacy, the resource consumption of the server can also be reduced, As a result, the latency of server processing is reduced and the quality of service is improved.

3) Area control: Let the points of the arc-shaped polygons that make up the anonymous area be from the middle to the periphery: $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots, (x_n, y_n)\}$, The distance between the request node and the proxy point d ≤ R. R is the maximum distance of the center of gravity of the anonymous polygon $G(x_0, y_0)$ to the node constituting the polygon of the anonymous region, *i.e.*:

$$R = Max\{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}\} \tag{1}$$

Among them, i=0,1,2,⋯ n, $x_0 = \frac{\sum_{i=1}^{n} x_i}{n}$, $y_0 = \frac{\sum_{i=1}^{n} y_i}{n}$. According to R, the size of the area can be controlled.

4) Process description of privacy protection: The agent point sends a broadcast to the surrounding vehicle node. The broadcast radius is r. The node closest to the agent point receives the broadcast first, and the surrounding nodes receive the message and return the confirmation message to the agent point. And the agent point calculates the anonymous basis according to the privacy requirement. Suppose that the number of nodes around the proxy point is N at this time, but the point required to form an anonymous region is K. Anonymous area S ∈ $[S_{min},S_{max}]$. K is positively related to S. Recorded as K∝ S.

Step 1. At this time, N< K, and the broadcast radius is r=r_0, the counting starts from the counterclockwise direction, and the angle α between the initial counting position and the x-axis of the Cartesian plane rectangular coordinate system is 0^0. As the angle α increases, the point gradually increases. (α ∈ $[0^0,360^0]$).

Step 2. If N=K, the coordinates of the Kth point at this time are $k(x_k,y_k)$, r=r_k.

$$tan\alpha = \frac{y_k}{x_k}. \tag{2}$$

Perform the anonymous generalization process, calculate the area S of the arc-side polygon, and calculate the total delay t_q of the anonymous process. If S ∈ $[S_{min}, S_{max}]$ and t<T, Anonymous process completed. otherwise if: S < S_{min} or if S>S_{max} or

timeout, the anonymity fails, and adaptive adjustment should be made at this time. Increase K or increase S_{max}.

Step 3. If there is still N<K, all the points added after the first broadcast are selected, and then the broadcast area is enlarged, let r=r_1, $\alpha = 0$, and α continues to increase. Calculate K again. If N=K at this time, perform Step 2. Otherwise continue this process until N=K. At this time, if S ϵ[S_{min},S_{max}], Anonymous process completed, otherwise the anonymity fails.

After the anonymity succeeds, The anonymous server sends the query result to the RSU. The RSU broadcasts the result, and the initiator receives the RSU broadcast and calculates the result that is needed. The entire process is completed at this time. As is shown in Algorithm 1.

Algorithm 1 The process of anonymous generalization in the environment of Internet of Vehicles

Begin
α=0, N=0, r=0; t=0 // Variable initialization
if N < K
r=r_N,$\alpha \uparrow$, N++; t++ // Start loop
if N < K
Go back to Step 3
else
Perform an anonymous generalization process and calculate the area S,calculate the total delay t
if Sϵ [S_{min},S_{max}] $\bigvee$ t< t_q
succeed
else
failed
$S_{max} \uparrow$ // Adaptive adjustment if the anonymous //requirement is not met
or K++
Return to Step 3
End

2.3 Anonymous Generalization Area Process Description

The area of the polygonal area is currently superior. Because the polygon is based on the boundary, that is, there are some points on the vertices and edges of the polygon. We optimize and improve the polygonal region division to form an anonymous area of the arc-edge polygon. This anonymous area is better than a polygon only in terms of the area of the anonymous area. Treating the vehicle as a particle, Suppose there are K vehicles at a certain time that need to complete the network formation. That is, K particles are placed in a closed area. The area of this closed area directly affects the processing speed of the server, so we are trying to find a smaller anonymous area.

Step 1. First connect the outermost particles with a straight line to get a random polygon. Random means that the shape is not fixed. It is a polygon of any side. Usually it is a convex polygon. Let the number of sides be i, i=3, 4, 5... k. At this time, a polygon is obtained, such as the polygon formed by the dashed line segments of d, d1, d2, d3, d4, and d5 in Figure 3. The final result of the boundary-based polygon generalization region is similar to this polygon. And it's a convex polygon.

Step 2. Let the inner angles of the polygons be: θ_1,θ_2,θ_3,$\cdots$ θ_n, and select the minimum angle θ_{min} of the polygon. Starting the arc from the side number d,d_2,d_3,$\cdots$ d_n in siquence, two points as the starting and ending points of one side of the arc. From the geometric knowledge:

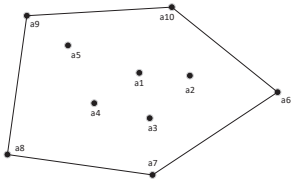


Figure 3: Anonymous region optimization process 1

$$sin(\frac{\theta_{min}}{2}) = \frac{d}{2r} \tag{3}$$

from which the radius of the arc is calculated:

$$r = \frac{d}{2\sin(\frac{\theta_{min}}{2})} \tag{4}$$

and an arc is made in each line segment. In the end a closed region is obtained, where d is the maximum of the two sides forming θ_{min}. As is show in Figure 4.

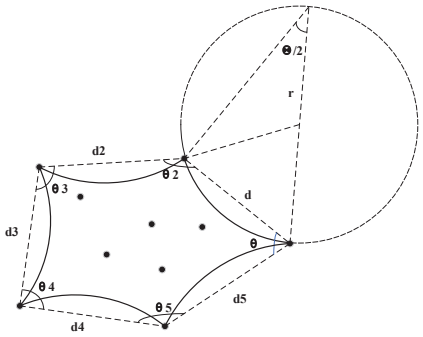


Figure 4: Anonymous region optimization process 2

Step 3. At this point, the area formed by the arc is obtained:

It can be seen from the image that the area of the arc side is smaller than the polygon, so it is superior. Here

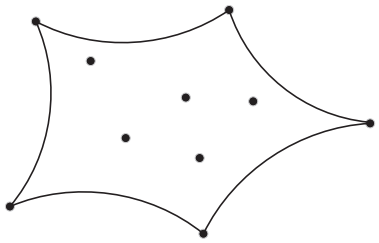


Figure 5: The resulting arc polygon

only the area of the arc and the area of the polygon are calculated. Then we only need to calculate the area Δ_s of the arc and the edge of the polygon. Let the area of the polygon be s_p and the area enclosed by the arc edge be s_a, then get the Equation (5):

$$\begin{aligned}\Delta_s &= \sum_{i=1}^{n}(\frac{\pi r^2\theta_i}{360}-\frac{1}{2}r^2\sin\theta_i) \\ &= \sum_{i=1}^{n}[\frac{\pi r^2\theta_i}{360}-\frac{1}{2}(\frac{d}{2\sin(\frac{\theta_{min}}{2})})^2]. \end{aligned} \quad (5)$$

Simplify Equation (5) and get the Equation (6):

$$\Delta_s = \sum_{i=1}^{n}[\frac{\pi r^2\theta_i}{360}-\frac{d^2\sin\theta_i}{4\sin^2(\frac{\theta_{min}}{2})}] \quad (6)$$

Then Equation (7):

$$\begin{aligned} s_a = s_p - \Delta_s = \frac{1}{2}\{\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix} + ... + \\ \begin{vmatrix} x_n & y_n \\ x_1 & y_1 \end{vmatrix}\} - \sum_{i=1}^{n}[\frac{\pi r^2\theta_i}{360}-\frac{d^2\sin\theta_i}{4\sin^2(\frac{\theta_{min}}{2})}] \end{aligned} \quad (7)$$

So $s_a < s_p$. The time complexity for calculating the arc area is O(kn).

2.4 An Example of the Whole Process of Anonymous Generalization

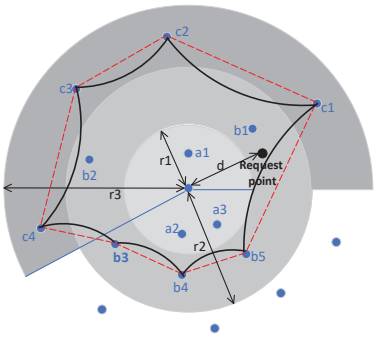


Figure 6: Anonymous generalization process of arc edge polygon

This is an example for whole process of anonymous generalization in the environment of the Internet of Vehicles.As shown in Figure 6, K=16 is taken as an example. When r=r_1, the angle is 0, and then the angle is continuously increased to 360 degrees. The nodes added in sequence are a_1, a_2, a_3, and when r=r_2, the scanning is gradually started from 0 degrees to 360 degrees. Adding the number of nodes b_1, b_2, b_3, b_4, b_5 in turn. Then join the request node, a total of 6 nodes. Similarly, r=r_3, adding c_1, c_2, c_3, c_4 in turn. This constitutes an anonymous area of K=16. After the generalization process, the shape is an arc-edge polygon.

3 Experiment and Analysis

3.1 Experiment Environment

The experiment used the Matalab 2018b environment. On Intel(R) Core i7-7700HQ CPU @2.80 GHZ processor, 16GB RAM. Nvidia GTX 1060 graphic display. Microsoft Windows 10 Professional operating system. The location simulation data of the mobile terminal is generated by the Thomas Brinkhoff road network data generator, using the traffic map of the German city of Oldenburg, to generate 2000 nodes.

Experiments start from three aspects: Hidden area, communication cost and relative anonymity. Compared with rectangular area, circular anonymous area and polygon anonymous area, demonstrated the superiority of an optimized anonymous area.

3.2 Analysis of Results

1) When the number of sides of the arc edge polygon is 6 and each arc edge is equal, we compare it to the perimeter and area of a regular hexagon. Let L be the perimeter and S be the area. It can be seen from Figure 7 that when the perimeters are equal, the arc-sided polygon has a smaller area. So, when the area is equal, the perimeter of the arc edge polygon is longer. In extreme cases, when all vehicles are distributed on the boundary,vehicles located on arc-side polygons are less likely to be attacked.

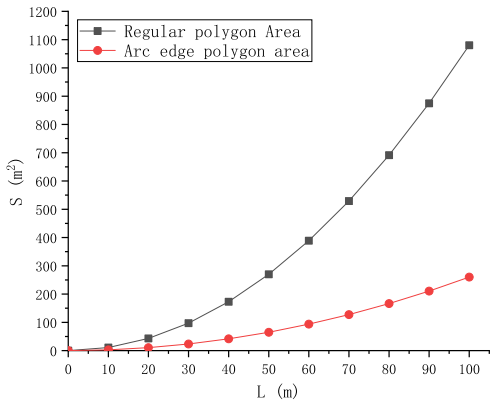


Figure 7: Relationship between perimeter and area

2) Anonymous areas : Comparing the arc-shaped polygon area with the rectangular area, the circular anonymous area, and the polygon anonymous area, it can be seen Figure 8 that the arc-side polygon area has the smallest area. The average anonymous area is an important indicator to measure the strength of privacy protection. Arc-shaped polygons are smaller and more flexible, and reduced invalid anonymous area, so the method in this paper has a smaller anonymous area. Therefore, the quality of service can be improved and the resource consumption of the server reduced.

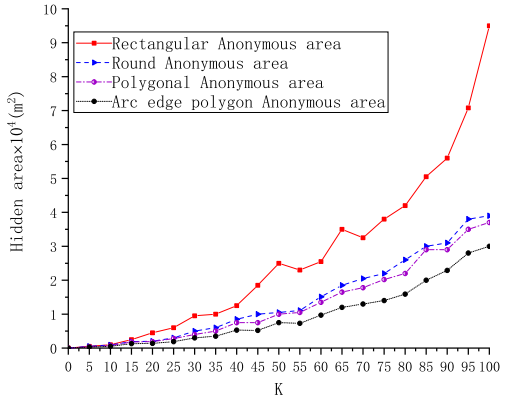


Figure 8: Comparison of different anonymous areas

3) Comprehensive evaluation index : The time consumption of the algorithm can be considered from the following aspects: anonymous waiting time T_w, anonymous processing time T_d, transmission delay T_t and query time T_q. The algorithm consumption time can be expressed as:

$$T = T_w + T_d + T_t + T_q. \tag{8}$$

Anonymous waitting time and transmission delay time are usually ignored. Only considering anonymous processing time and query time. When considering the performance of the algorithm, only the anonymous processing time Td is considered. And the average anonymous time is generally used to measure the performance of the algorithm. As Equation (9).

$$\bar{T_d} = \frac{\Sigma T_d}{\Sigma U_s} \tag{9}$$

Us is a user who is anonymously successful and Td is the time this user spent anonymously.

The average anonymity time includes the total delay that constitutes the anonymous area. The total delay of the algorithm is roughly equal to the delay in forming the anonymous region. Because the methods that the anonymous made up area are more flexible, And there is also no time to adjust nodes. so the delay is reduced. The algorithm complexity is: O(kn), so anonymous time is acceptable.

The arc polygon segmentation process has an arc segmentation process, so the time consumption is slightly more. But the area of the anonymous is effectively reduced. In this method, the smaller the area and the less time consumed, the better the system performance. But it's difficult to reduce both time and area. Therefore, a comprehensive evaluation index EI is used to measure the effect of considering both time and area. As is showed in Equation (10).

$$EI = \bar{T_d} * S. \tag{10}$$

Among them, S is the anonymous area. The smaller the EI, the better the system performance. Figure is a comparison of EI. It can be seen from Figure 9 that the anonymous area with arc edges has better system performance.

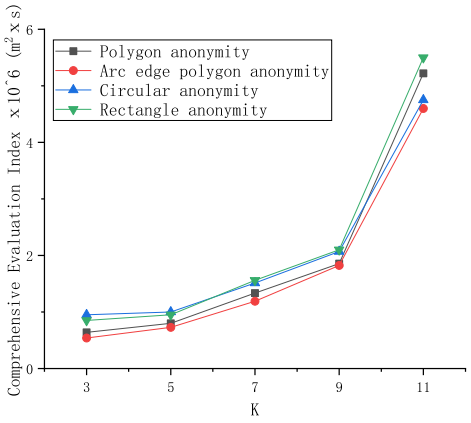


Figure 9: System performance comparison

4) Relative anonymity comparison:

$$K_{rel} = \frac{K_{act}}{K} \tag{11}$$

In Equation (11), Kact represents the number of vehicles that actually complete the anonymity, and K represents the number of vehicles that satisfy the K anonymity requirement. Due to the inherent properties of geometry, it is often difficult to accurately satisfy K anonymity in the actual anonymity process. There will be more than K vehicles in a fixed generalization area, so the number of users participating in anonymity tends to be larger than K. However, the anonymity of polygons and arc polygons is more flexible, the relative anonymity is smaller. And it is easier and more accurate to be controlled.

4 Conclusions

Privacy protection of Location-based service has become a research hotspot. How to provide better location privacy becomes a very important issue in privacy protection.

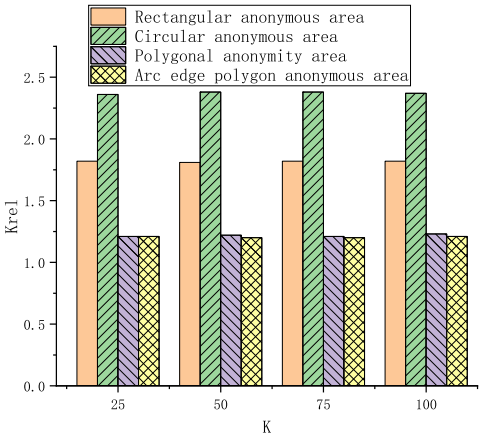


Figure 10: Relative anonymity

The paper compared some common anonymous region construction methods, such as rectangular anonymity, grid doubled, mesh layering, circular area division, sector division, and polygon division. On this basis, an arc-edge anonymous area constructing method is proposed to further reduce the anonymous areas. The proposed method improved service quality based on protecting user privacy in certain extent.

However, the proposed method also has some defects. The partition method is more complicated, and the time complexity is increased. The actual system of Internet of Vehicle is in a very complex environment, whether from the road network or the vehicle itself. From the perspective of road network, the method of regional generalization may not be entirely suitable because there are disturbances from buildings and other objects. So, it is a big challenge to design a regional construction method that fits the real Internet of Vehicle system. It is also the focus of the next step of research. Meanwhile, simplifying the partition method and reducing the time complexity are the further research contents.

Acknowledgments

This study was supported by the National Natural Science Foundations of China under Grants No.61862040 and No. 61762060 and No.61762059, The authors gratefully acknowledge the anonymous reviewers for their helpful comments and suggestions.

References

[1] R. A. Dhubhani, J. M. Cazala, "An adaptive geo-indistinguishability mechanism for continuous LBS queries," *Wireless Networks*, vol. 24, no. 8, pp. 3221–3239, 2018.

[2] P. Galdames, C. Gutierrez-Soto, A. Curiel, "Batching location cloaking techniques for location privacy and safety protection," *Mobile Information Systems*, vol. 2019, pp. 1–11, 2019.

[3] S. Hong, "Authentication techniques in the Internet of Things environment: A survey Sunghyuck Hong," *International Journal of Network Security*, vol. 21, no. 3, pp. 462–470, 2019.

[4] S. Ibrahim, M. Hamdy, E. Shaaban, "Towards an optimum authentication service allocation and availability in VANETs," *International Journal of Network Security*, vol. 19, no. 6, pp. 955–965, 2017.

[5] T. Jeyaprakash, R. Mukesh, "A new trusted routing protocol for vehicular ad hoc networks using trusted metrics," *International Journal of Network Security*, vol. 19, no. 4, pp. 537–545, 2017.

[6] Z. Jia, W. Liu, "Location privacy protection based on sector region in dynamic P2P network," *Computer Applications and Software (in Chinese)*, vol. 34, no. 3, pp. 316–328, 2017.

[7] H. Kang, W. Zhu, "Privacy protection of location services," *Journal of Shandong University (Science Edition)(in Chinese)*, vol. 53, no. 11, pp. 35–50, 2018.

[8] S. Liu, A. Liu, Z. Yan, W. Feng, "Efficient LBS queries with mutual privacy preservation in IoV," *Vehicular Communications*, vol. 16, pp.62–71, 2019.

[9] Y. Lin, Y. Xie, Y. Zhu, *et al.*, "Design and implementation of Hilbert's position k-anonymity algorithm based on spatial quartering," *Wuhan University Journal (Science Edition) (in Chinese)*, vol. 64, no. 3, pp. 225–230, 2018.

[10] S. Ni, M. Xie, Q. Qian, "Clustering based k-anonymity algorithm for privacy preservation," *International Journal of Network Security*, vol. 19, no. 6, pp. 1062–1071, 2017.

[11] A. S. Saxena, D. Bera, V. Goyal, "Modeling location obfuscation for continuous query," *Journal of Information Security and Applications*, vol. 44, pp. 130–143, 2019.

[12] G. Suna, V. Changc, *et al.*, "Efficient location privacy algorithm for Internet of Things (IoT) services and applications," *Journal of Network and Computer Applications*, vol. 89, pp. 3–13, 2017.

[13] D. Wu, X. Lu, "Anonymous region construction algorithm based on user cooperation in distributed architecture," *Computer Science (in Chinese)*, vol. 41, no. 4, pp. 90–94, 2019.

[14] X. Wu, M. Luo, "Privacy protection method based on location service in sparse environment," *Computer Engineering (in Chinese)*, vol. 43, no. 5, pp. 108–114, 2017.

[15] P. Xie, T. Fu, *et al.*, "An algorithm of the privacy security protection based on location service in the Internet of Vehicles," *International Journal of Network Security*, vol. 21, no. 4, pp. 556–565, 2019.

[16] M. Xu, H. Zhao, X. Ji, W. Shen, "Mobile P2P fast location anonymity algorithm based on user distribution perception," *Journal of Software (in Chinese)*, vol. 29, no. 7, pp. 1852–1862, 2018.

[17] Z. Zhai, X. Li, H. Liu, K. Lei, J. Ma, H. Li, "An anonymous zone construction scheme based on query range in LBS privacy protection," *Journal on Communications (in Chinese)*, vol. 38, no. 9, pp. 1311–1318, 2017.

[18] Y. B. Zhang, Q. Y. Zhang, Z. Y. Li, Y. Yan, and M. Y. Zhang, "A k-anonymous location privacy protection method of dummy based on geographical semantics," *International Journal of Network Security*, vol. 21, no. 6, pp. 937–946, 2019.

[19] L. Zhang, C. G. Ma, S. T. Yang, Z. P. Li, "CP-ABE based users collaborative privacy protection scheme for continuous query," *Journal on Communications*, vol. 38, no. 9, pp. 76–85, 2017.

[20] Y. Zhou, J. Du *et al.*, "Location privacy nearest neighbor query method based on Veno diagram," *Journal of Beijing University of Technology (in Chinese)*, vol. 44, no. 2, pp. 225–233, 2018.

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.

Xue-ming Han was born in Jan. 1990. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: hxmhan@163.com.

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

Yan Yan was born in Oct. 1980. She is a associate professor and a supervisor of master student at Lanzhou University of Technology. Her major research field is privacy protection, multimedia information security.E-mail: yanyan@lut.cn.

Guo-qiang Ma was born in Jun. 1992. He is a master student at Lanzhou University of Technology. His major research field is network and information security.E-mail: magq1514@163.com.

Chaotic NHCP: Building an Efficient Secure Framework for Cloud Computing Environment Based on Chaos Theory

Diaa Salama Abdul Minaam, Mostafa Abdullah Ibrahim, and Elsayed Badr

(Corresponding author: Diaa Salama Abdul Minaam)

Information Systems Department, Faculty of Computers and Informatics, Benha University
Benha City, Egypt
(diaa.salama@fci.bu.edu.eg)

(Received Dec. 4, 2018; Revised and Accepted June 5, 2019; First Online Jan. 21, 2020)

Abstract

Cloud computing is an advanced trend, which provides access to applications and resources over the internet. In a cloud computing environment, the data is stored on remote servers accessed through the internet. The increasing volume of necessary data brings up more focus on securely storing data. Encryption plays a vital role in security for different types of data. The existing methods encrypt all data using the same key without taking into account the confidentiality level of data, which in turn will increase the encryption time. In this research, a novel encryption algorithm based on chaos theory in the cloud computing environment is developed. The new hybrid cryptography algorithm based on chaotic mapped called (Chaotic NHCP). Chaotic NHCP uses a classification method. The new framework of data encryption operates as follows, Firstly, KNN method is used to classify the data credibility level, and then Fast RSA algorithm and blowfish algorithm are used to encrypt the data to achieve the effect of Fast data encryption. The objects are classified by a maximum value of its neighbours, with the object being assigned to the class with most common among its K-nearest neighbours. Then, the 32-bit plaintext data was split into two 16-bit plaintext data, and the 32-bit ciphertext data was synthesised after encryption by Fast RSA and Blowfish hybrid algorithm, respectively. The proposed method was tested with different encryption algorithms and evaluated according to the encryption time, throughput and power consumption. The experimental results show that the Chaotic NHCP method minimises the encryption time needed to secure data that leads to a suitable confidentiality level required for the data. In addition, it has high throughput and low power consumption along with time-saving. The proposed method has proven the superior in the performance of processing time when compared with other encryption algorithms.

Keywords: Chaotic Map; Classification; Cloud Computing; Fast RSA; Hybrid Cryptography Algorithms

1 Introduction

Today, cloud computing has become an incoming trend for many organisations and people as it provides a wide range of services such as, Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS) [52]. Cloud computing has many issues, and the most important ones are security and confidentiality. Confidentiality level of data is not taken into consideration in some cloud systems, which leads to encrypt additional or unrelated data [20, 52]. Figure 1 shows cloud service models [8, 12, 50].

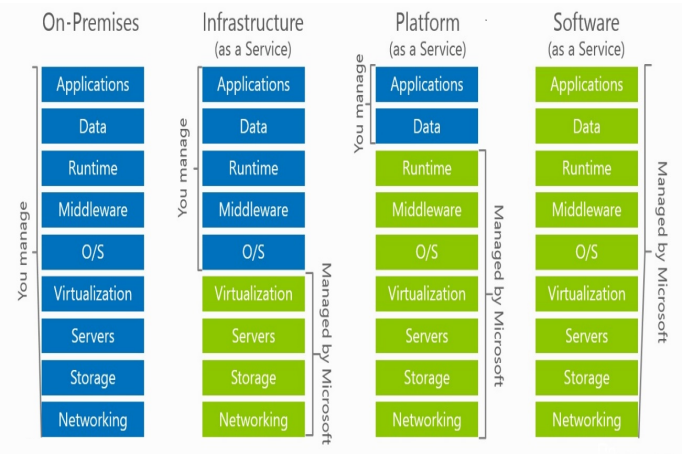


Figure 1: Cloud service models

In cloud computing, there are a lot of security challenges [31, 44, 45, 53] such Confidentiality, Privacy [1, 48], Data location [2, 3].

Encryption algorithms have two types: Symmetric and asymmetric key algorithms. Symmetric key algorithms uses the same key for encryption and decryption [16, 33]. DES, AES and Triple-DES, Blowfish [37, 49] are examples of symmetric key algorithms. Asymmetric algorithms have two keys; public key and private key for both encryption and decryption. RSA, Diffie-Hellman and homomorphic encryption are examples of asymmetric key algorithms. Symmetric algorithms are faster in performance than asymmetric algorithms because its key size is small. On the other hand, Symmetric algorithms have some drawbacks such as key transportation, as the key is transmitted to the received system before the original message is transmitted. Figure 2 shows the structure of the Blowfish [19, 32].

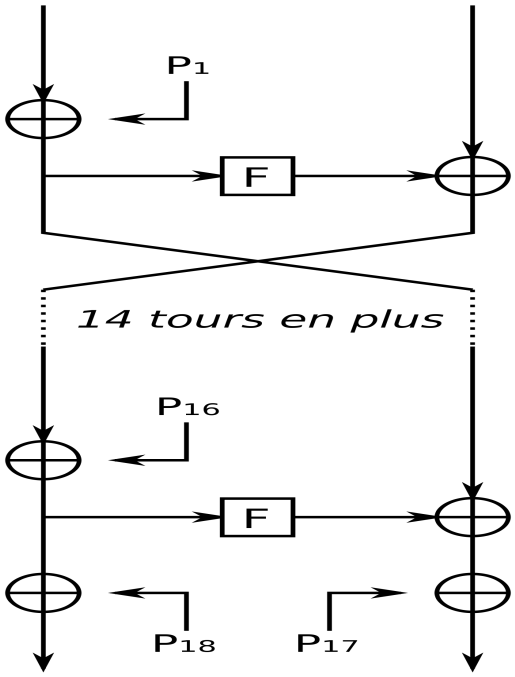


Figure 2: Structure of blowfish

In asymmetric algorithms, there is no need to exchange keys, thus solving the key distribution problem [44] of symmetric encryption algorithms. The primary advantage of public-key algorithms is increased security [48]. On the other hand, a disadvantage of using public-key cryptography for encryption is speed; There are secret-key encryption methods which are faster than currently available public-key encryption algorithm. RSA algorithm is illustrated in Figure 3. Disadvantages of symmetric and asymmetric encryption algorithms have motivated us to apply hybrid encryption algorithm.

Fast RSA [14, 23, 47] uses a modulus in form N = pr qs such that p, q are two distinct primes and r, s ¿= 2. It consists of the main three steps key generation, encryption, and decryption [18, 30]. So the main objective of this paper is to study the problem of data encryption algorithm based on chaos theory in the cloud computing environment and proposes a new framework of data encryption. Firstly, the KNN method is used to classify the data credibility level, and then Fast RSA algorithm and the Blowfish algorithm are used to encrypt the data to achieve the effect of Fast data encryption. An object is classified by a maximum value of its neighbours, with the object being assigned to the class with most common among its K nearest neighbours, which is named KNN. Then, the 32-bit plaintext data is splitted into two 16-bit plaintext data, and the 32-bit ciphertext data is synthesized after encryption by Fast RSA and Blowfish hybrid algorithm respectively.

1.1 Chaos Theory

Chaos theory [9] is a branch of mathematics that focuses on the behaviour of dynamic systems that are sensitive to initial conditions. It aims to predict the unexpected [36], and it concerns deterministic systems whose behaviour can be predicted [26]. Chaotic systems are predictable for a while and then 'appear' to become random. The amount of time that the behaviour of a chaotic system can be predicted depends on three factors: How much uncertainty can be tolerated in the forecast, how accurately its current state can be measured, and a time scale depending on the dynamics of the system. Chaos theory is based on the observation that simple rules when iterated can give rise to complex behaviour according to the following equation.

$$X_{N+1} = X_N(\text{mod } 1) where 0 \leq X_N \leq 1$$

Chaotic systems are sensitive to the control parameters and initial conditions; Therefore, it can be connected with some cryptographic features of good cyphers, such as diffusion and balance property. When comparing chaos with other traditional methods, the ones based on chaos theory are suitable for extensive data such as images and videos. Also, the chaos-based method has achieved excellent performance, and it is recommended for many cryptosystems. A chaotic system is considered as a symmetric block cipher. There are two methods of chaotic systems: analogue and digital. A chaotic digital system has a significant concern in the digital world [28,29]. In this paper, a matrix element M1Xi is encrypted in every round as follows:

$$C_{1Xi} = M_{1Xi} XOR(X\tilde{f} \text{ mod } 256).$$

One of the commonly used maps in chaos theory is the logistic map as described below.

$$X_{n+1} = rX_n\ (1 - X_n).$$

Where the parameter r belongs to the interval [0, 4] and determines the mapping behaviour, while n is the iteration number that determines the time.

The significant advantage of a chaotic system over a noisy one is that the chaotic system is deterministic; Therefore, the knowledge of system parameters and initial conditions enables one to recover a message [21].

Confusion and diffusion are related to the fundamental characteristics of chaos theory, and any strong cryptosystem should consider features of chaos or pseudo-randomness. Chaotic synchronisation is a type of chaotic systems.

Analogue implementation is an excellent advantage of chaotic synchronisation schemes. Chaotic communication offers the advantage of message waveform encryption without a need to digitalise it [7].

The following equation expresses confusion and diffusion processes

$$R = D^{\alpha}(C^{\beta}(P, K_C), K_D).$$

Where P and R are respectively plain text and cypher text, C and D are the confusion and diffusion functions, K_C and K_D are the confusion and diffusion keys, and α and β are numbers of rounds for total encryption and confusion, respectively. The chaotic map uses parameters as keys to providing high security.

1.2 Classification

Classification is the process of categorising data based on different classes [17]. One of the main classification techniques is a K-Nearest Neighbour (KNN). In this paper, we applied classification by KNN as it has high accuracy at K = 3, as mentioned in Section 5. Classification techniques can be parametric, semiparametric and non-parametric. For classification, a useful technique can be used to assign a weight to the contributions of the neighbours, so that the nearer neighbours contribute more to the average than the more distant ones [24].

K-nearest neighbour algorithm (k-NN) is a non-parametric method used for classification. The input consists of the k closest training examples in the feature space. KNN uses Euclidean distance to calculate the distance between two points of test data and training data [13]. The training examples are vectors in a multi-dimensional feature space, each with a class label. The training phase of KNN consists of storing the feature vectors and class labels of the training samples. K-nearest neighbour is considered as a type of instance-based learning, where the function is only approximated locally, and all computation is deferred until classification. It is easy to implement and apply for training data. KNN is good against noisy training data and is efficient if the training data is astronomical.

The rest of this paper is organised as follows. In the next section, we give a brief review of some related work. In Section 3, we introduce our proposed method. In Section 4, we give evaluation matrices. In Section 5, we give results. In Section 6, we discuss our results. Finally, we present our conclusions.

2 Related Work

A lot of different approaches proposed recently focusing on the challenges of security issues on cloud computing by using different encryption techniques. Some of these methods only use a single encryption techniques methods and other used hybrid encryption. In [6] uses FHE algorithm as the encryption is performed on the ciphertext. The system solves the security problem for stored data in the cloud.

Encrypted the data by a key is proposed in [46] that is not available for the provider. It based on the idea of manual classification and addressed data confidentiality problem. It compared with AES 128 and AES 256 with SHA 2. The results show that it achieved less processing time when compared with AES 128 and AES 256. While in [39], a model depends on simple key generation by an arbitrary matrix is proposed.

In [15] proposed a framework using fast RSA to provide security to the data in the cloud. This algorithm increases the speed up time for encryption and decryption when compared with RSA.

A hybrid cryptography algorithm is proposed in [25] that uses AES for file uploads and file download. AES key is encrypted using the RSA algorithm. In [41], the authors combine the DES algorithm, followed by a CAST encryption algorithm to achieve data protection.

In [5] applies Blowfish with a different number of rounds to achieve better security and reduce hacking while in [51] applied the ElGamal algorithm to enhance cloud security and allows encrypting ciphertext in two levels. [42] presents a new security framework for achieving data security. Data is split into blocks of bits. Genetic algorithm is applied to every two blocks of bits. The final output of every genetic algorithm is a cypher text, which is also two blocks of bits. Each cypher text is stored on the cloud at a distinct location. In [22] applies setup, keygen, encrypt and decrypt algorithms to perform encryption operations on ciphertext using the private key and public key. It applies two-party computation 2PC protocols between Key Generation Center and data storing centre to ensure security.

All mentioned methods used a single algorithm and manual classification to deal with security issues. However, we applied a hybrid encryption algorithm and classifier such as the K-Nearest Neighbor. Table 1 represents a summary of related work.

3 Proposed Method

Our proposed method is based on chaotic map and classification. Chaotic map depends on chaos theory. The chaotic map can generate values of low cost with simple iterations, which makes it suitable for the construction of stream ciphers. Therefore, cryptosystem can provide a fast and secure means for data encryption.

Table 1: Comparison between different security frameworks

	[6]	[46]	[10]	[39]	[15]	[38]	[25]
Parameter	FHE	Multi-cloud	Secure cloud model	probabilistic encryption	Fast RSA	Proposed algorithm symmetric	Proposed model
Algorithm used	Homomorphic Encryption	RSA	AES and SHA	probabilistic encryption	Fast RSA	Symmetric algorithm	AES
Applied security on cloud	Yes	Yes	Yes	No	No	Yes	Yes
Used chaos theory	No	No	No	No	No	No	No
Used hybrid algorithm	No	No	Yes	No	No	No	No
Performance	Complexity less than CAST-128	More secure when compared to regular system	Less processing time	Less encryption time when compared to AES and DES	Less encryption time when compared to cloud RSA	Less encryption time when compared to AES	file upload has less time than a file download

	[41]	[5]	[11]	[51]	[42]	[4]	[27]	[40]
Parameter	Hybrid DES&CAST	Recursive blowfish	Homomorphic Encryption	ElGamal	New security framework	Data splitting mechanism	homomorphic token and error correcting codes	Protection model
Algorithm used	DES&CAST	Enhanced blowfish	Homomorphic Encryption	ElGamal	Genetic algorithm	AES	homomorphic token and error correcting codes	AES
Applied security on cloud	No	No	Yes	Yes	No	Yes	Yes	Yes
Used chaos theory	No	No	No	No	No	No	No	No
Used hybrid algorithm	Yes	No	No	No	No	No	No	No
Performance	High encryption time when compared to DES	More secure than standard blowfish	More secure	More secure	More secure and efficient	Safer than similar methods	Safer	More secure

3.1 The Proposed Chaotic Encryption Algorithm (Chaotic NHCP)

The following sub-section illustrated the basic steps for key generation, encryption and decryption methods as essential building blocks for the proposed algorithm.

Block 1: Key generation

Input: $X_0 = 0.01$ and $r = 3.99$

Step 1: Compute $X_{n+1} = r\, X_n\, (1\text{-}X_n)$, where r = 3.99 and x ranges from 0 to 1

Step 2: Binary sequence can be [0.232,0.243 ,.632,0.729,0.385]

Step 3: Compute K_1 = each number in binary sequence X 255

Step 4: After that [59, 62,161,186, 98] is generated

Step 5: Convert sequence [59, 62,161,186, 98] to binary numbers.

Step 6: Then, XOR operation is executed on bit 0, bit 4, bit 5, bit 6 in k1 to generate k2.

Step 7: Compute Key = k_1 XOR K_2

Output: Key

Block 2: Encryption

Input: private key k and plaintext p

Ciphertext = Plaintext XOR Key

Output: Ciphertext m

Block 3: Decryption

Input: private key k and ciphertext m

Plaintext = Cipher text XOR Key

Output: Plaintext p

3.2 Classification

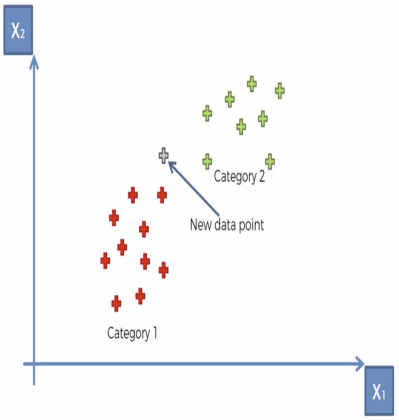


Figure 3: K-nearest neighbor

In this paper, we deal with the impossibility of encrypting all data without taking into account its confidentiality degree. So, we encrypt data based on the degree of confidentiality. We can take into consideration the degree of confidentiality in classifying data for saving the processing time. We applied classification by K-Nearest Neighbor (KNN). We classified data as highly sensitive or less sensitive. The output KNN is a class membership. An object is classified by a maximum value of its neighbours, with the object being assigned to the class with most common among its k nearest neighbours. If $k =\sim 1$, then the object is assigned to the class of the single nearest neighbour. KNN is illustrated in Figure 3.

3.3 Building Hybrid Cryptography Algorithms (NHCP)

After the classification process, we applied a hybrid algorithm which combines both Fast RSA and Blowfish cipher algorithm. The goal of the hybrid algorithm is to encrypt data efficiently, and this can reduce encryption time. Two encryption algorithms were implemented in the hybrid cryptography algorithm. These algorithms are implemented to improve the efficiency of encryption algorithm security and processing time. Hybrid encryption algorithm provides security since it encrypts data by two algorithms. It offers the advantage of reducing encryption time as FastRSA is an asymmetric algorithm and Blowfish is a symmetric one. By this way, data size is reduced to half. Figure 4 shows the encryption process for the hybrid algorithm as below.

1) 32-bit plaintext is divided into plaintext1 and plaintext 2;
2) FastRSA is used to encrypt plaintext1 generating ciphertext1;
3) Blowfish is used to encrypt plaintext2 generating ciphertext2;
4) Ciphertext1 and ciphertext2 are combined into 32-bit ciphertext.

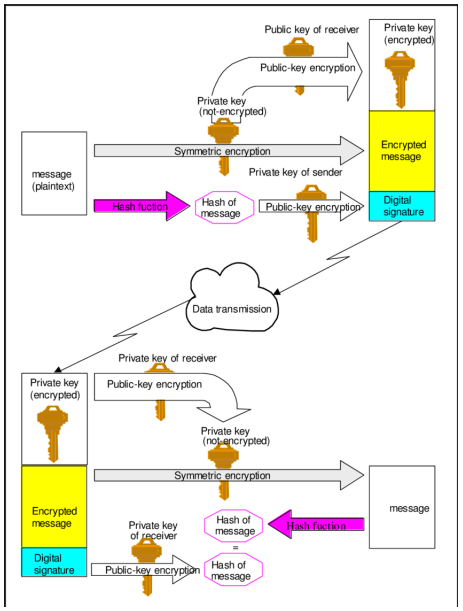


Figure 4: Hybrid algorithm using Fast RSA and blowfish

3.3.1 Proposed Encryption Algorithm

Input: M (Plain text), k(secret key of FastRSA encryption), s(32 bit size of block).

Output: C (Cipher text), ci (encrypted text using FastRSA), Ci (encrypted text using Blowfish).

1: $n = M/s$;
2: let $i = 0$;
3: do{
4: $m = \sum_{i=0}^{i=\frac{n}{2}-1}(Bi)$ the first part of plain text;
5: for $(j = 0; j <= n - I; j++)$
6: $c_i = E_{\text{FastRSA}}\ (K_j, B_i)$
7: $i++$;
8: }
9: while $(i < n/2)$;
10: $i = (n/2)$
11: let K be a private key of Blowfish
12: do {
13: $M = \sum_{i=n/2}^{i=n}(Bi)$ the second part of plain text which encrypted simultaneously with the first part ;
14: $C_i = E_{Blowfish}\,(K_j, B_i)$
15: $i++$;
16: }
17: while $(i < n)$
18: $C = c_i + C_i$

Where n is a number of blocks, i is a counting number, $(\boldsymbol{K})$ is Private key of Blowfish for the encryption process.

4 Evaluation Metrics

In order to evaluate the proposed algorithm, some performance metrics are used such as encryption time, throughput, battery power and Accuracy.

- The encryption time is considered the time that an encryption algorithm takes to produce a ciphertext from a plaintext. Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption.
- The throughput of the encryption scheme is calculated as in Equation (1).

$$\text{Throughput} = \frac{T_p}{E_t} \tag{1}$$

Where T_p : total plain text bytes) and E_t : encryption time (second).

The CPU process time is the time that a CPU is committed only to the particular process of calculations. It reflects the load of the CPU.

The CPU clock cycles are a metric, reflecting the energy consumption of the CPU while operating on encryption operations. Each cycle of CPU will consume a small amount of energy.

Measurement of Energy Consumption.

Energy consumption of security primitives can be measured in many ways. The used method can be measured by counting the number of computing cycles which are used in computations related to cryptographic operations. For the computation of the energy cost of encryption, we use the same techniques as described in the following equations.

$$\mathbf{B}_{\text{ cost encryption}}\ (\text{ampere-cycle}) = \boldsymbol{\tau} * \mathbf{I}$$

$$\mathrm{T}_{\text{energy cost}(ampere-seconds)} = \frac{B_{costencryption}(ampere-cycle)}{F(cycles/sec)}$$

$$\mathrm{E}_{\text{cost}}\ (\text{Joule}) = \mathrm{T}_{\text{energy cost}}(ampere-seconds) * V$$

Where

- B_cost_encryption: A basic cost of encryption (ampere-cycle).
- τ: The total number of clock cycles.
- I: The average current drawn by each CPU clock cycle.
- Tenergy_cost: The total energy cost (ampere-seconds).
- F: Clock frequency (cycles/sec).
- E_cost (Joule): The energy cost (consumed).

By using the cycles, the operating voltage of the CPU, and the average current drawn for each cycle, we can calculate the energy consumption of cryptographic functions. For example, on average, each cycle consumes approximately 270 mA on an Intel 486DX2 processor [34] or 180 mA on Intel StrongARM [43]. For a sample calculation, with a 700 MHz CPU operating at 1.35 Volt, encryption with 20,000 cycles would consume about 5.71 x 10-3 mA-second or 7.7 μ Joule. So, the amount of energy consumed by program P to achieve its goal (encryption or decryption) is given by

$$\mathbf{E} = \mathbf{V_{cc}} \times \boldsymbol{I} \times \boldsymbol{N} \times \boldsymbol{\tau}.$$

Where N: The number of clock cycles, τ: the clock period. $\boldsymbol{V}_{CC}$: The supply voltage of the system, I: The average current in amperes drawn from the power source for T seconds.

Since for a given hardware, both $\boldsymbol{V}_{CC}$ and τ are fixed $E \propto I \times N$. However, at the application level, it is more meaningful to talk about T than N, and therefore, we express energy as $E \propto I \times T$. Since for a given hardware V_{cc} are fixed [35].

Accuracy is one of the measures for evaluating classification models. Accuracy is the fraction of predictions our model got right. Accuracy=Number of correct predictions / Total number of predictions (2). Accuracy: It measures the correctness according to the following

$$\text{Accuracy} = (\mathbf{TP} + \mathbf{TN})/(\mathbf{TP} + \mathbf{N} + \mathbf{FP} + \mathbf{FN}). \tag{2}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

4.1 Experiments and Performance Analysis

Algorithms are implemented using Python programming language in Windows-10, the 64-bit operating system on a 2.20 GHz processor using 8GB RAM to analyse their performance. A twenty-two text different file size ranges from 8 KB to 15 MB.The twenty-two text files of different sizes are used to carry out the experiment, where we evaluate the performance of different algorithms AES, DES, chaotic and hybrid algorithm. The experiments are conducted on the test system. These implementations are thoroughly tested and are optimised to give the maximum performance for each algorithm. The performance of these algorithms is evaluated based on parameters like encryption time, throughput and power consumption.

The size of the ciphertext. Table 2 describes the output of the encryption process. It shows the size of the ciphertext in bytes.

Table 2: Size of cipher text (bytes)

Hybrid(Fast RSA+ Blowfish)	Chaotic	DES	AES	File size in KB
12	8	6	8	8
20	16	14	16	16
36	32	30	32	32
52	48	46	48	48
68	64	62	64	64
84	80	78	80	80
104	100	98	100	100
204	200	198	200	200
304	300	298	300	300
404	400	398	400	400
504	500	498	500	500
604	600	598	600	600
804	800	798	800	800
1.2 MB	1 MB	0.8 MB	1 MB	1 MB
2.2	2	1.8	2	2
3.2	3	2.8	3	3
5.2	5	4.8	5	5
7.2	7	6.8	7	7
9.2	9	8.8	9	9
11.2	11	10.8	11	11
13.2	13	12.8	13	13
15.2	15	14.8	15	15

Time of encryption and decryption processes. The encryption time is the time that an encryption algorithm takes to produce a ciphertext from a plaintext. The decryption time is the time that a decryption algorithm takes to produce a plaintext from a ciphertext.

Table 3 and Figure 5 show the time of the encryption process for different sizes of plain text. It is shown that proposed hybrid cryptography protocol based on chaotic map (Chaotic NHCP) achieve the least time for encryption followed by Hybrid between Fast RSA and Blowfish (NHCP). Table 4 and Figure 6 show the time of decryption process for different sizes of plain text. As in the encryption, it is clear that Chaotic NHCP achieve the least time for decryption followed by (NHCP).

Table 3: Encryption time (seconds) of cryptographic algorithms

Hybrid(Fast RSA+ Blowfish)	Chaotic	DES	AES	File size in KB
0.01	0.001	.04	.1	8
0.02	0.002	.055	.2	16
0.03	0.003	.07	.23	32
0.04	0.004	0.11	.26	48
0.05	0.005	0.13	0.33	64
0.06	0.006	0.15	0.41	80
0.07	0.008	0.17	0.5	100
0.13	0.01	0.27	0.6	200
0.19	0.012	0.37	0.7	300
0.25	0.014	0.47	0.8	400
0.31	0.016	0.57	0.9	500
0.37	0.018	0.66	1	600
0.5	0.025	0.8	1.15	800
0.6	0.043	0.9	1.3	1 MB
1.2	0.08	1.8	2.1	2
1.8	0.13	2.6	3	3
3	0.23	4.2	4.8	5
4.2	0.33	5.8	6.6	7
5.4	0.43	7.4	8.4	9
6.6	0.53	9	10.2	11
7.8	0.63	10.6	12	13
9	0.73	12.2	13.8	15

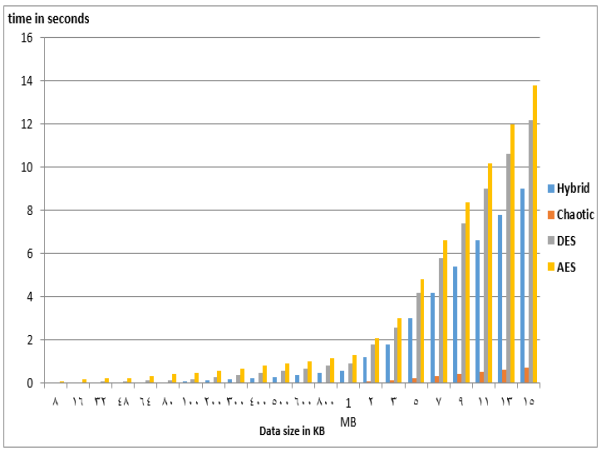


Figure 5: Encryption time of cryptographic algorithms

So it can be concluded from Table 3, Table 4, Figure 5, and Figure 6 that Chaotic NHCP and NHCP has encryption time and decryption time less than AES and DES. Chaotic NHCP has the least encryption time and decryption time.

Throughput. Encryption time is used to calculate the

Table 4: Decryption time (seconds) of cryptographic algorithms

Hybrid(Fast RSA+ Blowfish)	Chaotic	DES	AES	File size in KB
0.009	0.0009	.02	.07	8
0.01	0.001	.035	.17	16
0.02	0.002	.05	.2	32
0.03	0.003	0.09	.23	48
0.04	0.004	0.11	0.3	64
0.05	0.005	0.13	0.38	80
0.06	0.007	0.15	0.47	100
0.12	0.009	0.25	0.57	200
0.18	0.011	0.35	0.67	300
0.24	0.013	0.45	0.77	400
0.3	0.015	0.55	0.87	500
0.36	0.017	0.64	.89	600
0.49	0.024	0.78	.92	800
0.59	0.042	0.88	1	1 MB
1.19	0.07	1.78	1.8	2
1.79	0.12	2.58	2.7	3
2.99	0.22	4.18	4.5	5
4.19	0.32	5.78	6.3	7
5.39	0.42	7.38	8.1	9
6.58	0.52	8.98	9.9	11
7.78	0.62	10.58	11.7	13
8.98	0.72	12.18	13.5	15

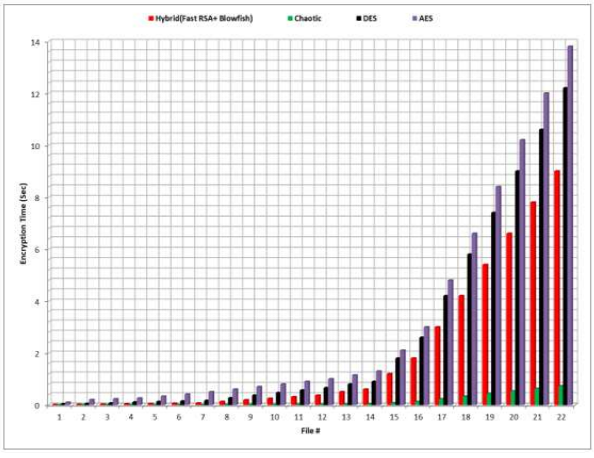


Figure 6: Decryption time of cryptographic algorithms

throughput of an encryption scheme. It indicates the speed of encryption. Table 5, and Figure 7 show that the encryption throughput of the proposed hybrid cryptography algorithm based on chaotic map (Chaotic NHCP) is more significant than other algorithms for different sizes of plain text. It is shown that both (Chaotic NHCP) and NHCP achieve the most significant values.

Table 5: Encryption throughput (KB/second) of cryptographic algorithms

File size in KB	AES	DES	Chaotic	Hybrid(Fast RSA+ Blowfish)
8	80	200	8000	800
16	80	290.91	8000	800
32	139.13	457.14	10666.67	1066.67
48	184.62	436.36	12000	1200
64	193.94	492.31	12800	1280
80	195.12	533.33	13333.33	1333.33
100	200	588.24	12500	1428.57
200	333.33	740.74	20000	1538.46
300	428.57	810.81	25000	1578.95
400	500	851.06	28571.43	1600
500	555.56	877.19	31250	1612.9
600	600	909.09	33333.33	1621.62
800	695.65	1000	32000	1600
1 MB	787.69	1137.78	23813.95	1706.67
2	975.24	1137.78	25600	1706.67
3	1024	1181.54	23630.77	1706.67
5	1066.67	1219.05	22260.87	1706.67
7	1086.06	1235.86	21721.21	1706.67
9	1097.14	1245.41	21432.56	1706.67
11	1104.31	1251.56	21252.83	1706.67
13	1109.33	1255.85	21130.16	1706.67
15	1113.04	1259.02	21041.1	1706.67

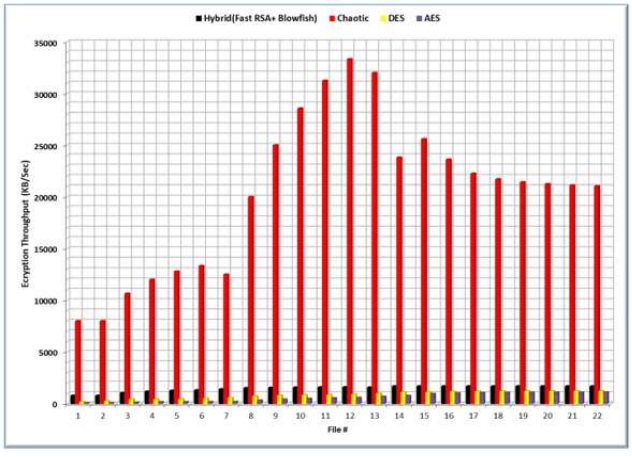


Figure 7: Encryption throughput of cryptographic algorithms(KB/Sec)

Table 6, and Figure 8 also show that the decryption throughput of the proposed hybrid cryptography algo-

rithm based on chaotic map (Chaotic NHCP) is more significant than other algorithms for different sizes of plain text. It is shown that both (Chaotic NHCP) and NHCP achieve the most significant values.

Table 6: Decryption throughput (KB/Second) of cryptographic algorithms

File size in KB	AES	DES	Chaotic	Hybrid(Fast RSA+ Blowfish)
8	114.29	400	8888.89	888.89
16	94.12	457.14	16000	1600
32	160	640	16000	1600
48	208.7	533.33	16000	1600
64	213.33	581.82	16000	1600
80	210.53	615.38	16000	1600
100	212.77	666.67	14285.71	1666.67
200	350.88	800	22222.22	1666.67
300	447.76	857.14	27272.73	1666.67
400	519.48	888.89	30769.23	1666.67
500	574.71	909.09	33333.33	1666.67
600	674.16	937.5	35294.12	1666.67
800	869.57	1025.64	33333.33	1632.65
1 MB	1024	1163.64	24380.95	1735.59
2	1137.78	1150.56	29257.14	1721.01
3	1137.78	1190.7	25600	1716.2
5	1137.78	1224.88	23272.73	1712.37
7	1137.78	1240.14	22400	1710.74
9	1137.78	1248.78	21942.86	1709.83
11	1137.78	1254.34	21661.54	1711.85
13	1137.78	1258.22	21470.97	1711.05
15	1137.78	1261.08	21333.33	1710.47

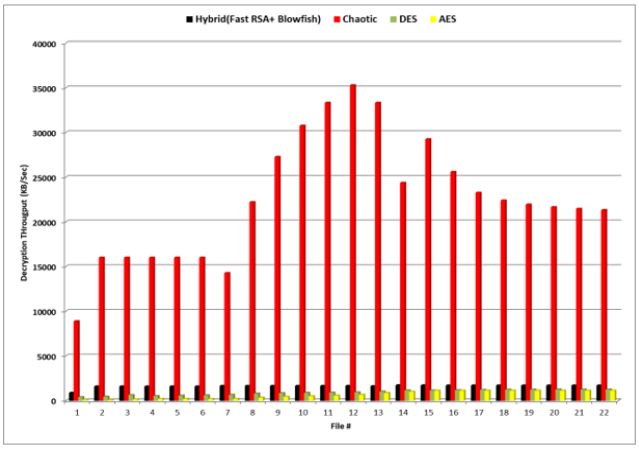


Figure 8: Decryption throughput (KB/Second) of cryptographic algorithms

Power consumption. It is noticed from Table 7, and Figure 9 chaotic NHCP, and NHCP has the least power consumption.

Accuracy of KNN depends on the value of k; in our case, $K = 3$.KNN with $K = 1$ gives better results and

Table 7: Power consumption (watt) for encryption of different cryptographic algorithms

File size in KB	AES	DES	Chaotic	Hybrid(Fast RSA+ Blowfish)
8	0.66	0.264	0.0066	0.066
16	1.32	0.363	0.0132	0.132
32	1.5	.462	0.02	0.2
48	1.7	0.7	0.03	0.3
64	2.18	0.86	0.033	0.33
80	2.7	1	0.04	0.4
100	3.3	1.12	0.053	0.46
200	3.96	1.8	0.066	0.86
300	4.6	2.4	0.08	1.25
400	5.28	3.1	0.1	1.65
500	6	3.76	0.106	2
600	6.6	4.36	0.12	2.44
800	7.6	5.28	0.17	3.3
1 MB	8.6	6	0.28	4
2	12.86	11.88	0.53	8
3	20	17.16	0.86	11.88
5	31.68	27.72	1.518	19.8
7	43.56	38.28	2.178	27.72
9	55.44	48.84	2.838	35.64
11	67.32	59.4	3.498	43.56
13	79.2	69.96	4.158	51.48
15	91.08	80.52	4.818	59.4

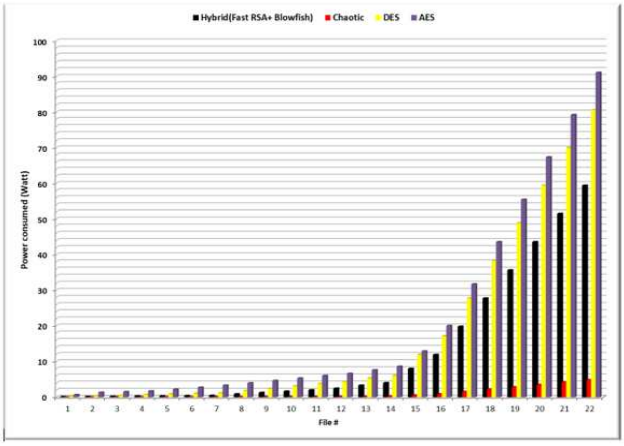


Figure 9: Power consumption (watt) of cryptographic algorithms

accuracy. KNN requires that classes can be separable to provide excellent results.

Results analysis. The results show the superiority of (Chaotic NHCP) algorithm over other algorithms in terms of the power consumption, processing time, and throughput followed by NHCP in case of encryption and decryption -(when the same data is encrypted by using DES and AES. it is found that NHCP requires approximately 60% of the time used for encryption which is consumed for AES and 71% in case of compared by DES). Another point can be noticed that Chaotic NHCP requires approximately 5% of the time used for encryption, which is consumed for AES and 6% in the case of comparing by DES).

In the case of decryption, the results also show the superiority of (Chaotic NHCP) algorithm over other algorithms in terms of decryption time. It is found that NHCP requires approximately 62.7% of the time used for encryption, which is consumed for AES and 71.5% in case of compared by DES). Another point can be noticed that Chaotic NHCP requires approximately 4.8% of the time used for encryption, which is consumed for AES and 5.4% in the case of comparing by DES).

In the case of power consumption for encryption, the results also show the superiority of (Chaotic NHCP) algorithm over other algorithms in terms of power consumption. It is found that NHCP requires approximately 60.12% of the time used for encryption, which is consumed for AES and 71.35% in case of compared by DES). Another point can be noticed that Chaotic NHCP requires approximately 4.7% of the time used for encryption, which is consumed for AES and 5.58% in case of compared by DES).

In case of power consumption for decryption, the results also show the superiority of (Chaotic NHCP) algorithm over other algorithms in terms of decryption time. It is found that NHCP requires approximately 61.4% of the time used for encryption, which is consumed for AES and 70.92% in case of compared by DES). Another point can be noticed that Chaotic NHCP requires approximately 4.6% of the time used for encryption, which is consumed for AES and 5.48% in the case of comparing by DES). Finally, It is shown from experimental results that the chaotic encryption algorithm is the fastest algorithm among other cryptographic algorithms.

The chaotic map has the least encryption time as it depends on simple operations like XOR, multiplication and logistic function. It uses a logistic function to generate random values that are used to produce key k. The encryption algorithm that has the least encryption time is the best algorithm. It can have the most value of throughput, and the least value of power consumption Nearest Neighbor (KNN) classifier has high accuracy as it has 83% when $K = 3$, as shown in Figure 11. Both AES and DES use 16 rounds with XOR operation, and this leads to high encryption time. The hybrid algorithm has encryption time less than AES and DES. It merges both FastRSA and Blowfish. FastRSA uses a modulus of the form N=prqs, so it has less encryption time. On the other hand, blowfish uses F function with 16 rounds.

5 Conclusion

In this paper, a novel secured, the optimised framework is proposed to improve the efficiency of security of the data to the cloud. This framework design an encryption method based on chaotic theory (chaotic NHCP) that reduces the encryption time and ensures confidentiality through data classification and a hybrid cryptographic algorithm (NHCP) that merges fast RSA and Blowfish cryptographic algorithms. This study presents a performance evaluation of selected encryption algorithms on power consumption to be used to provide security for the cloud environment. The selected algorithms are AES, DES, NHCP, and chaotic NHCP. Several points can be concluded from the experimental results. The experiment with these parameters, such as encryption time, throughput, and power consumption, is done, and those results show that chaotic NHCP has better performance to other cryptographic algorithms. Performance evaluation of selected this study presents a performance evaluation of selected encryption algorithms on power consumption to be used to provide security for the cloud environment. The selected algorithms are AES, DES, NHCP, and chaotic NHCP. Several points can be concluded from the experimental results. The experiment with these parameters, such as encryption time, throughput, and power consumption, is done, and those results show that chaotic NHCP has better performance to other cryptographic algorithms. Performance evaluation of selected encryption algorithms. Encryption algorithms. As shown in results, the chaotic map has the least encryption time; the hybrid algorithm has encryption time less than AES and DES. The data classification helps in decreasing the time of encrypting stored data. It is noticed from experimental results that K Nearest Neighbor (KNN) has high accuracy in the classification process. By comparing the result of this method with other cryptographic methods, we can recommend the implemented chaotic method to be used in securing data through cloud computing. We found that chaotic NHCP has better performance than other encryption algorithms, followed by NHCP in case of encryption time, throughput, and power consumption for encryption and decryption. Chaotic NHCP and NHCP are faster than DES, and AES. NHCP encrypts and decrypts data faster than DES and AES. Chaotic NHCP is faster than NHCP. These results are the same in encryption and decryption process with different packet size. So the chaotic NHCP and NHCP is sufficient to provide security on cloud computing.

References

[1] D. S. AbdElminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40–48, 2018.

[2] M. Aledhari, A. Marhoon, A. Hamad, and F. Saeed, "A new cryptography algorithm to protect cloud-based healthcare services," in *IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE'17)*, pp. 37–43, 2017.

[3] R. Arora, A. Parashar, "Secure user data in cloud computing using encryption algorithms," *International Journal of Engineering Research and Applications*, vol. 3, no. 4, pp. 1922–1926, 2013.

[4] V. R. Balasaraswathi and S. Manikandan, "Enhanced security for multi-cloud storage using cryptographic data splitting with dynamic approach," in *IEEE International Conference on Advanced Communications, Control and Computing Technologies*, pp. 1190–1194, 2014.

[5] N. Balkish, A. M. Prasad, and V. Suma, "An efficient approach to enhance data security in cloud using recursive blowfish algorithm," in *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I*, pp. 575–582, 2014.

[6] P. V. Bharati and T. S. Mahalakshmi, "Data storage security in cloud using a functional encryption algorithm," in *Emerging Research in Computing, Information, Communication and Applications*, pp. 201–212, 2016.

[7] M. Boumaraf and F. Merazka, "Speech encryption based on hybrid chaotic key generator for amr-wb g. 722.2 codec," in *The 12th International Conference for Internet Technology and Secured Transactions (ICITST'17)*, pp. 87–91, 2017.

[8] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[9] F. Dachselt and W. Schwarz, "Chaos and cryptography," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 12, pp. 1498–1509, 2001.

[10] N. S. Darwazeh, R. S. Al-Qassas, F. AlDosari, *et al.*, "A secure cloud computing model based on data classification," *Procedia Computer Science*, vol. 52, pp. 1153–1158, 2015.

[11] C. A. Dhote, "Homomorphic encryption for security of cloud data," *Procedia Computer Science*, vol. 79, pp. 175–181, 2016.

[12] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[13] S. E. El-Khamy, H. A. Elsayed, and M. M. Rizk, "Classification of multi-user chirp modulation signals using higher order cumulant features and four types of classifiers," in *The 28th National Radio Science Conference (NRSC'11)*, pp. 1–10, 2011.

[14] K. El-Makkaoui, A. Beni-Hssane, A. Ezzati, and A. El-Ansari, "Fast cloud-rsa scheme for promoting data confidentiality in the cloud computing," *Procedia Computer Science*, vol. 113, pp. 33–40, 2017.

[15] K. El-Makkaoui, A. Beni-Hssane, A. Ezzati, and A. El-Ansari, "Fast cloud-rsa scheme for promoting data confidentiality in the cloud computing," *Procedia Computer Science*, vol. 113, pp. 33–40, 2017.

[16] D. S. A. Elminaam, H. M. Abdual-Kader, and M. M. Hadhoud, "Evaluating the performance of symmetric encryption algorithms," *International Journal Network Security*, vol. 10, no. 3, pp. 216–222, 2010.

[17] A. Freier, P. Karlton, and P. Kocher, *The secure sockets layer (ssl) protocol version 3.0*, RFC 6101, 2011.

[18] K. Hansen, T. Larsen, and K. Olsen, "On the efficiency of fast RSA variants in modern mobile phones," *International Journal of Computer Science and Information Security*, vol. 6, no. 3, 2009.

[19] X. He, A. Machanavajjhala, and B. Ding, "Blowfish privacy: Tuning privacy-utility trade-offs using policies," in *Proceedings of the ACM SIGMOD international conference on Management of data*, pp. 1447–1458, 2014.

[20] W. F. Hsien, C. C. Yang and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133-142, 2016.

[21] M. Hu, H. Gao, and T. Gao, "Secure and efficient ranked keyword search over outsourced cloud data by chaos based arithmetic coding and confusion," *International Journal Network Security*, vol. 21, no. 1, pp. 105–114, 2019.

[22] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2271–2282, 2011.

[23] 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, Dec. 2003.

[24] J. P. Kandhasamy and S. Balamurali, "Performance analysis of classifier models to predict diabetes mellitus," *Procedia Computer Science*, vol. 47, pp. 45–51, 2015.

[25] Z. Kartit, A. Azougaghe, H. K. Idrissi, M. El-Marraki, M. Hedabou, M. Belkasmi, and A. Kartit, "Applying encryption algorithm for data security in cloud storage," in *International Symposium on Ubiquitous Networking*, pp. 141–154, 2015.

[26] A. Kumar and M. K. Ghose, "Overview of information security using genetic algorithm and chaos," *Information Security Journal: A Global Perspective*, vol. 18, no. 6, pp. 306–315, 2009.

[27] S. P. Kumar and R. Subramanian, "An efficient and secure protocol for ensuring data storage security in cloud computing," *International Journal of Computer Science Issues (IJCSI'11)*, vol. 8, no. 6, p. 261, 2011.

[28] C. Li, G. Luo, and C. Li, "A novel scheme for the preview of the image encryption based on chaotic ikeda map," *International Journal Network Security*, vol. 20, no. 6, pp. 1105–1114, 2018.

[29] C. Li, G. Luo, and C. Li, "An image encryption scheme based on the three-dimensional chaotic logistic map," *International Journal Network Security*, vol. 21, no. 1, pp. 22–29, 2019.

[30] C. Lu, A. L. M. dos Santos, and F. R. Pimentel, "Implementation of fast rsa key generation on smart cards," in *Proceedings of the ACM symposium on Applied computing*, pp. 214–220, 2002.

[31] H. Ma, X. Han, T. Peng, and L. Zhang, "Secure and efficient cloud data deduplication supporting dynamic data public auditing," *International Journal Network Security*, vol. 20, no. 6, pp. 1074–1084, 2018.

[32] P. C. Mandal, "Superiority of blowfish algorithm," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 9, 2012.

[33] D. S. A. Minaam, H. M. Abdual-Kader, and M. M. Hadhoud, "Evaluating the effects of symmetric cryptography algorithms on power consumption for different data types.," *International Journal Network Security*, vol. 11, no. 2, pp. 78–87, 2010.

[34] K. Naik and D. S. L. Wei, "Software implementation strategies for power-conscious systems," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 291–305, 2001.

[35] C. Panait and D. Dragomir, "Measuring the performance and energy consumption of aes in wireless sensor networks," in *Federated Conference on Computer Science and Information Systems (FedCSIS'15)*, pp. 1261–1266, 2015.

[36] T. S. Parker and L. O. Chua, "Chaos: A tutorial for engineers," *Proceedings of the IEEE*, vol. 75, no. 8, pp. 982–1008, 1987.

[37] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and blowfish," *Procedia Computer Science*, vol. 78, pp. 617–624, 2016.

[38] V. Ponnuramu and L. Tamilselvan, "Encryption for massive data storage in cloud," in *Computational Intelligence in Data Mining-Volume 2*, pp. 27–37, 2015.

[39] P. Ratha, D. Swain, B. Paikaray, and S. Sahoo, "An optimized encryption technique using an arbitrary matrix with probabilistic encryption," *Procedia Computer Science*, vol. 57, pp. 1235–1241, 2015.

[40] A. Sachdev and Mohit Bhansali, "Enhancing cloud computing security using aes algorithm," *International Journal of Computer Applications*, vol. 67, no. 9, 2013.

[41] N. Sengupta and R. Chinnasamy, "Contriving hybrid descast algorithm for cloud security," *Procedia Computer Science*, vol. 54, pp. 47–56, 2015.

[42] S. K. S. ShaluMalla, "A new security framework for cloud data," *Procedia Computer Science*, vol. 143, pp. 765–775, 2018.

[43] A. Sinha and A. P. Chandrakasan, "Jouletrack-a web based tool for software energy profiling," in *Proceedings of the 38th Design Automation Conference*, pp. 220–225, 2001.

[44] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 2017. ISBN 13: 978-0134444284.

[45] M. Sulochana and O. Dubey, "Preserving data confidentiality using multi-cloud architecture," *Procedia Computer Science*, vol. 50, pp. 357–362, 2015.

[46] M. Sulochana and O. Dubey, "Preserving data confidentiality using multi-cloud architecture," *Procedia Computer Science*, vol. 50, pp. 357–362, 2015.

[47] G. D. Sutter, J. P. Deschamps, and J. L. Imaña, "Modular multiplication and exponentiation architectures for fast rsa cryptosystem based on digit serial computation," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 7, pp. 3101–3109, 2010.

[48] A. A. Taha, D. S. A. Elminaam, and K. M. Hosny, "An improved security schema for mobile cloud computing using hybrid cryptographic algorithms," *Far East Journal of Electronics and Communications*, vol. 18, no. 4, 2018.

[49] J. Thakur and N. Kumar, "DES, AES and blowfish: Symmetric key cryptography algorithms simulation based performance analysis," *International Journal of Emerging Technology and Advanced Engineering*, vol. 1, no. 2, pp. 6–12, 2011.

[50] W. Voorsluys, J. Broberg, and R. Buyya, "Introduction to cloud computing," *Cloud Computing: Principles and Paradigms*, pp. 1–41, 2011.

[51] L. Xiong, Z. Xu, and Y. Xu, "A secure re-encryption scheme for data services in a cloud computing environment," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 4573–4585, 2015.

[52] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[53] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

Biography

Diaa Salama Abdul-Minaam was born on November 23, 1982, in KafrSakr, Sharkia, Egypt. He received the B.S from Faculty of Computers and Informatics, Zagazig University, Egypt in 2004 with grade very good

with honor, and obtains the master degree in information system from the faculty of computers and information, menufia university, Egypt in 2009 specializing in Cryptography and network security. He obtained his Ph.D. degree in information system from the faculty of computers and information, menufia university, Egypt in 2015. He is currently a Assistance Professor in Information systems department, Faculty of Computers and Information, Benha University, Egypt since 2011. He has worked on a number of research topics..Diaa has contributed more than 40+ technical papers in the areas of wireless networks, wireless network security, Information security and Internet applications, Cloud Computing, Mobile Cloud Computing, Internet of Things, and Machine learning in international journals, international conferences, local journals and local conferences. He majors in Cryptography, Network Security, IoT, Big Data, Cloud Computing, deep learning. (Mobile: +201019511000 ; E-mail: *ds$_d$esert@yahoo.com*)

Mostafa Abdullah Ibrahim received the B.S from Faculty of Computers and Informatics, Benha University, Egypt grade very good with honor, and register the master degree in information system from the faculty of computers and information, Benha university, Egypt.he is specializing in Cryptography and network security.

Elsayed Badr is an Associate professor of computer science at Benha Faculty of Computers & Informatics; Benha University in Egypt. He received his Ph.D. degree in Parallel Algorithms (mainly in parallel graph algorithms) in 2006 from the University of Macedonia; Greece. Dr. Badr holds a Certificate of Quality Assurance from the university of Benha, Egypt and M.Sc. in graph theory and graph algorithms applications, B.Sc. in Mathematics from Benha faculty of science in Egypt. In addition to over 8 years of teaching and academic experiences In Egypt and Greece, Dr. Badr has accumulated broad practical experiences and developed a solid set of skills in algorithms, Fuzzy theory, graph labeling, Wireless Networks, Distributed System, Parallel Programming and linear programming.

Low-Computation-Cost Data Hiding Scheme Based on Turtle Shell

Yu Chen[1], Jiang-Yi Lin[2,3], Chin-Chen Chang[3], and Yu-Chen Hu[4]
(Corresponding author: Chin-Chen Chang)

School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China[1]
33 Xuefu South Road, Fuzhou 350118, China
Department of Computer Science, Xiamen University of Technology, Xiamen 361024, China[2]
600 Ligong Road, Xiamen 361024, China
Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan[3]
100 Wenhua Road,Taichung 40724, Taiwan
Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan[4]
200, Sec. 7, Taiwan Boulevard, Shalu Dist., Taichung 43301, Taiwan
(Email: alan3c@gmail.com)
(Received May 16, 2018; Revised and Accepted Nov. 2, 2018; First Online July 16, 2019)

Abstract

Data-hiding technology is to study how to embed secret data into digital media such as images, audio, and video. Chang *et al.* adopted a novel turtle shell-based reference matrix to hide secret data, which resulted in better visual effects and higher embedding capacities. By changing the range of searching for elements, our proposed scheme improves the data hiding scheme of Chang *et al.* in terms of computational complexity and image quality. Experimental results verify that the proposed scheme improves the image quality of the stego images, accelerates the speed of the embedding operations, and maintains the same hiding capacity as the comparative method.

Keywords: Data Hiding; Exploiting Modification Direction; Turtle Shell

1 Introduction

Data hiding mainly studies how to hide secret data in public digital media. Usually, a certain method is designed to embed secret data into digital carriers, such as texts, audios, images, and videos. Among these digital media, digital images are used extensively as the cover medium for data hiding. Hiding secret data into the cover image makes the steganographic image slightly different from the original image, so that it will not attract attention when it is transmitted through a public network. This is one of the objectives of the data hiding scheme, and another objective is to increase the embedding capacity.

Many researchers have proposed various data hiding schemes [1, 2, 5, 10, 13, 16, 20, 21]. The aim of some of these schemes is to provide a good quality image, some focus on achieving high embedding capacity, and others focus on providing low computational cost. Chan *et al.* (2004) designed a data hiding scheme based on the simple least-significant-bit (LSB) substitution technique [1]. In their scheme, the simple LSB substitution method was for the initial hiding, and, then, the pixel values of the stego-image were modified appropriately according to the embedding error of the stego image and the original cover image. Thus, the optimization adjustment of the pixels was made, and the quality of the image was improved. Mielikainen (2006) presented a novel data hiding method named the LSB matching revisited scheme [16]. In his proposed scheme, two secret bits were embedded in a pair of cover pixels by modifying their directions. However, there is a weakness in this scheme in that exploitation is incomplete. Zhang and Wang proposed a novel steganographic method that they called exploiting modification directions (EMD) [24] in digital images, which overcame the weakness of Mielikainen's method. In Zhang *et al.*'s scheme, different embedded secret digits were represented by modifications in different directions. And each $(2k+1)$-based secret digit could be embedded by k cover pixels.

Chang and others proposed a novel information hiding method named Sudoku-S [3]. In their method, a reference matrix was generated by using a certain Sudoku solution. According to the reference matrix, each pixel pair can carry a 9-ary secret digit. Also, using the Sudoku solution, Hong and others presented an improved data hiding scheme named Sudoku-SR [9], which eliminated the shortcomings of the Sudoku-S. They proposed a search algorithm based on the nearest distance to determine where the secret digit was located, which further reduced the distortion of the image compared to the scheme of Chang *et*

al. Kim *et al.* introduced two new data hiding methods, EMD-2 and 2-EMD [12]. Compared to EMD proposed by Zhang and Wang, EMD-2 and 2-EMD improve the embedding rate and easily can be extended to EMD-K and K-EMD.

In recent years, some researchers have proposed some effective data hiding schemes [7, 8, 11, 17, 23]. Yang *et al.* proposed a scheme for embedding data using pixel-value differencing (PVD) [23]. In their scheme, the secret data were embedded by changing the value of the difference between two pairs of pixels instead of one pair of pixels. Their scheme increased the embedding capacity by using the more flexible method of searching the edge area. Chen proposed a PVD-based data hiding method that could embed secret information with a variable number of bits [7]. In his method, how many bits of secret information a pair of pixels could embed is determined by the complexity of the pixels in the area. Some reversible data hiding schemes have also been proposed [6, 15, 18, 19].

In addition, Chang and others proposed a new turtle shell-based data hiding scheme (TDH) [4]. In their scheme, the octal digits valued from 0 to 7 are arranged aptly in each hexagonal area in a constructed reference matrix, that is, in a turtle shell. In the TDH scheme, each cover pixel pair can be used to embed three bits of secret data, and the embedding capacity is improved compared to some previous schemes [2, 10].

The novelty of the turtle shell-based matrix has attracted some scholars to use it to conduct more research on data hiding. Liu *et al.* [14] improved Chang *et al.*'s scheme [4] by improving the hiding capacity. They used a positional relationship between the elements and the turtle shells to create a location table, which enabled each pixel pair to embed four bits. Xie *et al.* proposed a two-layer turtle shell-based data hiding scheme [22] in 2018. In their proposed scheme, the turtle shell-based reference matrix was considered as a layer. And different types of relationships were defined between the elements and the number of turtle shells involved, which constituted another layer, the type matrix. The proposed two-layer scheme can represent more cases than when only the turtle shell matrix is used. In this scheme, up to five bits of secret data can be embedded in each pixel pair. The above two schemes [4, 14] provided better embedding capacity than Chang *et al.*'s scheme [4], but their performances on the quality of the stego image and search time were not as good.

Inspired by Chang *et al.*'s scheme [4], we propose an improved turtle shell-based data hiding scheme that reduces the time required to generate the stego image and increases its image quality. In our scheme, first, a reference matrix is built and its internal elements are arranged in the form of turtle shells, which is the same as arrangement design in the TDH scheme. Then, when searching for the secret digit in the reference matrix, a 3×3 block is simply used as a search area instead of dealing with many turtle shell-related rules as in the TDH scheme.

The following content of this paper is arranged as follows. The TDH scheme proposed by Chang *et al.* is reviewed in Section 2. Section 3 specifies the improved scheme we proposed. Section 4 provides our experimental results, and Section 5 presents our conclusions.

2 Review of Chang and others' Scheme

In 2014, Chang *et al.* proposed the turtle shell-based data hiding scheme (TDH) [4]. In their scheme, a hexagonal area is named the turtle shell, and its range has exactly eight points, which can be used to represent the numbers 0 to 7. The reference matrix, R, is composed of many such turtle shells. Figure 1 shows the data distribution of a part of R. In R, the value of the adjacent elements in the horizontal direction increases by 1 in order, and the value of the increase between the adjacent elements in the longitudinal direction is alternating 2 and 3. Such turtle shells are arranged continuously until a reference matrix is obtained.

To generate the same reference matrix for embedding and extracting data, the element with the coordinates of (0,0) is set to 0. The other elements are generated by using the steps mentioned above. The resultant reference matrix, R, is shown in Figure 1, where the identifier p_i represents a selected pixel, the identifier p_{i+1} represents the pixel adjacent to the selected pixel, and the values of the horizontal and vertical coordinates from 0 to 255 indicate the gray-scale pixel values.

2.1 Data Hiding Procedure of TDH Scheme

Assume that a cover image I has a size $W \times H$. A cover pixel pair (p_i, p_{i+1}) will be mapped to the position (p_i, p_{i+1}) of R, where i=1, 3, ..., $(W \times H)-1$. $R(p_i, p_{i+1})$ represents the element at (p_i, p_{i+1}) in R. Chang *et al.* classified all the elements of R into normal elements and special elements. The elements within the turtle shell are classified as normal elements, and the remaining elements are classified as special elements. The normal elements are divided further into back elements and edge elements. In their proposed scheme, there are three cases to deal with for different categories of elements. Let S be the set of the area that contains the secret digit to be embedded. The specific processing cases are as follows:

Case **1:** If $R(p_i, p_{i+1})$ is a back element, S is the turtle shell where $R(p_i, p_{i+1})$ is located.

Case **2:** If $R(p_i, p_{i+1})$ is an edge element, there is at least one turtle shell that contains $R(p_i, p_{i+1})$, and S is the set of these turtle shells.

Case **3:** If $R(p_i, p_{i+1})$ is a special element, S is a set of all 3×3 blocks that contain $R(p_i, p_{i+1})$.

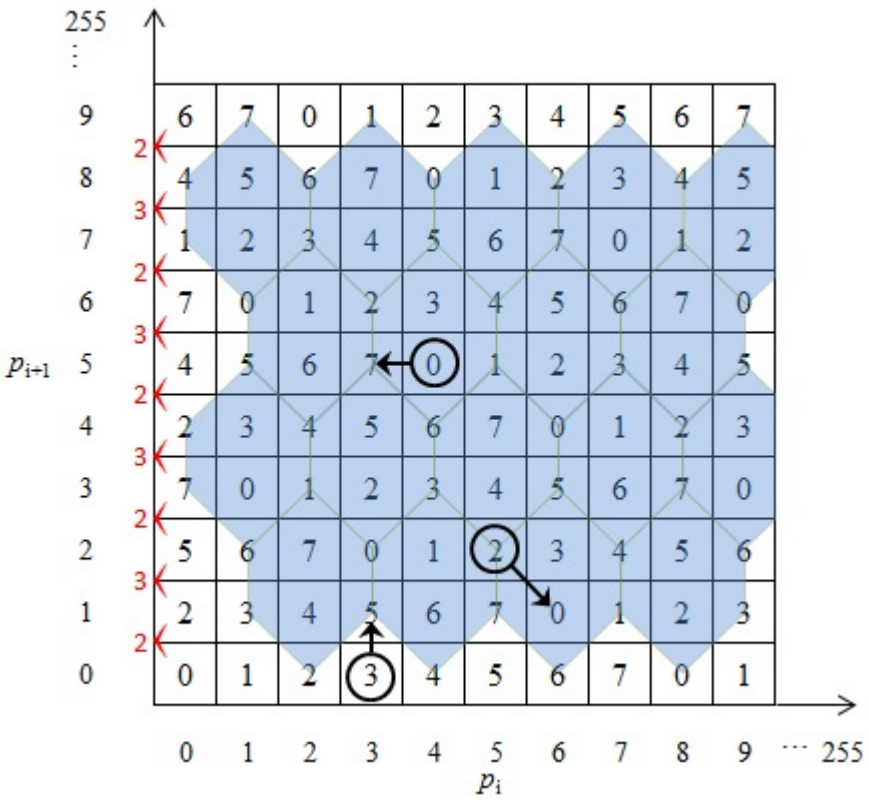


Figure 1: Example of the turtle shell-based reference matrix

Let d be the secret digit to be embedded. Because of the structural characteristics of the turtle shell-based matrix, in *Case* 2 and *Case* 3, there may be, at most, three turtle shells in S that contain d. Among all of the candidate elements included in S, the element that is the shortest distance from $R(p_i, p_{i+1})$ is selected and set as $R(p'_i, p'_{i+1})$. Then, the cover pixel pair (p_i, p_{i+1}) is modified to (p'_i, p'_{i+1}) to ensure the smallest distortion while also embedding d. After all of the pixel pairs have been processed, the stego image, I', is generated. Next is an example of embedding secret digits using the TDH scheme.

Example 1.

Assume that a binary secret data stream SD_2 is denoted as $SD_2 = (111000101)_2$, and the three stego pixel pairs used to embed secret data are (4, 5), (5, 2), and (3, 0). First, SD_2 is converted to an octal stream $SD_8 = (705)_8$. Then, each stego pixel pair is separately embedded with one digit in SD_8. The detailed embedding process is as follows. Figure 1 shows the relevant flags for the secret digits, pixel pairs, and embedding results.

1) Embed digit $(7)_8$ into the pixel pair (4, 5)

   Mapping the pixel pair (4, 5) to R, the corresponding element, $R(4, 5)$, is a back element. It is *Case* 1, and the only candidate element for digit $(7)_8$ is $R(3, 5)$, so the cover pixel pair (4, 5) is replaced by (3, 5) in I' to embed $(7)_8$.

2) Embed digit $(0)_8$ into the pixel pair (5, 2)

   Mapping the pixel pair (5, 2) to R, the corresponding $R(5, 2)$ is an edge element. It is *Case* 2, and there are three candidate turtle shells that involve $R(5, 2)$. The secret digits $(0)_8$ in the three candidate turtle shells are located at (3, 2), (6, 1), and (6, 4) in R, respectively. The squared distances between the above three elements and $R(5, 2)$ are 4, 2 and 5, respectively, where the value 2 is the smallest. Thus, the cover pixel pair (5, 2) is replaced by (6, 1) in I' to embed $(0)_8$.

3) Embed digit $(5)_8$ into the pixel pair (3, 0)

   Mapping the pixel pair (3, 0) to R, the corresponding $R(3, 0)$ is a special element. It is *Case* 3, and there are three candidate 3×3 blocks that involve $R(3, 0)$, and these three blocks contain two secret digits $(5)_8$, located at (3, 1) and (5, 0) in R, respectively. The squared distances between the above two elements and $R(3, 0)$ are 1 and 4, respectively, where the value 1 is the smallest. So, the cover pixel pair (3, 0) is replaced to (3, 1) in I' to embed $(5)_8$.

2.2 Extracting Procedure of TDH Scheme

In the TDH scheme proposed by Chang *et al.*, the reference matrix R used in the data embedding procedure also

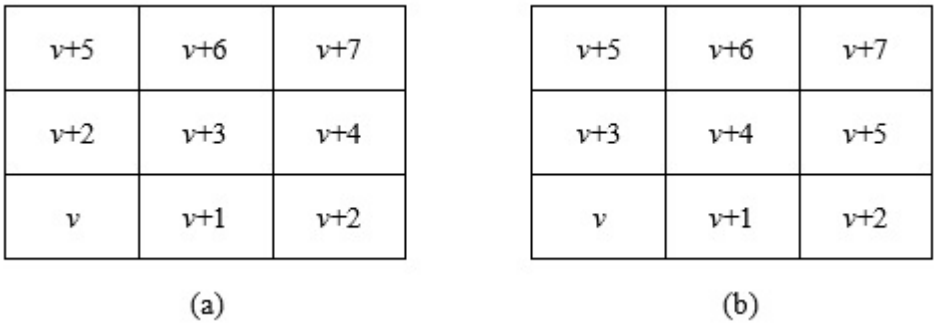


Figure 2: Two digital distributions in the 3×3 blocks

is used for the extraction of the embedded secret data. If we assume that (p'_i, p'_{i+1}) is a pixel pair of the stego image, it clearly can be mapped to $R(p'_i, p'_{i+1})$ in R. The element $R(p'_i, p'_{i+1})$ is just the embedded secret digit. The embedded secret data is obtained exactly from all of the extracted secret digits.

3 Proposed Secret Image Sharing Scheme

After studying the TDH scheme [4] proposed by the Chang *et al.*, we propose an improved turtle shell-based scheme with better image quality and faster data embedding process for its shortcomings. First, a reference matrix is constructed by the same process used in Chang *et al.*'s scheme. Both schemes are constructed based on hexagonal turtle shells. Then, each pixel pair of the cover image is used to carry three bits through the reference matrix to obtain the stego image. Unlike the TDH scheme, the method of locating the secret digit in the reference matrix of our scheme is simple and efficient, and it improves the quality of the stego image.

3.1 Secret Data Embedding

Let I be the cover image of size $M \times N$ and E be the binary secret data stream. A pixel pair of I is represented as (p_i, p_{i+1}), where $i = 1, 3, \ldots, M \times N - 1$. The reference matrix R used in our proposed scheme is shown in Figure 1. A pixel pair, (p_i, p_{i+1}), is simultaneously used as a coordinate in R corresponding to an element $R(p_i, p_{i+1})$. For example, in Figure 1, the pixel pair (2, 3) is mapped to the element $R(2, 3)$, and its value is 1. The detailed process steps for embedding secret data into each pixel pairs are shown below:

Step 1. Sequentially, read a 3-bit secret data from E and convert it to an octal secret digit n_j, where $n_j \in [0, 7]$.

Step 2. Read a pixel pair (p_i, p_{i+1}) of I and map it to a 3×3 block, which contains $R(p_i, p_{i+1})$. There are two cases to deal with.

Case 1: If the coordinate (p_i, p_{i+1}) can be the central point of a 3×3 block, then the block is set to the candidate block B.

Case 2: If the coordinate (p_i, p_{i+1}) cannot be the central point of a 3×3 block, then it will be subdivided into two cases.

Case 2.1: If $(0 < p_i < 255, p_{i+1} = 0)$ or $(0 < p_i < 255, p_{i+1} = 255)$ or $(p_i = 0, 0 < p_{i+1} < 255)$ or $(p_i = 255, 0 < p_{i+1} < 255)$, the 3×3 block whose center point coordinate of one of its edges is (p_i, p_{i+1}) is the candidate block B.

Case 2.2: If $(p_i = 0, p_{i+1} = 0)$ or $(p_i = 0, p_{i+1} = 255)$ or $(p_i = 255, p_{i+1} = 0)$ or $(p_i = 255, p_{i+1} = 255)$, the 3×3 block containing coordinate (p_i, p_{i+1}) is the candidate block B.

Step 3. Search for the secret digit, n_j, in B. If the element $R(p'_i, p'_{i+1})$ equal to n_j, change the cover pixel pair (p_i, p_{i+1}) to (p'_i, p'_{i+1}), which is a pixel pair of the stego image I'.

Step 4. Repeat Steps 1-3 until all pixel pairs of I are processed.

Step 5. Output I'.

In our scheme, any secret digit n_j can be found in the 3×3 block B. Our proof is given as below. Let v be the digit in the lower left corner of B. According to the construction of the turtle shell we are using, in general, the form of the other digits will be arranged as shown in Figures 2(a) and 2(b). Since the range of n_j is from 0 to 7, the numbers from v to $v + 7$ are to be executed by module 8 when they are greater than 7. Therefore, the numbers represented from v to $v + 7$ certainly contain eight numbers from 0 to 7. After the above process is completed, a stego image I' is produced.

Example 2. *Assume that the secret data in binary form is* $(101000110)_2$*, it can be converted to octal stream* $(506)_8$*. And assume that (3, 5), (5, 0), and (0, 0) are the three cover pixel pairs that will be used to embed the three octal digits. The following processing steps show the processes of embedding three octal digits into the pixel pairs, and the embedded results are shown in Figure 3.*

1) Embed $(5)_8$ into pixel pair (3, 5)

   The pixel pair (3, 5) is mapped to $R(3,5)$ and a 3×3 block B centered on $R(3,5)$ is determined. It is *Case* 1, and the octal digit $(5)_8$ in B is found at (3, 4). Then, the pixel pair (3, 5) in I is changed to (3, 4) in I' for embedding $(5)_8$.

2) Embed $(0)_8$ into pixel pair (5, 0)

   The pixel pair (5, 0) is mapped to $R(5,0)$, but any 3×3 block centered on $R(5,0)$ cannot be found. The element $R(5,0)$ is at the boundary of R. It is *Case* 2.1, and the corresponding 3×3 block B is marked in Figure 3. The secret digit $(0)_8$ in B is found at (6, 1). Therefore, the pixel pair (5, 0) in I is changed to (6, 1) in I' for embedding $(0)_8$.

3) Embed $(6)_8$ into pixel pair (0, 0)

   The pixel pair (0, 0) is mapped to $R(0,0)$, which is located at the corner of R. It is *Case* 2.2, and the corresponding 3×3 block B also is marked in Figure 3. The secret digit $(6)_8$ in B is found at (1, 2). Thus, the pixel pair (0, 0) in I is changed to (1, 2) in I' for embedding $(6)_8$.

3.2 Extraction of Secret Data

When a stego image I' of size $M\times N$ is received, the proposed scheme can accurately extract the secret data embedded therein. First, the receiver constructs a reference matrix, R, used in the secret data embedding process. Second, all pixel pairs in I' are read sequentially and mapped as coordinates to R. Let (p'_i, p'_{i+1}) be a pixel pair in I', and $R(p'_i, p'_{i+1})$ is the corresponding mapping element in R, where $i = 1, 3, \ldots, M\times N-1$. Then, the element $R(p'_i, p'_{i+1})$ is the octal secret digit embedded in the pixel pair (p'_i, p'_{i+1}). The extracted secret digits will be sequentially concatenated together to form an octal stream. After all the pixel pairs in I' have been processed in this way, a complete octal secret stream is obtained. The octal secret stream is then converted to a binary stream, and the receiver successfully extracts the secret data from I'.

Example 3. *Assume that the pixel pairs (3, 4), (6, 1), and (1, 2) are consecutive pixel pairs in the stego image I' generated in Example 2. We now extract the secret data embedded in them. In R, the elements corresponding to coordinates (3, 4), (6, 1), and (1, 2) are $R(3,4) = (5)_8$, $R(6,1) = (0)_8$, and $R(1,2) = (6)_8$, respectively. The obtained octal digits are connected one by one to form an octal stream $(506)_8$. Finally, the octal stream is converted to a binary stream $(101000110)_2$, which is the embedded secret data.*

4 Experimental Results

Some experiments were conducted on some test images to illustrate the correctness of the proposed scheme. Our experiments were conducted in MATLAB 8.0 software in a personal computer configured Intel(R) Core (TM) i7-3770 @ 3.40 GHZ and 8 GB of memory, and the operating system was installed is Windows 10 Education 64 bits.

4.1 Experiment Design

Eight original grayscale images, namely, Airplane, Baboon, Boat, House, Elaine, Lena, Man, and Peppers, were tested as cover images in our experiment. All test images had 512×512 pixels. The secret data used in the experiment consisted of a binary stream formed by randomly-generated, binary bits. The original grayscale test images are shown in Figure 4.

In the proposed scheme, the peak signal-to-noise ratio (*PSNR*) score is used to evaluate the quality of the stego image, which is defined as Equation (1):

$$PSNR = 10\log_{10}\frac{255^2}{MSE}, \tag{1}$$

where *MSE* represents the mean square error of the stego image and the cover image, which is defined in Equation (2).

$$MSE = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{ij} - Y_{ij}). \tag{2}$$

where M and N represent the height and width of the image, respectively, and X_{ij} and Y_{ij} are the values of the pixels of the stego image and the cover image, respectively. Equations (1) and (2) indicate that there is an inverse relation between *PSNR* and *MSE*, *i.e.*, the lower the *MSE* value is, the higher the *PSNR* value becomes. A higher *PSNR* value indicates that there is a smaller difference between the two images.

Another performance measurement we used to evaluate the data hiding capacity is the embedding capacity (*EC*), which represents the number of binary bits that can be embedded in a cover image. The *EC* of our scheme also is compared with the *EC* of TDH scheme, and Table 1 shows the results. Table 1 indicates that the *EC* of the proposed scheme had the same value as that of the TDH scheme, reaching 1.5 bpp. Average image qualities of 49.72 dB and 50.14 dB are achieved by the TDH scheme and the proposed scheme, respectively. In addition, the average execution time of the TDH scheme is 0.67 second, and for the proposed scheme, it is 0.58 second. Obviously, the proposed scheme gains better quality of stego images than the TDH scheme proposed by Chang *et al.*, and its computational cost for embedding data is less than that of Chang *et al.*'s TDH scheme.

To understand the visual qualities of the stego images of the TDH scheme and the proposed scheme, two sets of stego images produced by the two schemes are provided in Figures 5 and 6, respectively. Compared to the cover images shown in Figure 4, it is difficult to distinguish the difference between the stego images and the original

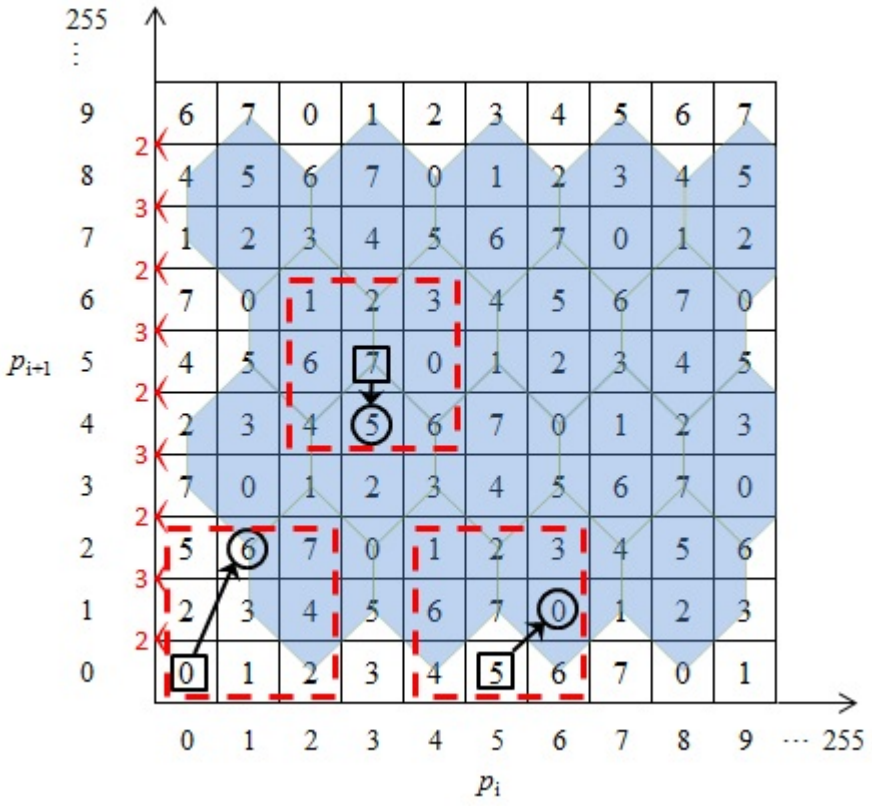


Figure 3: Examples of embedding process based on turtle shells

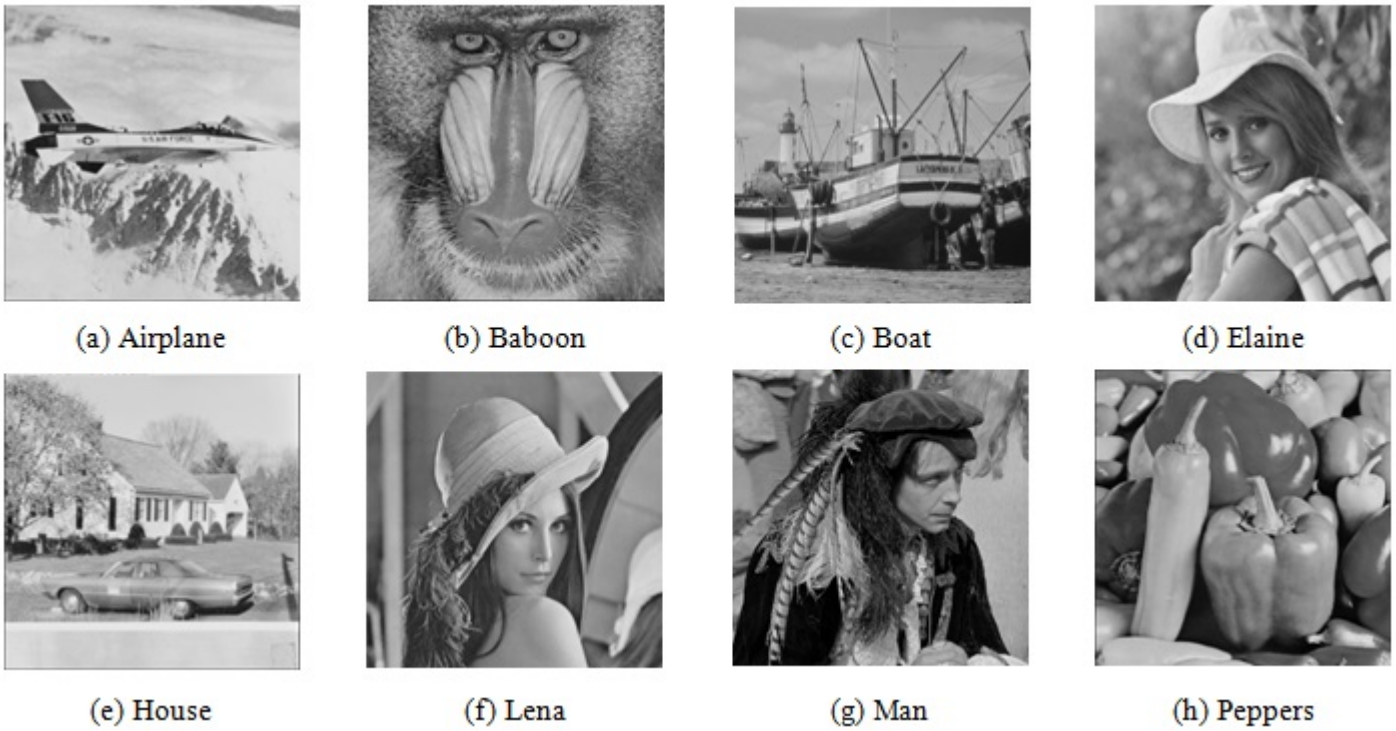


Figure 4: Grayscale test images

Figure 5: Stego images of Chang *et al.*'s scheme

Figure 6: Stego images of the proposed scheme

Table 1: Central point bits of the binary block and its corresponding shadow blocks

Cover Images	Chang *et al.*'s scheme (TDH)			Proposed scheme		
	PSNR	*EC*	Running Time	*PSNR*	*EC*	Running Time
Airplane	49.75	1.5	0.64	50.17	1.5	0.60
Baboon	49.75	1.5	0.75	50.16	1.5	0.60
Boat	49.75	1.5	0.67	50.16	1.5	0.58
Elaine	49.75	1.5	0.67	50.17	1.5	0.58
House	49.76	1.5	0.69	50.16	1.5	0.58
Lena	49.75	1.5	0.63	50.17	1.5	0.59
Peppers	49.76	1.5	0.68	50.18	1.5	0.58
Man	49.48	1.5	0.68	49.93	1.5	0.56
Average	49.72	1.5	0.67	50.14	1.5	0.58

images. In other words, the visual qualities of the stego images of the two schemes are very good.

4.2 Analysis of the Experimental Results

In the data embedding process, the area for searching for the secret digit in our scheme is in a certain 3×3 block, while the search scope of Chang *et al.*'s scheme is in turtle shells or some 3×3 blocks. Due to the structural features of the turtle shell, there will be some cases in which the secret digit within the turtle is far from the element that corresponds to a cover pixel pair. For example, the distance between $R(3,3)$ and the secret digit 7 inside the turtle shell reaches 2. And in our scheme, the secret digit 7 at (2, 2) is selected, and the distance is , which is less than 2. The cover pixel pair is replaced with the coordinate value of the nearest element to embed the secret digit, resulting in a small change in the cover pixel value. Thus, the *MSE* value computed by Eq. (2) is small, and the *PSNR* value is high. Therefore, the proposed scheme outperforms the TDH scheme proposed by Chang *et al.* in the quality of the stego images.

In terms of running time, the range of looking for the secret digit in our scheme is only a 3×3 block, while this search range in Chang *et al.*'s scheme is in a set that may involve more than one turtle shell or one 3×3 block, so the proposed scheme consumes less computational cost.

The above analysis shows that the proposed scheme outperforms the TDH scheme proposed by Chang *et al.* in terms of implementation efficiency and image quality. The experimental results validated this analysis.

5 Conclusions

An improved turtle shell-based data hiding scheme is proposed in this paper. The reference matrix used in the proposed scheme is the same as that in the TDH scheme. But, in the proposed scheme, a 3×3 block instead of a hexagon is used to perform the search for the element of the corresponding secret digit in the reference matrix. This improvement reduces the distortion of the embedded image and speeds up the search for elements for data embedding. The experimental results show that our scheme improves the visual quality and processing speed compared to the TDH scheme, in which the image quality is improved by an of 0.42 dB and the execution time is reduced by an average of 0.09 seconds.

References

[1] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, pp.469–474, 2003.

[2] C. S. Chan, C. C. Chang, and Y. C. Hu, "A color image hiding scheme using image differencing," *Optical Engineering*, vol. 44, no. 1, pp. 1–9, 2005.

[3] C. C. Chang, Y. C. Chou, and T. D. Kieu, "An information hiding scheme using sudoku," in *Proceedings of The Third International Conference on Innovative Computing Information and Control (ICICIC'08)*, pp. 17–21, June 2008.

[4] C. C. Chang, Y. J. Liu, and T. S. Nguyen, "A novel turtle shell based scheme for data hiding," in *Proceedings of The Tenth International Conference Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP'14)*, pp. 89–93, Aug. 2014.

[5] C. C. Chang, Y. H. Yu, and Y. C. Hu, "Hiding secret data in images via predictive coding," *Pattern Recognition*, vol. 38, no. 5, pp. 691–705, 2005.

[6] I. C. Chang, Y. C. Hu, W. L. Chen, and C. C. Lo, "High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding," *Signal Processing*, vol. 108, pp. 376–388, 2015.

[7] J. Chen, "A pvd-based data hiding method with histogram preserving using pixel pair matching," *Signal Processing: Image Communication*, vol. 29, no. 3, pp .375–384, 2014.

[8] W. Hong and T. S. Chen, "A novel data embedding method using adaptive pixel pair matching," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 176–184, 2012.

[9] W. Hong, T. S. Chen, and C. W. Shiu, "A minimal euclidean distance searching technique for sudoku steganography," in *Proceedings of International Symposium on Information Science and Engineering*, pp. 515–518, Dec. 2008.
[10] Y. C. Hu, "High capacity image hiding scheme based on vector quantization," *Pattern Recognition*, vol. 39, no. 9, pp. 1715–1724, 2006.
[11] T. D. Kieu and C. C. Chang, "A steganographic scheme by fully exploiting modification directions," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10648–10657, 2011.
[12] H. J. Kim, C. Kim, Y. Choi, S. Wang, and X. Zhang, "Dual-image-based reversible data hiding method using center folding strategy," *Signal Processing*, vol. 60, no. 2, pp. 319–325, 2010.
[13] M. H. Lin, Y. C. Hu, and C. C. Chang, "Both color and gray scale secret image hiding in a color image," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 6, pp. 697–713, 2002.
[14] Y. Liu, C. C. Chang, and T. S. Nguyen, "High capacity turtle shell-based data hiding," *IET Image Processing*, vol. 10, no. 2, 130–137, 2016.
[15] C. C. Lo, Y. C. Hu, W. L. Chen, and C. M. Wu, "Reversible data hiding scheme for btc-compressed images based on histogram shifting," *International Journal of Security and Its Applications*, vol. 8, no. 2, pp. 301–314, 2014.
[16] J. Mielikainen, "Lsb matching revisited," *IEEE Signal Process Letters*, vol. 13, no. 5, pp. 285–287, 2006.
[17] C. Qin, C. C. Chang, Y. H. Huang, and L. T. Liao, "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1109–1118, 2013.
[18] C. Qin and Y. C. Hu, "Reversible data hiding in VQ index table with lossless coding and adaptive switching mechanism," *Signal Processing*, vol. 129, pp. 48–55, 2016.
[19] P. Y. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, no. 6, pp. 1129–1143, 2009.
[20] Y. C. Tseng, Y. Y. Chen, and H. K. Pan, "A secure data hiding scheme for binary images," *IEEE Transactions on Communications*, vol. 50, pp. 1227–1231, 2002.
[21] A. Westfeld, "F5: A steganographic algorithm," *Lecture Notes in Computer Science*, vol. 2137, pp. 289–302, 2001.
[22] X. Z. Xie, C. C. Lin, and C. C. Chang, "Data hiding based on a two-layer turtle shell matrix," *Symmetry*, vol. 10, no. 2, 2018.
[23] C. H. Yang, C. Y. Weng, H. K. Tso, and S. J. Wang, "A data hiding scheme using the varieties of pixel-value differencing in multimedia images," *Journal of Systems and Software*, vol. 84, no. 4, pp. 669–678, 2011.
[24] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Communications Letters*, vol. 10, no. 11, pp. 781–783, 2006.

Biography

Yu Chen received the B.S. degree in Computer and Application from Hunan University, Hunan, China in 1993, and M.S. degree in Software Engineering from Fuzhou University, Fujian, China, in 2006. Currently, he is an associate professor in the School of Information Science and Engineering, Fujian University of Technology(FJUT), China. His current research interests include information retrieval, data mining, and digital image processing.

Jiang-Yi Lin received the B.S. and M.S. degrees in Computer science and Technology from FuZhou Uniersity, FuJian, China, in 2005 and 2008, repectively. He is currently pursuing the Ph.D degree with the Multimedia and Secure Networking Laboratory (MSN lab), the Department of Information Engineering and Computer Science of Feng Chia University, Taichung, Taiwan. His research interests include image processing, secret sharing and steganography.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from February 2005. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And, since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression, and data structures.

Yu-Chen Hu received his PhD. degree in computer science and information engineering from the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan in 1999. Currently, Dr. Hu is a professor in the Department of Computer Science and Information Management, Providence University, Sha-Lu, Taiwan. He is a senior member of IEEE. He is also a member of Computer Vision, Graphics, and Image Processing (CVGIP), Chinese Cryptology and Information Security Association (CCISA), Computer Science and Information Management (CSIM) and

Phi Tau Phi Society of the Republic of China. He servers as the Editor-in-Chief of International Journal of Image Processing from June 2009 to May 2015. In addition, he is the managing editor of Journal of Information Assurance & Security since March 2009. He is the associated editor of Human-centric Computing and Information Sciences since Feb. 2011. He joints the editorial boards of several other journals. His research interests include digital forensics, information hiding, image and signal processing, data compression, information security, and data engineering.

A PSO-based Wavelet-core ELM for Abnormal Flow Detection

Yueyang Su[1], Jing Wan[1], and Junkai Yi[2]
(Corresponding author: Jing Wan)
College of Information Science and Technology, Beijing University of Chemical Technology[1]
North Third Ring Road 15, Chaoyang District, Beijing, China
College of Information Science and Technology, Beijing Information Science and Technology University[2]
North Fourth Ring Road 35, Chaoyang District, Beijing, China
(Email: suyy1225@163.com)
(Received July 31, 2018; Revised and Accepted Jan. 9, 2019; First Online July 16, 2019)

Abstract

Abnormal flow detection is an effective approach to discover the covert data during the transmission process of mass data. However, there exist some issues to tackle such as the high complexity of network traffic data, Low detection efficiency and low accuracy. To solve these problems, we proposes an improved wavelet-core extreme learning machine based on particle swarm optimization. First, the particle swarm optimization algorithm is applied to determine the input weights and bias thresholds of the extreme learning machine, which effectively reduces the number of hidden layer nodes. Furthermore, wavelet kernel function is proposed to be the kernel function of kernel extreme learning machine. Then the topology of the KELM can be established, and can be applied to classify the abnormal traffic. We introduce overall-accuracy and F-measure for performance measure in abnormal flow detection. To verify the effectiveness of our work, we compare the approach with the representative algorithms, and experimental results show that the improved wavelet-core extreme learning machine based on particle swarm optimization has better detection performance.

Keywords: Abnormal Flow Detection; KELM; Particle Swarm Optimization Algorithm; Wavelet Kernel Function

1 Introduction

In the distributed Internet environment, the discovery and analysis of covert data in the process of mass data transmission is a serious problem to be solved, and it is also a main guarantee for the healthy development of the virtual economy in the future. In recent years, the data transmission technology based on covert channel has developed rapidly. The covert data transmission of massive multi-modal information based on blockchain [4, 12] has been brought to our attention. Meanwhile, the security issues has been increasingly significant, the discovery and analysis of covert data has become an important requirement for new network applications.

Abnormal traffic detection is an important technology for the discovery of covert data. The detection of network abnormal flow is to analyze network flow data via statistical analysis, data mining and machine learning with the intention to discover abnormal information of network data.

Statistical analysis is an early method for anomaly detection of traffic data. First, count the number of network traffic packets, the length of packets and other characteristic information. Then, discover the characteristics rules of the traffic data. Finally, in order to detect the abnormal flow information, establish the normal behavior profile of traffic data as the standard of judgment of the traffic data to be detected. Hoang *et al.* applied the principal component analysis method and the wavelet transform to make the model, which combined with the spatio-temporal correlation of the feature matrix of traffic data [6]. They proposed an PCA-based network anomaly detection algorithm. Although this method has a good effect, it is difficult to detect during network transmission and achieve real-time abnormal traffic detection. Bhuyan *et al.* proposed a DDOS attack detection method based on the characteristics of traffic and extended entropy metric, which effectively reduced the computational complexity and detection time [3]. However, the process of establishing the traffic model is still complex and it is also difficult in the practical application. Although the statistical analysis method has a good detection effect, it is sensitive to the change of the threshold value, and at the same time, it cannot reflect the autocorrelation of the abnormal behavior in time.

Data mining is a major approaches for the detection of anomaly traffic data, which aims to establish awareness model of anomalous traffic by analyzing the mass flow

data [5]. Clustering algorithm is an important method of data mining. Unsupervised learning method can be used to classify the heterogeneous and high-dimensional massive traffic data. The density peak clustering algorithm [11] based on the assumptions as following: Within clusters, the local density of the clustering center points is the highest; Among clusters, the clustering center is away from other clusters'. The algorithm has a good effect on various data distributions, but it is sensitive to the global abnormalities in some special cases and has poor results. Ahmed *et al.* established a collective anomaly detection framework via partition clustering technology to detect DoS attacks and improve the detection accuracy [2]. However, the algorithm has limitations and works weakly in the detection of other attacks. Although the clustering algorithm can be used with unclassified sample data, the speed and accuracy is still far away from application.

The classification algorithm is a supervised machine learning method. Hua [7] applied the K-means algorithm to improve the traditional KNN and divided the process of anomaly detection into two parts: off-line preprocess and on-line classification, which improved the efficiency and classification accuracy. But the feature redundancy and dimensionality disasters is still the most serious problem of the algorithm. Ma [9] and his partners applied the Naïve Bayesian network to construct classifiers for traffic classification, which limited by the fixed assumptions. Roy [10] detected and analyzed attack behaviors via deep neural networks, which improved the efficiency and performance of anomaly detection. However, the low iteration speed is still the most serious problem and it is also easy to fall into local convergence.

The extreme learning machine is a single-hidden layer feedforward neural networks. The model randomly selects the hidden layer nodes and replaces the iterative process for adjusting parameters by analyzing to get the weight matrix between the hidden layer and the output layer. With the great learning efficiency and self-adaptive ability, many researchers have devoted to the study of improving extreme learning machines. Kumari *et al.* proposed a semi-supervised support vector machine with fuzzy c-means clustering, which greatly reduced the computational complexity and improved the classification efficiency [8].

There are many abnormal traffic detection approaches with some problems around. Some existed approaches are simple to cope with the weight of traffic statistical features, and the process with equal weight may cause the loss of information. Otherwise some methods also use the whole traffic as analysis objects, and the amount of data is enormous, which can lead to low accuracy rate and efficiency. In this paper, we propose an abnormal traffic detection approach based on Particle Swarm Optimization (PSO) and wavelet kernel Extreme Learning Machine (ELM). PSO is an algorithm of global searching optimal solutions. The algorithm can be introduced to set the optimal input weight and the bias threshold of kernel ELM, which eliminates redundant nodes of hidden layer.

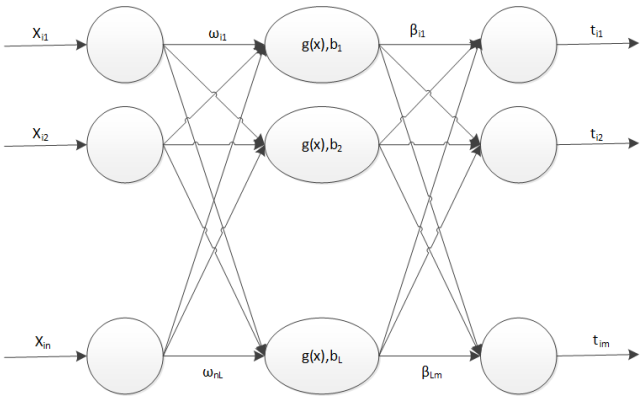


Figure 1: The extreme learning machine network model

With the application of PSO, the classification accuracy and learning efficiency is greatly improved and it is not sensitive to the number of training samples and hidden layer nodes as before. We choose wavelet kernel function as the kernel function of ELM, which improves the ability of nonlinear approximation and generalization. The experiments shows that the model we proposed has great robustness and better detection performance.

2 The PSO-based Wavelet-Core Extreme Learning Machine

2.1 Extreme Learning Machine

Extreme learning machine is a single hidden layer neural network. With random hidden layer nodes, ELM effectively reduces the training time and improves the generalization ability.

Similar to the traditional network model, the model of extreme learning machine is divided into three layers: input layer, hidden layer, and output layer. The specific structure is shown in Figure 1.

Given N sets of training data (x_i, t_i), $x_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in R^N$, $t_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T \in R^m$, The mathematical model of SLFN with L hidden layer nodes can be described as Equation (1):

$$y_j = \sum_{i=1}^{L} \beta_i g_i(x_i) = \sum_{i=1}^{L} \beta_i g(w_i x_j + b_i) = T_j, \quad (1)$$

where $g(x)$ is the activation function, β_i is the connection weight vector between the hidden layer and the output layer, w_i is the connection weight vector between the hidden layer and the input layer. b_i is the threshold of the ith node in the hidden layer. The above formula can also be simplified as Equation (2):

$$H\beta = T, \quad (2)$$

where H is the output matrix of the hidden layer node, β is the weight vector of the output layer, T is the expected

output matrix of the sample.

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_L \cdot x_1 + b_L) \\ \vdots & \dots & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_L \cdot x_N + b_L) \end{bmatrix}_{N \times L} \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}, T = \begin{bmatrix} t_1^T \\ \vdots \\ t_m^T \end{bmatrix}_{N \times m} \tag{4}$$

The ELM can be trained without adjusting the weights and bias threshold of the input layer, the output matrix of the hidden layer is only determined by the random w_i and b_i. Therefore, the training of extreme learning machines can be transformed into the process of deriving the output weights β according to the $H\beta = T$. β can be expressed as Equation (5):

$$\beta = H^{\dagger}T, \tag{5}$$

where $H^{\dagger}$ is the Moore-Penrose generalized inverse matrix.

2.2 Kernel-ELM

In order to improve the generalization ability of the extreme learning machine, a kernel function was introduced and then the kernel-ELM (KELM) was proposed.

For the traditional extreme learning machine, the output matrix of hidden layer can be expressed as Equation (6):

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} \tag{6}$$

where $h(x_i)$ can be regarded as a non-linear mapping of x_i, if the mapping is unknown, a kernel function M can be constructed instead of HH^T. According to Mercer, the kernel matrix can be defined as:

$$HH^T, m_{ij} = h(x_i)h(x_j) = k(x_i, x_j), \tag{7}$$

where $i, j \in (1, 2, \cdots, N)$

$$h(x)H^T = \begin{bmatrix} \mathrm{k}(x, x_1) \\ \vdots \\ k(x, x_N) \end{bmatrix} \tag{8}$$

where $k(x_i, x_j)$ is the kernel function. Then the output function $f(x)$ of KELM can be expressed as Equation (9):

$$f(x) = [k(x, x_1), \cdots, k(x, x_N)] \left[\frac{1}{C} + M\right]^{-1} T. \tag{9}$$

2.3 The Morlet Wavelet Function

The kernel function which satisfies the premise of Mercer's theorem can be used as the kernel function of the kernel-ELM. Linear kernel is the simplest kernel function; polynomial kernel is a kind of non-standard kernel function which is suitable for orthogonal normalized data; Gaussian kernel function is widely used in image processing and has a great anti-jamming capability of noise in data. For the complexity of network traffic data, the Morlet wavelet function is introduced to be the kernel function of the kernel-ELM in this paper, which has great classification effect in the space without training data.

In general, the wavelet basis function can be expressed as Equation (10):

$$h_{a,b}(x) = \sqrt{a}\Phi(\frac{x-b}{a}), \tag{10}$$

where $h(x)$ is the mother wavelet function, a is the scaling factor, b is the balance factor, According to the tensor product theory, any multidimensional wavelet function can be expressed as a tensor product of multiple one-dimensional wavelet functions as Equation (11):

$$h(x) = \prod_{i=1}^{n} h(x_i). \tag{11}$$

Construct the translation-invariant kernel function via Equation (10) :

$$K(x, x')=K(Kx - x') = \prod_{i=1}^{n} \Phi(\frac{x_i - x_i{}'}{a}). \tag{12}$$

For sample x, $x' \in R$, the Morlet wavelet function $h(x) = \cos(1.75x)\exp(-\frac{x^2}{2})$, construct the kernel-ELM with the corresponding wavelet kernel function. The specific formula is as Equation (13):

$$\begin{aligned} & Waveletkernel(x, x') \\ = & \prod_{i=1}^{n} [\cos(1.75(\frac{x_i - x_i{}'}{a}))\exp(-\frac{(x_i - x_i{}')^2}{2a^2})] \end{aligned} \tag{13}$$

2.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a strategy proposed by Eberhart and Kennedy to solve optimization problems inspired by the feeding behavior of birds. In the POS, the solution to each optimization problem is considered as a location in the search space, called "particles." The particle velocity determines the direction and distance of the particle's flight. It can also track its own optimal position in the iterative process and the best position of all particles in the entire particle group with their own privacy memory. As a basis, it can update the speed and location.

The individual extremum is $R_i^b(t)$, the global extremum is $R_g^b(t)$. Then the motion equation of particle is as Equation (14):

$$\begin{aligned} v_i(t+1) &= \omega v_i(t) + c_1 R_1 \left[R_i^b(t) - x_i(t)\right] \\ &\quad + c_2 R_2 \left[R_g^b(t) - x_i(t)\right], \end{aligned} \tag{14}$$

$$x_i(t+1) = x_i(t) + \phi v_i(t+1), \tag{15}$$

where $v_i(t)$ and $x_i(t)$ are the velocity and position of the ith particle at the tth iteration; c_1, c_2 are the learning factors, R_1 and R_2 are the random variable which are evenly distributed over the interval $[0, 1]$, ϕ is the contraction factor.

The pocess of PSO algorithm is as follows:

Step 1: Initialize the particle group, each particle is set with a random position and velocity;

Step 2: Evaluate the fitness of each particle;

Step 3: For each particle, compare the fitness value and its historical best position pbest, if better, update the pbest;

Step 4: For each particle, compare the fitness value with its gbest, if better, update the gbest;

Step 5: Adjust the speed and position of the particles according to (2) and (3);

Step 6: If it does not satisfied the ending condition, go to Step 5.

The ending condition of iteration depends on the specific problem. In general, it can be ended when the optimal position of the particle swarm satisfies the predetermined minimum adaptive threshold or the times of iterations reaches the maximum number.

Particle Swarm Optimization (PSO) is a global optimization algorithm via randomly searching. The cooperation mechanism between groups is introduced to reach the optimal solution. It has been widely applied to engineering optimization with the good robustness, simple operation, and free from constraints.

2.5 The Wavelet-core ELM Based On The Particle Swarm Optimization (PW-ELM)

In general, the wavelet-core extreme learning machine has good performance in abnormal flow detection, but the accuracy of the ELM is affected by many factors, such as the number of hidden layer nodes, bias threshold and so on. The number of nodes in the hidden layer has a great influence on the generalization ability and learning speed of the ELM. Too many nodes may lead to the increasing of the network complexity and overfitting. The value of the bias threshold and connection weight can also affect the training process of the ELM because of the direct relationship with the output weight. When they are both zero, some nodes of hidden layer will be invalid. In order to improve the learning process of the ELM and optimize the connection weights and thresholds, the particle swarm optimization algorithm is introduced.

In the wavelet-core ELM based on the particle swarm optimization, we abstract the input weights and bias thresholds into particles in the particle swarm, and take the root mean square error of the particles as the fitness function. The particle length(L) is determined by the number of hidden layer nodes (k) and the number of input layer nodes (m), as Equation (16):

$$L = k(m + 1). \tag{16}$$

The process of algorithm is as follows:

Step 1: Select the training data, set the input vector and the expected output vector;

Step 2: Set the topological structure of the wavelet-core ELM, initialize the number of neurons in the input layer, hidden layer, and output layer;

Step 3: Create a particle swarm based on the input vector and bias threshold of the ELM. Set the initial speed, position of the particles, and the optimize space;

Step 4: Set a suitable fitness function, the root mean square error of the particle is choosed for our model . Set the maximum number of iterations = 600, learning factor $c_1 = c_2 = 1.5$, population size $M = 25$ and the particle dimension D;

Step 5: Calculate the fitness of the particle based on the training set, find the individual extreme value and the global extreme value;

Step 6: Update the position and speed of the particles;

Step 7: If it does not satisfied the ending condition, go to Step 5;

Step 8: Establish the wavelet-core ELM with the input weights and hidden layer bias thresholds generated by the particle swarm optimization algorithm.

3 Experimental Results And Analysis

3.1 Date Collection

The wavelet-core extreme learning machine based on particle swarm optimization has good generalization ability and classification accuracy. In order to verify the performance of the improved kernel-ELM, the KDD 99 dataset was selected as the analysis object.

The KDD 99 dataset is a competition dataset used by the International Data Mining and Knowledge Discover competition in 1999. The dataset was established based on the Intrusion Detection Evaluation Project of US Department of Defense Advanced Planning Agency (DARPA) in 1998, which collected data from the simulated military network in the Lincoln Lab. The collection of data lasted for two and a half months, including different network traffic and attack methods. The aims of competition is to detect the network intrusion and achieve the abnormal classification of network connections.

A network connection consists of a sequence of TCP packets from the beginning to the end in a certain period

of time. During this period of time, the data transfers between the original address and the destination address based on a predefined protocol. The network connection record contains a status bit to mark it normal or attack. The types of exceptions can be categorized as: Remote-to-Login Attack (R2L), Denial of Service Attack (DoS), Probing Attack (PROBING) and User-to-Root Attack (U2R).

The KDD 99 dataset is divided into two subset, the one is the training dataset which contains about 5,000,000 records, and another is the test dataset including about 2,000,000 records. The distribution of samples is shown in Table 1.

3.2 Extract Features Of Network Flow

There are various network attacks divided into 4 categories. The Denial of Service is the most common one including UDP floods, Land attacks, e-mail bombs, *etc.*; The other is Exploitable Attacks which contains Password Guessing, Trojans, Buffer Overflows, *etc.*; The information-gathering Attacks is used to obtain the useful information including Address Scanning, Port Scanning and DNS Domain Conversion; The last one is False-messaging Attacks which mainly contains DNS Cache Pollution and Fake Emails, *etc.* [1]. Different network attack methods are different in the abnormal behavior of traffic data. And it is important to select the appropriate statistics features of data flow. The number of features is also important for the classification accuracy. In general, the larger the number of feature values is, the higher the classification accuracy will be. However, when the number is too large, the overall performance of the classifier would be worse [13].

We selected 16 representative characteristics of data flow in this paper including the network service type of the target host, the number of urgent packets, the number of error segments, the connection status (normal or error), and transmission protocol, *etc.* The specific information is in Table 2.

3.3 Data Preprocessing

For the complexity of the sample data about network connection, the input data of classifier needs to be normalized to reduce the classification error and accelerate the convergence speed.

$$X = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{17}$$

where, $x_{\max}$ is the maximum in the dataset, $x_{\min}$ is the minimum.

3.4 Evaluation Function

In order to analyze the experimental results, the $Overall - accuracy$ and $F - measure$ were used to evaluate the classification performance of different methods. TP is the number of samples which is correctly classified, FP is the number of other classes' samples which are erroneously divided into this class, FN is the number of the a certain class's samples which are misclassified.

$$precision(i) = \frac{TP_i}{TP_i + FP_i} \tag{18}$$

$$recall(i) = \frac{TP_i}{TP_i + FN_i} \tag{19}$$

The $Overall - accuracy$ used in this paper is the ratio of the model's correct prediction to the total number on all the test sets, the specific formula is as Equation (20):

$$Overall - accuracy(i) = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m} TP_i + FN_i} \tag{20}$$

There are sometimes contradictions between the *precision* indicator and the *recall* indicator. In order to consider them comprehensively, $F - measure$ is introduced. It is a reconciliation measure between *recall* and *precision*. The specific formula is as Equation (21):

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{21}$$

3.5 Experimental Results

In the experiment, we selected 10% training subsets and 10% test subsets from the KDD 99 dataset. The distribution of sample is shown in Table 3.

First, the features should be numerically normalized to convert the data to the standard input data of ELM. After 10 experiments, we find the average of the 10 accuracy as the final result of the experimental accuracy.

In order to verify the performance of the wavelet-core extreme learning machine based on particle swarm optimization in anomaly detection, we selected the non-kernel ELM, Gaussian kernel ELM (Gauss-kernel ELM), and Gaussian kernel support vector machine (Gauss-kernel SVM) as contrast on the KDD 99 dataset. The parameters of each classifier are shown in Table 4.

After 10 experiments, we find the $Overall - accuracy$ and $F - measure$ for analysis, the results and average of $Overall - accuracy$ in the 10 experiments are shown in Table 5 and Table 6.

According to the Table 5 and Table 6, we can realize that the $Overall - accuracy$ of PW-ELM, Gauss-kernel SVM, Gauss-kernel ELM and ELM on the KDD 99 dataset are: 94.746%, 90.205%, 83.207% and 74.942%. The $Overall - accuracy$ of PW-ELM is obviously higher than the other algorithms, approaching 95%. The performance of Gauss-kernel SVM is higher than the ELM and Gauss-kernel ELM. In summary, the performance of the PW-ELM achieve an ideal $Overall - accuracy$ in abnormal flow detection.

The $F - measure$ of the four algorithms is shown in Figure 3. On the KDD 99 dataset, PW-ELM has

Table 1: The distribution of KDD 99 dataset

Category	**Normal**	Abnormal			
	Normal	Dos	U2R	R2L	PROBE
Training Dataset	0.1969	0.7924	0.0001	0.0022	0.0083
Test Dataset	0.1975	0.7490	0.0007	0.0528	0.0136

Table 2: The representative characteristics of data flow

characteristics	**description**	**amount**
Network Connection	network service type of the target host, the number of expedited packets,the number of error segments, connection status (normal or error), transmission protocol	5
Package	the number of packages	1
Bytes	the bytes of data from the source host to the destination host, The bytes of data from the target host to the source host	2
Packet size	the average, maximum, minimum, standard deviation of packet size	4
Connection time	the average, maximum, minimum, standard deviation of connection time	4
total		16

Table 3: The distribution of training subsets and test subsets

Category	**Normal**	Abnormal			
	Normal	Dos	U2R	R2L	PROBE
Training Dataset	97278	391458	52	1126	4107
Test Dataset	60593	229853	228	16189	4166

Table 4: The parameters of each classifier

Algorithm / Parameters	**ELM**	**Gauss-kernel ELM**	**Gauss-kernel SVM**	**PW-ELM**
Penalty factor(C)	1000	1000	1000	1000
Kernel parameters(a)		2.5	1.8	2.0
The number of hidden layer nodes(L)	800			

Table 5: The $Overall-accuracy$ of classifiers

$Overall-accuracy$	**1**	**2**	**3**	**4**	**5**
ELM	74.273	74.611	75.902	75.059	74.385
Gauss-kernel ELM	81.925	82.328	83.667	83.516	82.739
Gauss-kernel SVM	90.325	90.816	89.884	89.857	90.251
PW-ELM	94.561	95.092	93.829	94.966	95.362

Table 6: The $Overall-accuracy$ of classifiers

$Overall-accuracy$	**6**	**7**	**8**	**9**	**10**	**Average**
ELM	75.109	75.433	74.927	74.658	75.062	74.942
Gauss-kernel ELM	84.051	83.152	82.694	84.973	83.049	83.207
Gauss-kernel SVM	90.537	90.032	89.334	90.238	90.776	90.205
PW-ELM	94.466	94.752	95.093	95.372	93.964	94.746

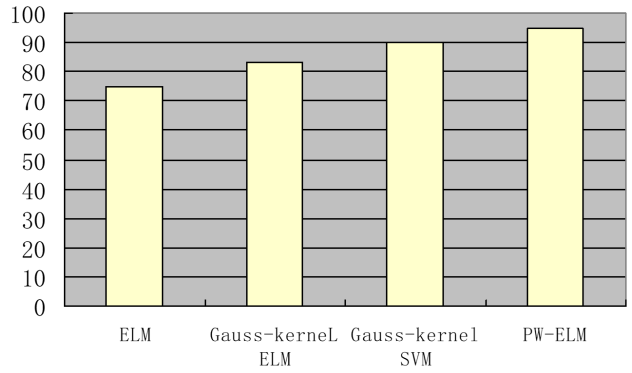


Figure 2: The $Overall - accuracy$ on the KDD 99

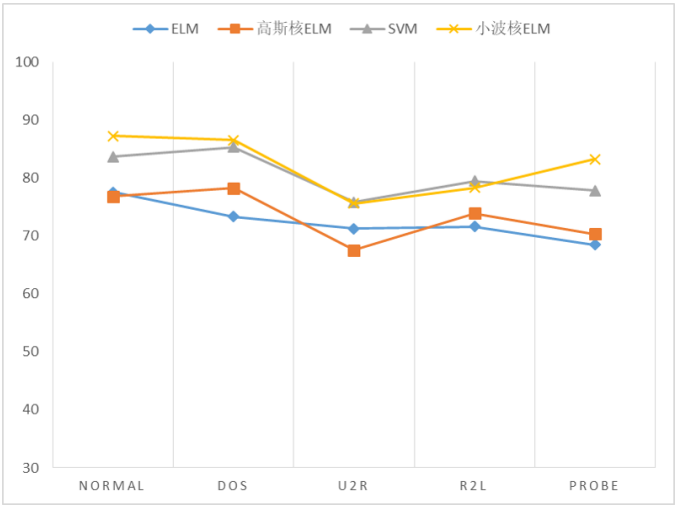


Figure 3: The $F - measure$ on the KDD 99

a good classification effect on Dos and PROBE. Gauss-kernel SVM is slightly better than PW-ELM in detecting R2L. However, in conclusion, compared with ELM, Gauss-kernel ELM and Gauss-kernel SVM, PW-ELM has more advantages and can be widely used in application.

4 Conclusions

In this paper, we introduced a wavelet-core extreme learning machine based on particle swarm optimization to detect abnormal traffic. Experiments show that the model can achieve good performance in the detection of abnormal network traffic. Compared with the ELM, Gauss-kernel ELM and Gauss-kernel SVM, the nonlinear approximation ability and generalization ability are greatly improved. And the model also solve the problem about the redundancy of hidden layer node and inefficiency. With the better learning efficiency and classification accuracy, the wavelet-core extreme learning machine based on particle swarm optimization can be widely use in the anomaly detection of network traffic data.

Acknowledgments

This study was supported by Projects U1636208 funded by National Natural Science Foundation of China (NSFC).

References

[1] M. Ahmed, "Collective anomaly detection techniques for network traffic analysis," *Annals of Data Science*, vol. 5, no. 4, pp. 497-512, 2018.

[2] M. Ahmed and A. N. Mahmood, "Novel approach for network traffic pattern analysis using clustering-based collective anomaly detection," *Annals of Data Science*, vol. 2, no. 1, pp. 111–130, 2015.

[3] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "E-ldat: A lightweight system for ddos flooding attack detection and ip traceback using extended entropy metric," *Security and Communication Networks*, vol. 9, no. 16, pp. 3251–3270, 2016.

[4] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, *et al.*, "On scaling decentralized blockchains," in *International Conference on Financial Cryptography and Data Security*, pp. 106–125, 2016.

[5] S. M. A. M. Gadal and R. A. Mokhtar, "Anomaly detection approach using hybrid algorithm of data mining technique," in *International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE'17)*, pp. 1–6, 2017.

[6] D. H. Hoang and H. D. Nguyen, "A pca-based method for iot network traffic anomaly detection," in *The 20th International Conference on Advanced Communication Technology (ICACT'18)*, pp. 381–386, 2018.

[7] H. Y. Hua, Q. M. Chen, H. Liu, Y. Zhang, and P. Q. YUAN, "Hybrid kmeans with knn for network intrusion detection algorithm," *Computer Science*, vol. 3, pp. 32, 2016.

[8] V. V. Kumari and P. R. K. Varma, "A semi-supervised intrusion detection system using active learning svm and fuzzy c-means clustering," in *International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC'17)*, pp. 481–485, 2017.

[9] Y. Ma, S. Liang, X. Chen, and C. Jia, "The approach to detect abnormal access behavior based on naive bayes algorithm," in *The 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'16)*, pp. 313–315, 2016.

[10] S. S. Roy, A. Mallik, R. Gulati, M. S. Obaidat, and P. V. Krishna, "A deep learning based artificial neural network approach for intrusion detection," in *International Conference on Mathematics and Computing*, pp. 44–53, 2017.

[11] S. Wang, D. Wang, C. Li, Y. Li, and G. Ding, "Clustering by fast search and find of density peaks with

data field," *Chinese Journal of Electronics*, vol. 25, no. 3, pp. 397–402, 2016.

[12] A. Wright and P. D. Filippi, "Decentralized blockchain technology and the rise of lex cryptographia," *SSRN Electronic Journal*, 2015. (`https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2580664`)

[13] L. Zheng, R. Diao, and Q. Shen, "Self-adjusting harmony search-based feature selection," *Soft Computing*, vol. 19, no. 6, pp. 1567–1579, 2015.

Biography

Yueyang Su Yueyang Su is a graduate student in School of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China. His research focuses on cyberspace security,artificial intelligence.

Jing Wan Jing Wan received her Ph.D. degrees in Beijing University of Chemical Technology. She is now a Professor at School of Information Science and Technology in Beijing University of Chemical Technology, China. Her research focuses on cyberspace security, artificial intelligence, intelligent information system, Knowledge Graph.

Junkai Yi Junkai Yi received his MS and Ph.D. degrees in Computer Science from Beijing Institute of Technology in 1995 and 1998 respectively. He is now a Professor at School of Information Science and Technology in Beijing Information Science and Technology University, China. His research focuses on cyberspace security, artificial intelligence, intelligent information system, text classification, pattern recognition.

Medical Image Encryption Based on Stream Cipher Algorithm and Krill Group

Chu Zhao, Shoulin Yin, Hang Li, and Yang Sun
(Corresponding author: Shoulin Yin and Hang Li)

Software College, Shenyang Normal University
Shenyang 110034, China
(Email: 352720214@qq.com; lihangsoft@163.com)

(Received Oct. 13, 2018; Revised and Accepted Feb. 7, 2019; First Online June 15, 2019)

Abstract

The traditional image encryption algorithm is with low key sensitivity, low efficiency, low security, image scrambling and diffusion and high correlation. When using stream cipher to encrypt medical, the key sequence is too long and difficult to store and distribute. Therefore, this paper puts forward a new stream cipher method to encrypt medical image. First, medial image is coded as the text form. And then, we use the mutated character encoding table to encode text as the plaintext. Finally, we use krill group-based stream cipher algorithm to encrypt plaintext. By comparing the new method with the state-of-the-art encryption methods, experimental results show that the new method has greatly shortened the length of storing and distributing key sequence. The new algorithm has certain advantages in statistical performance, robust performance and key sensitivity, and also meets the requirements of image security and real-time performance.

Keywords: Character Encoding; Krill Group; Medical Image Encryption; Stream Cipher Algorithm

1 Introduction

Image encryption [12,20] is widely used in military reconnaissance and public security inspection. Medical image transmission in open network environment is vulnerable to eavesdropping attacks, so it is necessary to encrypt medical image before transmission [4–7,16,17]. However, how to balance the scrambling performance, security performance, robustness performance and the quality maintenance of decryption image is always difficult. Stream cipher [1,8] is a symmetric key encryption where the cryption key used to encrypt the binary image is randomly changed so that the cipher image produced is mathematically impossible to break. The advantage of using stream cipher is that the execution speed is higher when compared to block ciphers and have lower hardware complexity.

The stream cipher methods of medical image encryption include Martin image key system, rapid security sequence key, discrete cosine transform and stream cipher encryption methods, based on chaotic sequences and discrete wavelet transform partial encryption system, as well as classic RC4 stream cipher and Buddha sequence cipher system [2, 15]. These methods have a common disadvantage that the key sequence is too long.

Sudeepa [13] stated that maximum length sequence was applied for the RNS (Residue Number System) based additive stream cipher system. When the key sequence period was greater than the size of plain text, the system approaches secured one time pad cipher system. Imamura [9] analyzed the integrity of these schemes both in the standard INT-CTXT (integrity of ciphertext) notion and in the RUP (releasing unverified plaintext) setting called INT-RUP notion. Pu [14] presented a new algorithm by combining the true random sequences and the Tree Parity Machine (TPM), which was proven experimentally. Different from common method, true random sequences were proposed as dynamic inputs of TPM in this work compared to the pseudo-random sequences in the latest report. Xiao [19] proposed a new digital watermarking algorithm in encrypted image based on compressive sensing measurements and 2-D discrete wavelet transform (DWT). However, they cannot guarantee the privacy security.

In this paper, a stream cipher method similar to Vernam cipher employing an krill group [**?**] based approach to generate keys for encrypting medical images is proposed. This novel approach called stream cipher krill group image encryption (SCKGIE) algorithm is proposed to generate the keystream. The novelty in the approach is that an krill group approach is used to generate the keystream used for encryption based on the distribution of characters in the plain text denoting the image so that the keys in the keystream are encoded using a mutated character code table which would enable to increase the security of the system.

The advantage of the proposed stream cipher method

is that it would increase the security of the system by encoding the keys in the keystream and the characters in the plain text representing the encoded binary image using the mutated character code table. It reduces the number of keys to be stored and distributed when compared to that of Vernam cipher considered to be the perfect cipher. It overcomes the drawback of boolean cellular automaton method for image encryption and scan pattern method of image encryption in terms of the number of keys to be stored and distributed. The length of the key in SCKGIE algorithm is less when compared to DWT and chaos based image encryption method.

2 Algorithm Description

2.1 Encryption Process

First, the medical image is encoded as a text form using the characters in the ASCII code table (ASCII code values are from 32 to 126), and then it uses the mutated character code table to encode characters in the text, the result is as the plaintext. To encode the initial key sequence, the characters appearing in the plaintext use the mutated character encoding table to encode. Key sequences that do not appear in plaintext are encoded by using ASCII values. The same order of key sequences cannot be used to ensure security. The advantage of using a mutated character encoding table to encode characters in key sequences and text is that it can enhance the security of the system.

If the length of the plaintext sequence is greater than the length of the key sequence, the original key sequence is added with a predetermined value to generate the key sequence corresponding to the plaintext sequence to ensure the character length greater than the length of the key sequence.

The predetermined value is calculated by dividing the plaintext of the medical image into two parts as shown in Equation (1).

$$Indexvalue = int[length(P)/2]. \quad (1)$$

The plaintext sequence is partitioned into the size of the length of the key sequence, and the first corresponding key sequence is composed of the original key sequence. The key sequence corresponding to block i is obtained by adding the predetermined value to the key sequence corresponding to block $i-1$ as shown in Equation (2).

$$S_i = S_{i-1} + Indexvalue(i \geq 2). \quad (2)$$

Key value and plaintext value use XOR to obtain the ciphertext image.

2.2 Decryption Process

Decryption and encryption process use the same key sequence. The mutated character encoding table of contents and initial key sequence are sent to the receiver through the security channel, the receiver encodes the received initial key sequence through a simple table lookup operation. The character in key sequence that does not appear in character encode table is used ASCII to code. The receiver calculates the predetermined value based on the length of the ciphertext image. The corresponding key sequence is generated for the part of the ciphertext image exceeding the length of the key sequence. The key sequence and ciphertext image are subjected to XOR operation to obtain the plaintext, which is then decoded using the mutated character encoding table to get the text that represents the medical image, which is then decoded to obtain the original medical image.

2.3 Text Form for Medical Image

In order to encode the medical image as the text form represented by the characters in the ASCII table, each row of the 0,1 bit stream of the medical image is divided into several groups. There are only 95 characters in the ASCII table, so each group contains up to 94 bits. The grouping process continues until all the bit sequences are divided into one group and each group is coded separately. Characters from ! (ASCII code value is 33) to ∼ (ASCII code value is 126) are assigned a value of 1-94 respectively. To encode the 0, 1 bit stream of a medical image into text, the character value is established by an Equation (3).

$$Charactervalue = column(mod94). \quad (3)$$

If the value of the corresponding column position of the medical image is 0, replacing with character *space*. If the value of the corresponding column position of the medical image is 1, and the value of the column position mod94 is 0, replacing with character ∼. If the value of the corresponding column position of the medical image is 1, and the value of the column position mod94 is not 0, then replacing with the character value with corresponding character value. Strings are concatenated to form text to represent coded medical images.

2.4 Krill Swarm Algorithm

Krill swarm optimization algorithm is one of simulation swarm intelligence algorithms by simulating the action of krill, which was proposed by Alavi [3]. Each krill will be attracted or repelled by a certain range neighboring krill, so it can make local optimization. And the food center determined by fitness of krill would guide krill to make global optimization. In addition, the time interval needs to be adjusted, and the rest required parameters can be obtained from the research achievements of krill real ecological behavior. Meanwhile, krill swarm algorithm adopts Lagrangian model, therefore, the performance of krill swarm is superior to other optimization algorithms. The detailed processes of krill swarm algorithm are as follows:

1) Determine the Lagrangian model of krill swarm.
2) Motion induced by other krill individuals.
3) Foraging motion.
4) Stochastic diffusion process.
5) Updating krill positions.

3 Proposed Image Encryption

The objective function of the algorithm is to generate a key sequence, which satisfies the constraint that the energy value is greater than or equal to 80%. The energy value released by the krill is calculated by the counter, as Equation (4).

$$Energy(K_i) = \frac{count(C_j^i \in P)}{length(i)}. \quad (4)$$

Here K_i represents maximum number of krill, $i = 1, 2, \cdots$. C_j^i represents the key sequence length of $j-th$ character in the $i-th$ key sequence. P is the text of the medical image.

The krill path with the maximum energy value greater than or equal to the specified threshold is the solution of the problem. The key sequence is selected to encrypt the medical image. This allows the mutated character encoding table to encode most of the characters in the key sequence to increase the security of system.

Entropy coding has some properties related to cryptography. A character encoding tree is generated based on the statistical distribution of characters in text to encode characters in text and key sequences. The character encoding table is generated according to the character encoding tree. The purpose of establishing character encoding table is to encode the character appearing in text and key sequence through simple table lookup operation. Mutations in the character encoding tree can occur randomly at any node to enhance the security of the system. Research has shown that encryption should be combined with entropy coding using multiple statistical tables, and the benefit of using multiple statistical tables is that encryption can be done at a reasonably high security level.

The specific processes of the initial key sequence and text encoding are as follows:

1) Count the characters in the text and conduct Huffman encoding according to the possibility of character occurrence. The left branch of the tree is marked 0, and the right branch is marked 1.
2) The initial tree mutates in different non-leaf nodes by switching left and right branches. The character values in the character encoding tree are set up in decimal form.
3) For characters with the same value, the length of encoding is added to the character value to make the character value in the table correspond to the character value one to one.

4 Security of New Scheme

4.1 Key Space

The key sequence space that can be generated by 95 characters is given in Equation (5).

$$\sum_{i=1}^{95} \frac{95!}{(95-i)!} \approx 95!e. \quad (5)$$

In this size, it would take about 3×10^{125} years to decrypt the key, even though the world's fastest supercomputer, tianhe-1.

The character appeared in plaintext in key sequence will be replaced by the value in mutated character encoding table, which can increase the security of the system. Because character encoding table generation depends on the characters in plain text, the adversary needs to anticipate all possible sequences of character encoding trees. The Huffman tree encodes t characters, and the initial Huffman tree has $t-1$ non-leaf nodes, and the tree that may be different from the original tree through mutation has $2^{t-1}-1$. The key space generated by character encoding table is given by Theorem 1 and Theorem 2.

Theorem 1. *The number of possible character code tables is* 2.64×10^{176} *for images of size* $m \times n$ *where* $n \geq 94$.

Proof. Medical images are encoded as text. A character encoding tree is generated based on the occurrence probability of characters in text. The value of each character is generated by traversing the tree. The maximum possible number of characters is 95, the possible number of characters can be from 1 to 95, and the characters can appear in any order in the tree. Therefore, the possible number of initial character encoding trees is given in Equation (5). Each initial character encoding tree has $2t-1$ variants. Therefore, the maximum possible number of tables is:

$$\sum_{i=1}^{95} \frac{95!}{(95-i)!} \times 2^{i-1} \approx 2.64 \times 10^{176} \approx 2^{586}.$$

□

Theorem 2. *For images of size* $m \times n$, *where* $n < 94$ *the number of possible character code tables is:*

$$\sum_{i=1}^{n+1} \left(\frac{(n+1)!}{(n+1-i)!}\right) \times 2^{i-1}. \quad (6)$$

Proof. For images where the number of columns n is less than 94, the maximum number of characters will be $n+1$. Since the character code tree is generated based on the number of occurrence of the characters, the tree can have the characters of all possible orderings. That is out of the $n+1$ characters there can be all possible 1 or 2 or $\cdots$ or $n+1$ combination of all possible orderings without repetition. Thus the total number possible initial character

code tree is $\sum_{i=1}^{n+1}(\frac{(n+1)!}{(n+1-i)!})$. Each initial character code tree has 2^{t-1} tables as discussed above where $t-1$ are the number of inner nodes in a tree. Thus the total number of maximum possible tables is given in Equation (6). □

4.2 Histogram Analysis

Histogram describes the distribution of pixel points by drawing the number of pixels in each pixel level. In order to prevent information leakage, ciphertext image and plaintext image should have different statistical features. By analyzing the histogram of the ciphertext image and the original image, the number of pixels in each gray level is significantly different. The x-axis of the histogram represents the change in hue, and the y-axis represents the number of pixels in a particular hue. We chose two grayscale medical images with size 128×128. Figure 1,2 are the original images and histograms. Figures 3, 4 are the encrypted images and histograms. Figures 5, 6 are the decrypted images and histograms.

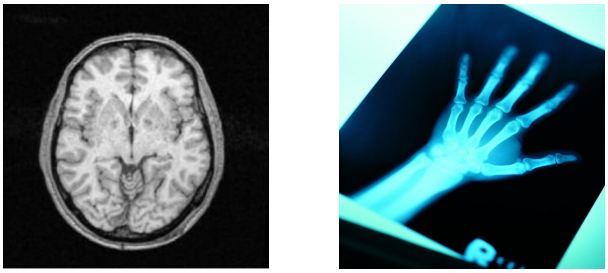


Figure 1: Original images

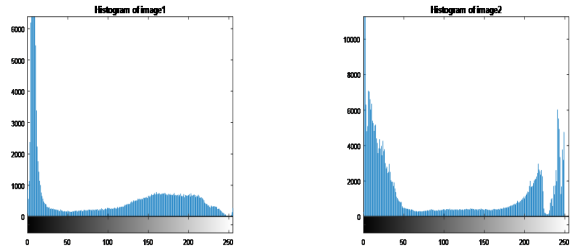


Figure 2: Histogram of original images

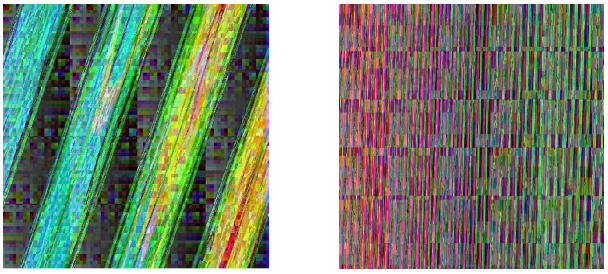


Figure 3: Encrypted images

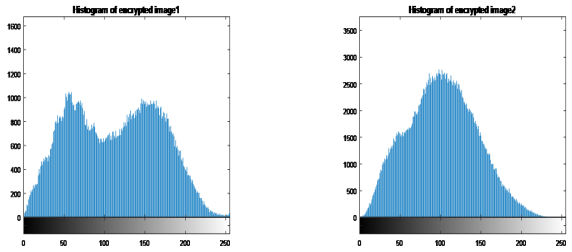


Figure 4: Histogram of encrypted images

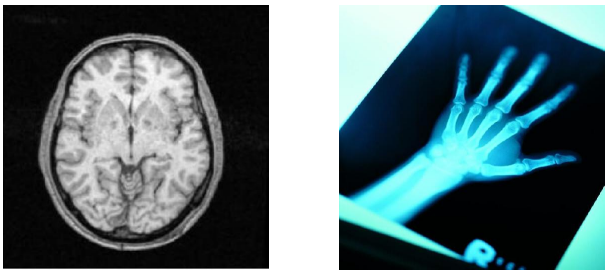


Figure 5: Decrypted images

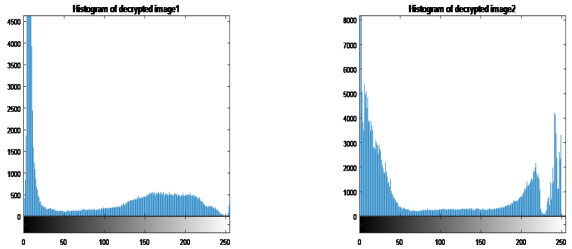


Figure 6: Histogram of decrypted images

It can be seen from the statistical histogram of ciphertext that all the peaks are almost uniformly scattered, and the plaintext does not have a statistical similarity with the plaintext. Therefore, using the new medical image encryption algorithm, ciphertext does not provide clues for the statistical attack.

4.3 Correlation Coefficient Analysis

The correlation coefficient between plaintext and ciphertext also shows their similarities. From the similarity between them, it can be inferred that the correlation coefficient between plaintext and ciphertext determines whether the algorithm has good diffusion and chaos characteristics, as well as resistance to statistical attacks.

If the correlation coefficient is between 0.5 and 1.0 or between -0.5 and -1.0, it means that there is a strong positive correlation or a strong negative correlation between them. If the correlation coefficient is between 0.0 and 0.5 or between -0.1 and -0.5, it means that there is a weak positive correlation or a weak negative correlation between them. Table 1 shows the correlation coefficients between plaintext and ciphertext, where key sequence lengths are 5 and 15, respectively.

Table 1: Correlation coefficient between plaintext and ciphertext

Image	Length=5	Length=15
Image1	0.0097	0.0074
Image2	0.0096	0.0073

It can be seen from Table 1 that there is a weak correlation between plaintext image and ciphertext image, which means that the medical image encryption algorithm based on SCKGIE encryption algorithm can resist statistical attacks.

5 Experiment Results and Analysis

In order to verify the effectiveness of proposed image encryption, We make comparison with RCM [10] and CCSC [11] conducted on MATLAB. We analyze differential attack, information entropy and robustness of new encryption method.

5.1 Differential Attack

Modifying the original plaintext image, high sensitivity is an important attribute in the image encryption algorithm. General experimental method is that it only modifies one pixel in the original image, and then observe the change of image to get quantitative relationship between ciphertext image and original image, if the original image has small changes that can cause larger cipher text image change, it argues that the encryption algorithm has good robustness for differential attack.

In order to test the effect of a pixel change on the entire ciphertext image, two famous measurement methods are adopted: UACI and NPCR. Setting two encrypted images, there is only one different pixel in the two images as I_1 and I_2, the corresponding gray values are $I_1(i,j)$ and $I_2(i,j)$. Define a bipolar array B, I_1 and I_2 have the same image size. $B(i,j)$ is determined by $I_1(i,j)$ and $I_2(i,j)$. If $I_1(i,j) = I_2(i,j)$, then $B(i,j) = 1$. Otherwise, $B(i,j) = 0$. So

$$NPCR = \frac{\sum_{i,j} B(i,j)}{W \times H} \times 100\%.$$

Where W and H represent the width and height of the encrypted image, and NPCR measures the ratio of the number of pixels with different pixel values between the two images to the total pixel values.

$$UACI = \frac{1}{W \times H}[\sum_{ij} \frac{I_1(i,j) - I_2(i,j)}{255}] \times 100\%.$$

UACI measures the average strength of the two images, and tests the medical images by modifying one pixel. The results are shown in Table 2. From the UACI and NPCR values in the table, it can be seen that the encryption algorithm in this paper has a great sensitivity to the small difference of the original image.

Table 2: NPCR, UACI comparison with different methods

Image	RCM	CCSC	Proposed algorithm
Image1(NPCR)	94.4	95.6	99.2
Image2(NPCR)	93.2	94.7	98.5
Image1(UACI)	33.7	32.1	27.6
Image2(UACI)	33.1	31.9	28.7

Table 3: Information entropy comparison

Method	Image1	Image2
RCM	0.715	0.726
CCSC	0.727	0.728
Proposed method	0.834	0.828

5.2 Information Entropy

Information entropy denotes the degree of uncertainty system, and it is used to describe the uncertainty of image information. The information entropy can be used to analyze the distribution of gray value in the image. Let $P(m_i)$ be proportion of pixel with gray value m_i in image and $\sum_{i=0}^{255} P(m_i) = 1$. The information entropy of the pixel is defined as:

$$H(m) = -\sum_{i=0}^{255} P(m_i \log_2 P(m_i)).$$

The comparison results are as shown in Table 3.

5.3 Robustness Analysis

Since noise is inevitably introduced in the encryption process, the robustness of the algorithm in this paper is tested, and PSNR value is used to judge the quality of the encrypted image as defined below:

$$PSNR = 10log\frac{WH255^2}{\sum_{i=0}^{H-1}\sum_{j=0}^{W-1}(f_1(i,j) - f_2(i,j))^2}$$

Where $f_1(i,j)$ is the pixel value of the original image pixel (i,j), and $f_2(i,j)$ represents the pixel value of the decryption terminal pixel (i,j). Obviously, the higher the PSNR value is, the better the performance of the encryption algorithm is. Table 4 is the PSNR value of Image1 and Image2. Obviously, the chaotic encryption algorithm in the transform domain has good robustness.

Table 4: PSNR comparison with different methods

Image	RCM	CCSC	Proposed method
Image1	52.18	53.75	59.76
Image2	52.38	54.55	58.59

6 Conclusion

This paper proposes a stream cipher based on krill group algorithm method, this new method is compared with both the sequence code method, which greatly reduces the need of storing and distributing the key sequence length, solves the key storage and distribution problem. At the same time, it uses the mutated character encoding table to encode the character in key and plaintext, which makes it hard for the adversary to break character coding table, this improves the security of system. And the experimental results show that the system has a high security degree by analyzing differential attack, information entropy and robustness.

7 Acknowledgments

This study was supported by the Natural Science Fund Project Guidance Plan in Liaoning Province of China (No. 20180520024).

References

[1] A. Belmeguenai, Z. Ahmida, S. Ouchtati, *et al.*, "A novel approach based on stream cipher for selective speech encryption," *International Journal of Speech Technology*, vol. 20, no. 9, pp. 1-14, 2017.

[2] X. Chai, Z. Gan, Y. Chen, *et al.*, "A visually secure image encryption scheme based on compressive sensing," *Signal Processing*, vol. 134, pp. 35-51, 2017.

[3] A. H. Gandomi, A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science & Numerical Simulations*, vol. 17, no. 12, pp. 4831-4845, 2012.

[4] L. C. Huang, M. S. Hwang, L. Y. Tseng, "Reversible and high-capacity data hiding in high quality medical images," *KSII Transactions on Internet and Information Systems*, vol. 7, no. 1, pp. 132–148, 2013.

[5] L. C. Huang, M. S. Hwang, and L. Y. Tseng, "Reversible data hiding for medical images in cloud computing environments based on chaotic Henon map," *Journal of Electronic Science and Technology*, vol. 11, no. 2, pp. 230–236, 2013.

[6] 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.

[7] L. C. Huang, L. Y. Tseng, M. S. Hwang, "A reversible data hiding method by histogram shifting in high quality medical images", *Journal of Systems and Software*, vol. 86, no. 3, pp. 716–727, Mar. 2013.

[8] T. Hwang, P. Gope, "Robust stream-cipher mode of authenticated encryption for secure communication in wireless sensor network," *Security & Communication Networks*, vol. 9, no. 7, pp. 667-679, 2016.

[9] K. Imamura, K. Minematsu, T. Iwata, "Integrity analysis of authenticated encryption based on stream ciphers," *International Journal of Information Security*, no. 3, pp. 1-19, 2016.

[10] M. Kumari, S. Gupta, "A novel image encryption scheme based on intertwining chaotic maps and RC4 stream cipher," *3d Research*, vol. 9, no. 1, pp. 10, 2018.

[11] Z. Lin, S. Yu, X. Feng, *et al.*, "Cryptanalysis of a chaotic stream cipher and its improved scheme," *International Journal of Bifurcation & Chaos*, vol. 28, no. 7, 2018.

[12] J. Liu, S. L. Yin, H. Li and L. Teng, "A density-based clustering method for K-anonymity privacy protection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 8, no. 1, pp. 12-18, Jan. 2017.

[13] K. B. Sudeepa, G. Aithal, "Generation of maximum length non-binary key sequence and its application for stream cipher based on residue number system," *Journal of Computational Science*, vol. 21, pp. 379-386, 2016.

[14] X. Pu, X. J. Tian, J. Zhang, *et al.*, "Chaotic multimedia stream cipher scheme based on true random sequence combined with tree parity machine," *Multimedia Tools & Applications*, vol. 76, no. 19, pp. 1-15, 2016.

[15] L. Teng, H. Li, S. Yin, "A multi-keyword search algorithm based on polynomial function and safety inner-product method in secure cloud environment," *International Journal of Network Security*, vol. 8, no. 2, pp. 413-422, 2017.

[16] 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.

[17] 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.

[18] G. G. Wang, A. H. Gandomi, A. H. Alavi, "A chaotic particle-swarm krill herd algorithm for global numerical optimization," *Kybernetes*, vol. 42, no. 6, pp. 962-978, 2013.

[19] D. Xiao, Y. Chang, T. Xiang, *et al.*, "A watermarking algorithm in encrypted image based on compressive sensing with high quality image reconstruction and watermark performance," *Multimedia Tools & Applications*, vol. 76, no. 7, pp. 1-32, 2017.

[20] S. L. Yin and J. Liu, "A K-means approach for map-reduce model and social network privacy protection," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, no. 6, pp. 1215-1221, Nov. 2016.

Biography

Chu Zhao received the M.Eng. degree from Shenyang Normal University, Shenyang, Liaoning province, China in 2011. Her research interests include Network Security and Data Mining. Email:910675024@qq.com.

Shoulin Yin received the B.Eng. and M.Eng. degree from Shenyang Normal University, Shenyang, Liaoning province, China in 2016 and 2013 respectively. Now, he is a doctor in Harbin Institute of Technology. His research interests include Network Security, image processing. Email:ysl352720214@163.com.

Hang Li obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hang Li is a full professor of the Kexin software college at Shenyang Normal University. He is also a master's supervisor. He has research interests in wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Prof. Li had published more than 30 international journal and international conference papers on the above research fields. Email:lihangsoft@163.com.

Yang Sun obtained his master degree in Information Science and Engineering from Northeastern University. Yang Sun is a full associate professor of the Kexin software college at Shenyang Normal University. He is also a department head of network engineering. He has research interests in wireless networks, mobile computing, cloud computing, social networks and network security. Yang Sun had published more than 15 international journal and conference papers on the above research fields.

LinkedIn Social Media Forensics on Windows 10

Ming Sang Chang and Chih Ping Yen
(Corresponding author: Chih Ping Yen)
Department of Information Management, Central Police University
Taoyuan 33304, Taiwan
(Email: peter@mail.cpu.edu.tw)
(Received Mar. 12, 2019; Revised and Accepted Sept. 3, 2019; First Online Sept. 16, 2019)

Abstract

Many people have gradually changed their way of living habits on account of the great popularity progression of social networking sites. There are varied kinds of social networking sites coming out in recent years, for example, Facebook, Twitter, Instagram, LinkedIn. Furthermore, social networking sites have already made people more convenient to make friends and communicate with each other much easier than before. However, there are some problems we should concern. Owing to the cyberworlds are flourishing, there are several kinds of crimes emerge in endlessly in recent years. This paper focuses on the digital forensics of LinkedIn by running on three different browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge. They are running respectively under windows 10 operating system. In our work, we strive to find digital evidences that user has been done on the computers. We make use of authoritative digital forensic tools to obtain significant evidences and analyze the correlation between these evidences in detail. Besides, we will find out which behaviors of the suspect will leave what kind of evidences on the computer. These findings could be important references for the law enforcement agency to investigate digital crime.

Keywords: Crime Investigation; Digital Forensics; LinkedIn; Social Media

1 Introduction

In recent years, the popularity of social networking sites has given rise to the number of social networking users for recreation and business purposes. A social network is a community where people across the globe world online that can develop a network with different individuals for a specific purpose [1]. Besides, the prevalence of these social networking websites has changed the living habits of many people. These people usually browse social networking sites to relieve their working pressure or any other kinds of pressures in their daily life.

People can make use of social networking sites to build up their profile. A profile is a list of identifying information that can portray users' online identity, including photographs, name, birthday, hometown, personal interest and so on [9]. Furthermore, social networking sites can connect people and maintain relationships from all parts of their lives [6]. They can share everything with their friends on the websites. There is no doubt that people have incorporated social networking sites into their lives and made using social networking sites as frequent daily activities.

Due to the advance of technology, the type of crime is getting much more complex than before. At present, traditional crime is on the decrease. In other words, high technology crime is increasing nowadays. There are a lot of perpetrators using social networking sites to commit the cybercrime because of its convenience and anonymity characteristics. Therefore, the traditional crimes such as killing people, domestic violence, stealing and robbing are decreasing nowadays. On the contrary, computer crime and cybercrime have already become the mainstream of all the crimes. Cybercrime refers to a perpetrator that abused or destroyed a computer to commit a crime. Therefore, cybercrime is definitely different from traditional crime. The following shows the characteristics of cybercrime [7]:

- Making use of the computer characteristics to commit the crime.
- The high dark figure of crime.
- The time and dimension features between crime behaviors and crime results.
- Take a computer as a crime scene.
- Take a computer as a target.

Over the past 10 years, the terrorists use the Internet have become of great concern. The gang of terrorist has successfully used the Internet to enlarge their memberships [11]. This will cause widespread harm to Internet victims.

According to the survey of National Police Agency, Ministry of the Interior Republic of China, the statistics show the cybercrimes happened in Taiwan between

January and June in 2017, there are 6,567 cybercrime cases occurred. The cybercrime ratio increases 4.39 percentages relative to the same period of last year. However, the perpetrators who are at the age of 18 to 23 called adolescents are increasing 28.07 percentages relative to the same period of last year. The victims who are more than 50 years old are increasing 43.54 percentages relative to the same period of last year [15]. Over the past few years, various kinds of cybercriminals have emerged endlessly due to the anonymity characteristic of the Internet. Therefore, anonymity is largely tied to the cybercrime nowadays. Moreover, it is also claimed that the anonymity characteristic allows perpetrators to use the Internet without the possibility of detection. Catherine D. Marcum, *et al.* categorized different types of social networking criminality, for instance, texting, identity theft, cyberbullying, digital piracy, sexual violence, and so forth [13]. Therefore, we can realize that social networking websites have seriously become a hotbed of cybercrimes based on these significant literatures. According to the survey of eBizMBA [10], popular social networking sites are prevalent nowadays, such as Facebook, YouTube, Twitter, Instagram, LinkedIn and so on. Many of them have over than 100 million members, a quite large number for the time.

This paper focuses on the digital forensics of LinkedIn by running on three different browsers, including Google Chrome, Mozilla Firefox and Microsoft Edge. They are running respectively under windows 10 operating system. In our work, we strive to find digital evidences that user has been done on the computers. The results will be served as a reference for the future researchers in social network cybercrime investigation or digital forensics.

The rest of this paper is organized as follows. In the next section, we present the related works. In Section 3, we introduce our investigation methodologies. In Section 4, we present results and findings of digital forensics on LinkedIn. Finally, we summarize the conclusions.

2 Related Works

2.1 LinkedIn Social Networking Site

LinkedIn is a business and employment-oriented service that operates via websites and mobile applications. It founded on December 28 in 2002 and launched on May 5 in 2003.

LinkedIn [19] is mainly used for professional networking, including employers posting jobs and job seekers posting their curriculum vitae. According to the survey of Alexa [2], LinkedIn was ranked 31st relative to other social networking websites in the world. As of April 2017, LinkedIn had 500 million members in 200 countries, out of which more than 106 million members are activities [12]. LinkedIn allows members to create profiles and connections to each other in an online social network that may represent real-world professional relationships. Members can invite anyone to become a connection [17].

Such dissemination of confidential information is possibly more likely concerning social networking applications such as LinkedIn where users may be actively looking for employment or maybe in contact with individuals from competitor organizations.

However, there are many kinds of literature focus on the forensic analysis of social networking sites nowadays. Azfar *et al.* [5] proposed the utility model for the evidence extraction of five social networking applications, including Twitter, POF Dating, Snapchat, Fling and Pinterest.

Neha [16] focused on the forensic analysis of WhatsApp application on storage devices and volatile memory. Mutawa *et al.* [14] focused on the forensic analysis of three popular social networking sites, including Facebook, Twitter, and Myspace. Dezfouli, *et al.* [8] examined four well-known social networking applications, Facebook, Twitter, LinkedIn, and Google+. They were able to recover artefacts, such as usernames, passwords, login information, personal information, posts, messages and comments from these social networking sites. However, they only focus on the mobile phone forensics for these four applications. They didn't perform computer forensics which refers to browser forensics for these four applications. As a result, we take the LinkedIn application as one of our experiment targets.

This paper studies the behavior of a user who log into LinkedIn from different browsers. We strive to extract the evidence of posts creation, making comments, chatting records, browsing behaviors, adding friends and so forth. All of these behaviors are conducted under Windows 10 operating system. Furthermore, this paper analyzes correlations between these evidences and discusses how these evidences can help law enforcement agencies to investigate a crime.

2.2 Tools

Due to the high dynamics and heterogeneity of social media, digital forensics can use different and complex software tools to conduct effective and legal evidence collection [3]. There are many forensic tools on the market today. The mainstream of digital forensic products such as Autopsy, Forensic Toolkit and EnCase forensic have support digital forensics. The study described in this paper has been executed by a series of processes. In the experiments, the hard disk and memory were examined to extract and analyze the data generated by LinkedIn website. With the advanced development of forensic tools, the forensic tools and techniques should keep investigators ahead of the criminals [18].

Arthur *et al.* [4] conducted an investigation into some of the forensic tools, including PC Inspector File Recovery, EnCase, Forensic Toolkit and FTK Imager. However, the main function of FTK Imager is to view and to image storage devices. In light of these advantages, we adopt AccessData FTK Imager V4.1.1 to create an image file for the hard disk. Forensic Toolkit is a computer forensics software made by AccessData. It scans a hard disk

searching for various types of information. The toolkit comprises a standalone disk imaging program called FTK Imager. The FTK Imager is a simple tool that saves an image of a hard disk in a file. The result is an image file that can be saved in several formats.

On the other hand, there are many kinds of tools used for memory forensics nowadays. The manipulation of these memory forensic tools is roughly different, but the theorem concepts are the same. The goal of these tools is to read physical memory for the sake of achieving memory forensics. Therefore, this paper adopts the MANDIANT tool to create an image file for the memory. MANDIANT is an open source tool that can be downloaded on the Internet. There are a few basic functions describe as follows:

- MemoryDD.bat: This batch file is used to create an image file for volatile memory.
- Process.bat: This batch file is used to list all the running processes.
- DriverSearch.bat: This batch file is used to list which SYS file is loading on the computer.
- HookDetection.bat: This batch file is used to list which hooks file is executing on the computer.

In the experiment, in order not to influence the integrity of digital evidence, this paper makes use of MemoryDD.bat file to dump the memory for the sake of creating image files. Finally, this paper makes use of AccessData FTK Imager V4.1.1 to analyze all the image files which were generated by the previous processes. However, the most important of all is that we take another clean computer to analyze these image files.

In this paper, all the experiments were conducted on the real computer system. The computer system was installed Windows 10 professional 64-bit operating system. The central processing unit is Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz. The memory size is 8 Gigabytes. This paper selects two common browsers, including Google Chrome V59.0.3071.115 and Microsoft Edge V 40.15063.0.0.

3 Methodology

3.1 Research Goal

The research described in this paper is done by a series of processes, each involving a specific scenario. In the experiment, we log into LinkedIn websites via three different browsers. All of these operations are executed under Windows 10 operating system. After login, we do a series of same behaviors, such as login account, adding friends, chatting with friends, writing posts, making comments, clicking the "Like" button. Afterward, we make use of Forensic Toolkit Imager to extract digital evidence of these behaviors left. Finally, we analyze and compare the difference between these digital evidences.

3.2 Experiment Elaboration

In order to ensure the integrity of digital evidence and avoid the interference between digital evidences, we separate the experiments into three scenarios according to the different browsers. We chose three clean computers and each of them was installed Windows 10 professional operating system. We did these three scenarios in different computer environments. They were not placed on the same computer system. Afterward, we performed a series of behaviors on the LinkedIn. The following shows the details for these three scenarios.

3.2.1 Scenario 1: Google Chrome

In scenario 1, all the operations were conducted via Google Chrome browser. We entered the personal account and password to log into the LinkedIn website. After log into the LinkedIn website, we created posts and uploaded the pictures. Moreover, we did a lot of users common activities, for example, adding friends, chatting with friends, making comments, clicking the "Like" button and so forth. After we did these user common activities, we didn't do anything anymore. We created image files for the hard disk and memory respectively. Thereafter, we adopted Forensic Toolkit Imager to extract and analyze the digital evidence.

3.2.2 Scenario 2: Mozilla Firefox

In scenario 2, all the operations were conducted via the Mozilla Firefox browser. We entered the personal account and password to log into the LinkedIn website. After login, we created posts and uploaded the pictures. Moreover, we did a lot of users common activities, for example, adding friends, chatting with friends, making comments, clicking the "Like" button and so forth. After we did these user common activities, we didn't do anything anymore. We created image files for the hard disk and memory respectively. Thereafter, we adopted Forensic Toolkit Imager to extract and analyze the digital evidence.

3.2.3 Scenario 3: Microsoft Edge

In scenario 3, all the operations were conducted via the Microsoft Edge browser. We entered the personal account and password to log into the LinkedIn website. After login, we created posts and uploaded the pictures. Moreover, we did a lot of users common activities, for example, adding friends, chatting with friends, making comments, clicking the "Like" button and so forth. After we did these user common activities, we didn't do anything anymore. We created image files for the hard disk and memory respectively. Thereafter, we adopted the Forensic Toolkit Imager to extract and analyze the digital evidence.

4 Results and Findings

We log into the LinkedIn website by entering the email account and password on the computer. Afterward, we execute a series of processes, such as creating posts, chatting with friends, making comments, adding friends. After executing these user common activities, we create image files for the hard disk and the memory. We separate analysis procedure into two parts, hard disk and memory. We make use of one practical function of FTK Imager to execute a quick search for the keyword. The following are our analyses and description of forensic results.

4.1 Findings: Scenario 1: Google Chrome

4.1.1 Account and Password

In the hard disk, there are various kinds of evidence we can extract. First, we can find out user account information by searching the key string "www.linkedin.com", as shown in Figure 1. We can find two keywords in the context, there are "session_key" and "session_password". These two keywords reveal important information about the e-mail login account and password. However, we can't find password information because the text of password was garbled. We can't comprehend its meaning by our first intuition. Therefore, we infer that the password may be encrypted.

In the part of memory, we can only find an e-mail login account, but the password was garbled as well.

4.1.2 Posting Evidence

Every posting has it's unique post ID. Post ID is a string of numbers. When a user writes a post on the feed, the system will automatically assign a unique ID to the posting, for example, "https:// www.linkedin.com/feed/update/urn:li:activity:6378505126512074752/". We can easily realize the post ID is 6378505126512074752. Therefore, the posting network address is often built in the form of "https://www.linkedin.com/feed/update/urn:li:activity: Post ID". In the hard disk, by searching the key string "https://www.linkedin.com/feed/update/urn:li:activity:", we can find creating evidence of posting, as shown in Figure 2. Therefore, we can match the post ID we found in the image file and the network address. If both of them are the same, we can definitely infer that they must have posted that article on the LinkedIn website in the past. Moreover, we also found post contents by searching the key string, as shown in Figure 3.

In the part of memory, we can also find creating evidence and posting contents by searching the key strings.

4.1.3 Making Comment Evidence

LinkedIn allows any people to make any comments on any articles. In the hard disk, by searching the keyword "comment", we can find comment evidence that we made on the other user's posting, as shown in Figure 4. On the other hand, by reverse searching the key string, we can find comment content, as shown in Figure 5.

In the part of memory, the situation is the same as in the hard disk, we can also find comment evidence and its content by searching the keyword and the key string.

4.1.4 Chatting Records

In the hard disk, we can extract chatting record evidence. When a user chatted with friends, the system will automatically record chat ID. Moreover, the network address of chatting page would show friend's chat ID, for example, "https://www.linkedin.com/messaging/thread/63784999 18738399232". Therefore, we can easily realize that the majority of the chatting record network address is often built in the form of "https://www.linkedin.com/messaging/thread/Chat-ID". By searching the key string "https://www.linkedin.com/messaging/thread/", we can easily find the chatting record evidence, as shown in Figure 6. Furthermore, we can also find chatting content evidence by searching the key string in reverse, as shown in Figure 7.

In the part of memory, we can also find chatting record evidence by searching the key string "https://www.linkedin.com/messaging/thread/", and find chatting content evidence by searching the key string in reverse.

4.1.5 Clicking "Like" Button Evidence

There is a function on the LinkedIn website called "Like". If people like a post, they may click the "Like" button on that post. In the part of hard disk, we can find clicking "Like" evidence by searching the keyword "likes".

In the part of memory, we can also find clicking "Like" evidence by searching the keyword "likes". As a result, by analyzing the clicking "Like" evidence, the investigator can easily realize the preference of a perpetrator.

4.1.6 Friend List and Friend Request

In the hard, we can find friends request evidence. When we search the keyword "invite-sent", we can see that there is a key string, for example, "https://www.linkedin.com/mynetwork/invite-sent/jing-you-lin-a9017b15b/?isSendInvite=true", as shown in Figure 8. The string "jing-you-lin-a9017b15b" is friend's personal ID. The string "?isSendInvite=true" represents that the user must have sent a friend request to other LinkedIn members in the past. Therefore, we can easily realize that the majority of friend-request format is often built in the form of "https://www.linkedin.com/mynetwork/invite-sent/Personal-ID/?isSendInvite=true". By searching the key string "https://www.linkedin.com/mynetwork/invite-sent/", we could easily understand whether the user had sent a friend request to other LinkedIn members or not. However, when we searched the key

string "https://www.linkedin.com/mynetwork/invite-connect/connections/" in the hard disk, we cannot find out the friend list.

In the part of memory, we can also find friend request evidence by searching the key string "https://www.linkedin.com/mynetwork/invite-sent/". However, we cannot find out friend list by searching the key string "https://www.linkedin.com/mynetwork/invite-connect/connections/" as well.

To sum up, the evidence we found in the memory is quite the same in the hard disk. In the memory, we also found login information, the evidence of writing a post, making comments, chatting with friends, clicking "Like" records and so on. Therefore, there is no difference between in the hard disk and in the memory that evidences we found on the Google Chrome browser.

4.2 Findings: Scenario 2: Mozilla Firefox

In scenario 2, we also aim to the hard disk and memory forensics. We did the same thing as the previous scenario did. However, the forensic target in this scenario is different from the previous scenario. In scenario 2, we did the experiment on the Mozilla Firefox browser.

4.2.1 Account and Password

In the Mozilla Firefox, we can find user account information by searching the key string "www.linkedin.com". We can also easily realize that the user must have been used this computer to perform LinkedIn activities. On the other hand, when we conduct a search for the password, we can find out password information by typing a user's password string directly.

In the part of memory, the situation is the same as in the hard disk. We can find login account and password information by searching the account string and password string directly.

4.2.2 Posting Evidence

As the same to the previous scenario, every posting has it's unique post ID. Post ID is a string of numbers. When a user writes a post on the feed, the system will automatically assign a unique ID to the posting. The posting network address is often built in the form of "https://www.linkedin.com/feed/update/urn:li:activity: Post ID". In the hard disk, by searching the key string "https://www.linkedin.com/feed/update/urn:li:activity:", we can find the evidence of post creation. Therefore, we can match the post ID we found in the image file and the network address. If both of them are the same, we can definitely infer that they must have posted that article on the LinkedIn website in the past.

In the part of memory, we can also find out the evidence of post creation by searching the key strings.

4.2.3 Making Comment Evidence

As the same to the previous scenario, LinkedIn allows any people to make any comments on any articles. In the hard disk, by searching the keyword "comment", we can find out comment evidence that we made on the other user's posting.

In the part of memory, the situation is the same as in the hard disk, we can also find comment evidence and its content by searching the keyword "comment".

4.2.4 Chatting Records

As the same to the previous scenario, the majority of chatting record network address is often built in the form of "https://www.linkedin.com/messaging/thread/Chat-ID". Therefore, by searching the key string "https://www.linkedin.com/messaging/thread/", we can easily find out the chatting record evidence as well. Furthermore, we can also find out chatting content evidence by looking for the key string in reverse searching.

In the part of memory, we can also find out chatting record evidence by searching the key string "https://www.linkedin.com/messaging/thread/", and find chatting content evidence by looking for the key string in reverse searching.

4.2.5 Clicking "Like" Button Evidence

As the same to the previous scenario, we can find out clicking "Like" evidence by searching the keyword "likes" in the hard disk.

In the part of memory, we can also find out clicking "Like" evidence by searching the keyword "likes". As a result, by analyzing the clicking "Like" evidence, the investigator can easily realize the preference of a perpetrator.

4.2.6 Friend List and Friend Request

As the same to the previous scenario, we can find out friend request evidence in the hard disk. The majority of friend-request format is often built in the form of "https://www.linkedin.com/mynetwork/invite-sent/Personal-ID/?isSendInvite=true". By searching the key string "https://www.linkedin.com/mynetwork/invite-sent/", we could easily understand whether the user had sent a friend request to other LinkedIn members or not. However, when we searched the key string "https://www.linkedin.com/mynetwork/invite-connect/connections/" in the hard disk, we cannot find out the friend list as well.

In the part of memory, we can also find out friend request evidence by searching the key string "https://www.linkedin.com/mynetwork/invite-sent/". However, we cannot find out friend list by searching the key string "https://www.linkedin.com/mynetwork/invite-connect/connections/" as well.

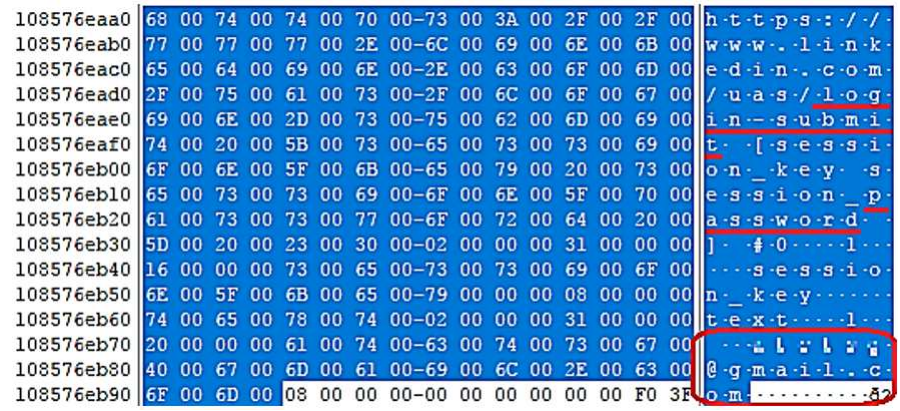


Figure 1: The result of searching login information

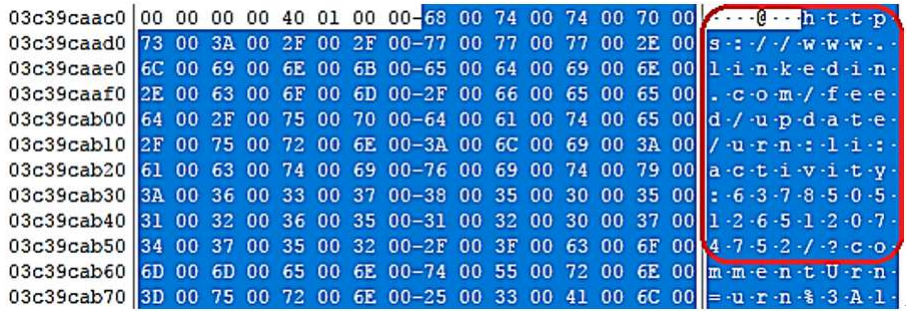


Figure 2: The evidence of post creation was found

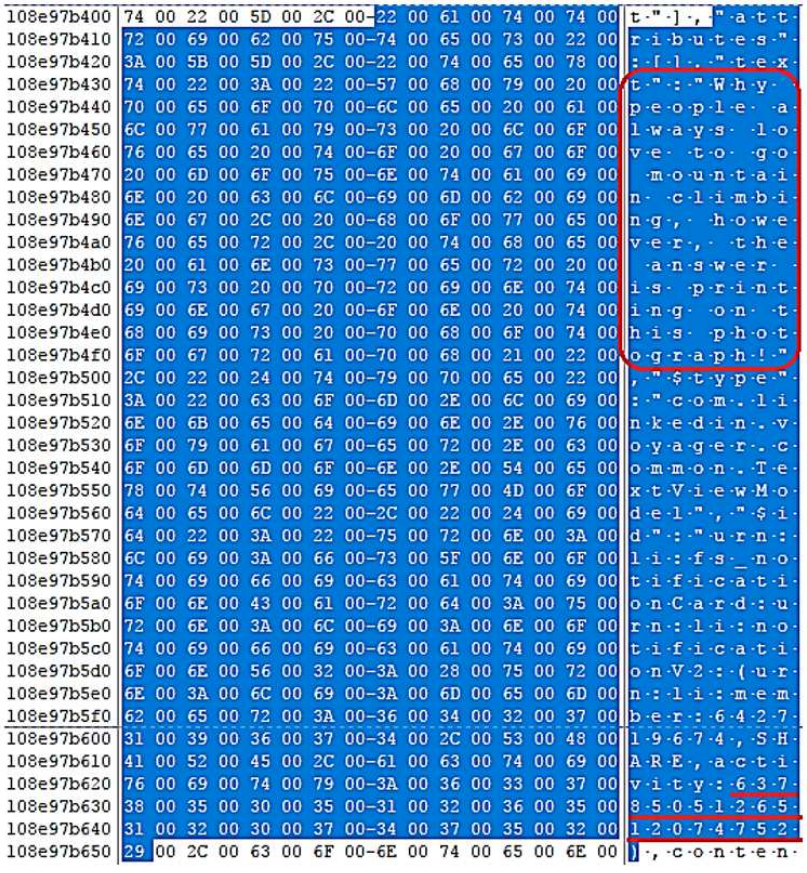


Figure 3: Post content was found by searching key string

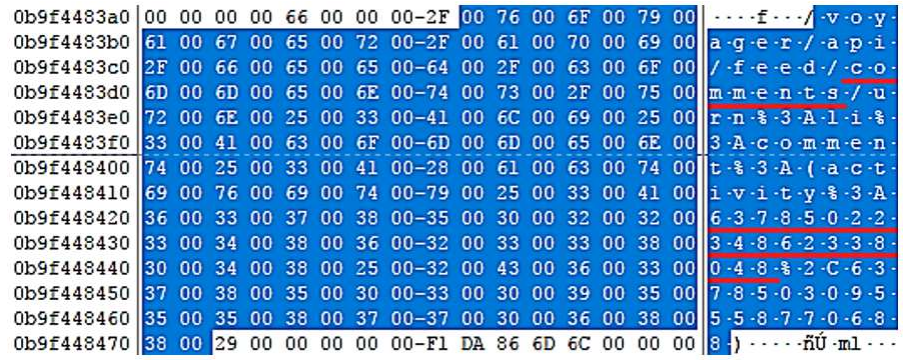


Figure 4: The evidence of comment creation was found by searching keyword "comment"

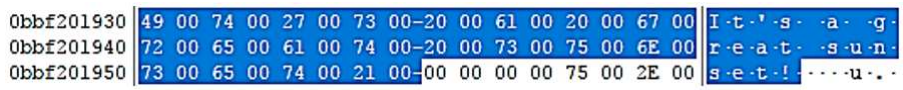


Figure 5: Post content was found by searching key string

Figure 6: The evidence of chat record

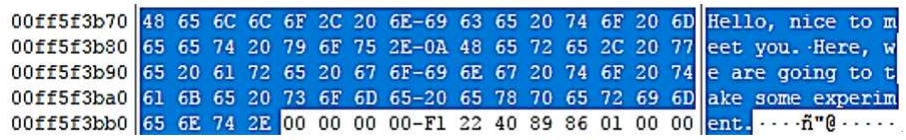


Figure 7: The evidence of chat content

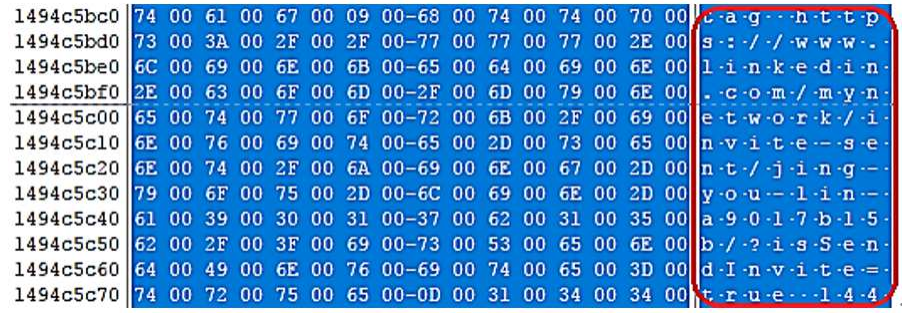


Figure 8: The evidence of friend request

To sum up, the evidence we found in the memory is quite the same in the hard disk. In the part of memory, we also found login information, the evidence of post creation, making comments, chatting records, clicking "Like" records and so on. Therefore, there is no difference between in the hard disk and in the memory that evidences we found on the Mozilla Firefox browser.

4.3 Findings: Scenario 3: Microsoft Edge

In scenario 3, we also aim to the hard disk and memory forensics. We did the same thing as the previous scenarios did. However, the forensic target in this scenario is different from the previous two scenarios. In scenario 3, we did the experiment on the Microsoft Edge browser.

4.3.1 Account and Password

In Microsoft Edge, we can find user account and password information by looking for the string in reverse searching. By typing account string and password string, we can see there is a key string "www.linkedin.com" in the context. Therefore, we can easily realize that the user must have been used this computer to perform LinkedIn activities.

In the part of memory, the situation is the same as in the hard disk. We can find login account and password information by searching account string and password string directly.

4.3.2 Posting Evidence

As the same to the previous scenarios, every posting has it's unique post ID. Post ID is a string of numbers. When a user writes a post on the feed, the system will automatically assign a unique ID to the posting. The posting network address is often built in the form of "https://www.linkedin.com/feed/update/urn:li:activity: Post ID". In the hard disk, by searching the key string "https://www.linkedin.com/feed/update/urn:li:activity:", we can find the evidence of post creation. Therefore, we can match the post ID we found in the image file and the network address. If both of them are the same, we can definitely infer that they must have posted that article on the LinkedIn website in the past. Furthermore, we can find the content of posting by inverse searching.

In the part of memory, we can also find out the evidence of post creation and its contents by searching the key strings.

4.3.3 Making Comment Evidence

As the same to the previous scenarios, LinkedIn allows any people to make any comments on any articles. In the hard disk, by searching the keyword "comment", we can find out comment evidence that we made on the other user's posting.

In the part of memory, the situation is the same as in the hard disk, we can also find comment evidence and its content by searching the keyword "comment".

4.3.4 Chatting Records

As the same to the previous scenarios, the majority of chatting record network address is often built in the form of "https://www.linkedin.com/messaging/thread/Chat-ID". Therefore, by searching the key string "https://www.linkedin.com/messaging/thread/", we can easily find out the chatting record evidence as well. Furthermore, we can also find chatting content by looking for the key string in reverse searching.

In the part of memory, we can also find out chatting record evidence by searching the key string "https://www.linkedin.com/messaging/thread/", and find chatting content by looking for the key string in reverse searching.

4.3.5 Clicking "Like" Button Evidence

As the same to the previous scenarios, we can find out clicking "Like" evidence by searching the keyword "likes" no matter in the hard disk or in the memory.

4.3.6 Friend List and Friend Request

As the same to the previous scenarios, we can find out friend request evidence in the hard disk. The majority of friend-request format is often built in the form of "https://www.linkedin.com/mynetwork/invite-sent/Personal-ID/?isSendInvite=true". By searching the key string "https://www.linkedin.com/mynetwork/invite-sent/", we could easily understand whether the user had sent a friend request to other LinkedIn members or not. However, when we searched the key string "https://www.linkedin.com/mynetwork/invite-connect/connections/" in the hard disk, we cannot find out the friend list as well.

In the part of memory, we can also find out friend request evidence by searching the key string "https://www.linkedin.com/mynetwork/invite-sent/". However, we cannot find out friend list by searching the key string "https://www.linkedin.com/mynetwork/invite-connect/connections/" as well. To sum up, the evidence we found in the memory is quite the same in the hard disk. In the part of memory, we also found login information, the evidence of post creation, making comments, chatting records, clicking "Like" records and so on. Therefore, there is no difference between in the hard disk and in the memory that evidences we found on the Microsoft Edge browser.

4.4 Experiment Comparison

After we conducted these three scenarios, we drew a table to clearly comparing the difference between them. As shown in Table 1, we can realize that there is no difference between them. No matter the evidence stored in the hard disk or in the memory, the evidence we can find in the Google Chrome, in the Mozilla Firefox or in the Microsoft Edge were the same. Moreover, all the searching keywords

Table 1: The comparison of findings between browsers

Category / Activity	Google Chrome		Mozilla Firefox		Microsoft Edge	
	Hard Disk	Memory	Hard Disk	Memory	Hard Disk	Memory
Account	O	O	O	O	O	O
Password	—	—	O	O	O	O
Post evidence	O	O	O	O	O	O
Make comment evidence	O	O	O	O	O	O
Click "Like" button evidence	O	O	O	O	O	O
Chat records	O	O	O	O	O	O
Chat contents	O	O	O	O	O	O
Friend list	—	—	—	—	—	—
Friend request	O	O	O	O	O	O

O: Found —: None

or key strings are the same in the hard disk as compared in the memory. Therefore, the majority of evidence can be found in the hard disk and in the memory.

5 Conclusions

Nowadays, thanks to the rapid development of new technologies, thousands of new social networking sites have sprung up over the past few years, such as Facebook, Twitter, Instagram, and so on. However, there are still some problems we should concern, that is, various kinds of cybercrime emerge endlessly in recent years. In order to assist investigators to investigate cybercrimes, this paper proposes a forensic way to investigate a perpetrator who commits a crime via the LinkedIn social networking site on the computer. We did a series of user activities that users may operate it. All of these behaviors were conducted respectively on three different browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge. Moreover, these three different browsers were conducted respectively on three different clean computers.

After completing these procedures, we adopt a forensic tool called FTK Imager to create an image file for the hard disk. On the other hand, we adopt the MANDIANT tool to create an image file for the memory. Thereafter, in order not to influence the integrity of digital evidence, we make use of FTK Imager to analyze image files on the other clean computer. In our experiment, we can find many kinds of evidences, for example, post creation, comment creation, browsing evidence, chatting records, clicking the "Like" button on the other postings and so forth. Finally, we compare our findings between these three different browsers, as shown in Table 1.

All of the findings could be used for cybercrime investigation. The investigators can analyze preference or daily activities of a perpetrator based on important information. Furthermore, if computer crime happened, all of the evidences extracted and analyzed by the investigator could be a crucial admission on the court.

References

[1] A. Abhyankar, "Social networking sites," *SAMVAD*, vol. 2, pp. 28–21, 2011.

[2] Alexa, *Linkedin [Online]*, Apr. 2018. (`https://www.alexa.com/siteinfo/linkedin.com`)

[3] H. Arshad, A. Jantan, and E. Omolara, "Evidence collection and forensics on social networks: Research challenges and directions," *Digital Investigation*, vol. 28, pp. 126–138, 2019.

[4] K. K. Arthur and H. S. Venter, "An investigation into computer forensic tools," in *Proceedings of the ISSA Enabling Tomorrow Conference*, pp. 1–11, 2004.

[5] A. Azfar, K. K. R. Choo, and L. Lin, "An android social app forensics adversary model," in *The 49th Hawaii International Conference on System Sciences (HICSS'16)*, pp. 5597–5606, 2016.

[6] D. Boyd and N. Ellison, "Social network sites: Definition, history, and scholarship," *IEEE Engineering Management Review*, vol. 38, no. 3, pp. 16–31, 2010.

[7] Criminal Investigation Bureau, *2017 Cybercrime overview [Online]*, 2017. (`https://www.cib.gov.tw/Crime/Detail/981`)

[8] F. N. Dezfouli, B. Eterovic-Soric A. Dehghantanha, and K. K. R. Choo, "Investigating social networking applications on smartphones detecting Facebook, Twitter, Linkedin and Google+ Artefacts on Android and Ios platforms," *Australian Journal of Forensic Sciences*, vol. 48, pp. 469–488, 2016.

[9] C. Dwyer, S. Hiltz, and K. Passerini, "Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace," in *Americas Conference on Information Systems (AMCIS'07)*, 2007. (`https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1849&context=amcis2007`)

[10] eBizMBA, *Top 15 Most Popular Social Networking Sites [Online]*, 2019. (`http://www.ebizmba.com/articles/social-networking-websites`)

[11] K. Jaishankar, "Cyber criminology as an academic discipline: History, contribution and impact," *Inter-*

national Journal of Cyber Criminology, vol. 12, no. 1, pp. 1–8, 2018.

[12] LinkedIn, *About US Statistics [online]*, 2018. (https://news.linkedin.com/about-us\#statistics)

[13] C. D. Marcum and G. E. Higgins, *Social Networking as a Criminal Enterprise*, pp. 49–144, 2014.

[14] N. A. Mutawa, I. Baggili, and A. Marrington, "Forensic analysis of social networking applications on mobile devices," *Digital Investigation*, vol. 9, pp. 24–33, 2012.

[15] Ministry of the Interior Republic of China National Police Agency, *The Cybercrime Rate in January to June, 2017 [online]*, 2017. (https://wsww.npa.gov.tw/NPAGip/wSite/ct?xItem=86451\&ctNode=12594\&mp=1)

[16] S. T. Neha, *Forensic analysis of WhatsApp on Android smartphones (master's thesis)*, 2013. (https://scholarworks.uno.edu/cgi/viewcontent.cgi?article=2736&context=td)

[17] Account Restricted, *Linkedin Help Center [Online]*, 2018. (https://www.linkedin.com/help/linkedin?lang=en)

[18] P. Stephenson, "The right tools for the job," *Digital Investigation*, vol. 1, no. 1, pp. 24–27, 2004.

[19] Wikipedia, *Linkedin [Online]*, 2019. (https://en.wikipedia.org/wiki/LinkedIn)

Biography

Ming Sang Chang received the Ph.D. degree from National Chiao Tung University, Taiwan, in 1999. In 2001 he joined the faculty of the Department of Information Management, Central Police University, where he is now a Professor. His research interest includes Computer Networking, Network Security, Digital Investigation, and Social Networks.

Chih Ping Yen is an Associate Professor, Department of Information Management, Central Police University. Received his Ph.D. degree from Department of Computer Science and Information Engineering, National Central University, Taiwan, in 2014. His research interest includes Digital Investigation, Artificial Intelligence & Pattern Recognition, Image Processing, and Management Information Systems.

Run-based Modular Reduction Method

Zhengjun Cao[1], Zhen Chen[1], Ruizhong Wei[2], and Lihua Liu[3]
(Corresponding author: Zhengjun Cao)
Department of Mathematics, Shanghai University, No. 99, Shangda Road, 200444, Shanghai, China[1]
Department of Computer Sciences, Lakehead University, 955, Oliver Road, Thunder Bay, Canada[2]
Department of Mathematics, Shanghai Maritime University, No.1550, Haigang Ave, Shanghai, China[3]
(Email: caozhj@shu.edu.cn)
(Received Oct. 19, 2018; Revised and Accepted Feb. 7, 2019; First Online June 11, 2019)

Abstract

The existing lookup-table modular reduction methods partition the binary string of an integer into fixed-length blocks such as 32 bits or 64 bits. This approach requires a fixed amount of looking up tables. In this paper, we introduce a new modular reduction method which partitions the binary string of an integer into blocks according to its runs. The new method can efficiently reduce the amount of looking up tables. Its complexity depends essentially on the amount of runs or 1's in the left segment of the binary string of an integer to be reduced. We show that the new reduction is almost twice as fast as the popular Barrett's reduction.

Keywords: Barrett's Reduction; Montgomery's Reduction; Run-based Modular Reduction

1 Introduction

The performance of public key cryptographic schemes depends heavily on the speed of modular reduction. Among several modular reduction algorithms, Montogomery's reduction and Barrett's reduction are more competitive. In 1985, P. Montgomery [13] invented an elegant reduction method. His method is not efficient for a single modular multiplication, but can be used effectively in computations where many multiplications are performed for given inputs.

At Crypto'86, P. Barrett [1] proposed a novel reduction method which is applicable when many reductions are performed with a single modulus. Barrett's reduction and Montgomery's reduction are similar in that expensive divisions in classical reduction are replaced by less-expensive multiplications. At Crypto'93, A. Bosselaers *et al.* [2] compared the performances of classical algorithm, Barrett's algorithm and Montgomery's algorithm. It is reported that these algorithms all have their specific behavior resulting in a specific field of application. No single algorithm is able to meet all demands. In 1998, Win *et al.* [18] reported that the difference between Montgomery's and Barrett's reduction was negligible in their implementation on an Intel Pentium Pro of field arithmetic in $\mathbb{F}_p$ for a 192-bit prime p. In 2011, Dupaquis and Venelli [4] modified Barrett's reduction and Montgomery's reduction. Their technique allows the use of redundant modular arithmetic. The proposed redundant Barrett's reduction algorithm can be used to strengthen the differential side-channel resistance of asymmetric cryptosystems.

In order to further speed up modular reduction, lookup table has been adopted by several researchers [7–9, 12, 14, 15, 17]. If the size of a pre-computed table is manageable, the method is very effective. These reduction methods partition the binary string of an integer into fixed-length blocks such as 32 bits or 64 bits. This approach requires a moderate size table. In 1997, Lim *et al.* [10] experimented on Montgomery's reduction, classical reduction, Barrett's reduction and some reduction algorithms using lookup table. It reported that the proposed lookup-table method runs almost two to three times faster on a workstation than the Montgomery's reduction. Although the experimental results are interesting, they did not present a complexity analysis of these combined lookup table methods.

The principles of the existing reduction algorithms can be briefly summarized as following:

- *Division.* The classical reduction algorithm adopts this principle.
- *Multiplication.* Both Barrett's and Montgomery's reduction adopt this principle.
- *Addition* [look up table according to *fixed-length blocks*]. All current reduction algorithms based on this principle look up table according to fixed-length blocks such as 32 bits or 64 bits.

The last principle, intuitively, is more applicable because it totally eliminates multiplications although it requires a moderate size table and a fixed amount of looking up tables. However, it seems that they are not suitable for small devices [5, 11, 16] such as smart phones.

In this paper, we put forth a new reduction method based on the principle of *addition* [look up table according to *runs*]. Unlike the traditional lookup-table reduction, the proposed method partitions the binary string

of an integer into blocks according to its runs instead of fixed-length blocks. The performance of the new method depends essentially on the amount of runs or 1's in the left segment of the binary string of an integer to be reduced. The new method can efficiently reduce the amount of looking up tables. We also provide a thorough complexity analysis of the method.

2 Related Reduction Methods

2.1 Montgomery's Reduction

Let $R > p$ with $\gcd(R, p) = 1$. The method produces $zR^{-1} \bmod p$ for an input $z < pR$. If $p' = -p^{-1} \bmod R$, then $c = zR^{-1} \bmod p$ can be obtained via

$$c \leftarrow (z + (zp' \bmod R)p)/R, \text{ if } c \geq p \text{ then } c \leftarrow c - p.$$

Given $x \in [0, p)$, let $\tilde{x} = xR \bmod p$. Define $\text{Mont}(\tilde{x}, \tilde{y}) = (\tilde{x}\tilde{y})R^{-1} \bmod p = (xy)R \bmod p$. The transformations $x \mapsto \tilde{x} = xR \bmod p$, and $\tilde{x} \mapsto \tilde{x}R^{-1} \bmod p = x$ are performed only once when they are used as a part of a larger calculation such as modular exponentiation.

2.2 Barrett's Reduction

The following description of Barrett's reduction comes from [6], which calculates $z \bmod p$. The algorithm first selects a suitable base b (e.g., $b = 2^L$ where L is near the word size of the processor). It then calculates $\mu = \lfloor \frac{b^{2k}}{p} \rfloor$, where $k = \lfloor \log_b p \rfloor + 1$. Suppose $0 \leq z < b^{2k}$. Let $q = \lfloor \frac{z}{p} \rfloor$, $r = z \bmod p = z - q\,p$. Since $\frac{z}{p} = \frac{z}{b^{k-1}} \cdot \frac{b^{2k}}{p} \cdot \frac{1}{b^{k+1}}$, we have

$$0 \leq \hat{q} = \left\lfloor \frac{\lfloor \frac{z}{b^{k-1}} \rfloor \cdot \mu}{b^{k+1}} \right\rfloor \leq \left\lfloor \frac{z}{p} \right\rfloor = q.$$

If μ is computed in advance, then the main cost of calculating $\hat{q}$ consists of one multiplication and two types of bit operations for $\lfloor \frac{z}{b^{k-1}} \rfloor$ and $\lfloor \frac{y}{b^{k+1}} \rfloor$, where $y = \lfloor \frac{z}{b^{k-1}} \rfloor \cdot \mu$. Set $\alpha = \frac{z}{b^{k-1}} - \lfloor \frac{z}{b^{k-1}} \rfloor$, $\beta = \frac{b^{2k}}{p} - \left\lfloor \frac{b^{2k}}{p} \right\rfloor$. Then $0 \leq \alpha, \beta < 1$ and

$$q = \left\lfloor \frac{\left(\lfloor \frac{z}{b^{k-1}} \rfloor + \alpha\right)\left(\left\lfloor \frac{b^{2k}}{p} \right\rfloor + \beta\right)}{b^{k+1}} \right\rfloor \leq \left\lfloor \frac{\lfloor \frac{z}{b^{k-1}} \rfloor \cdot \mu}{b^{k+1}} + \frac{\lfloor \frac{z}{b^{k-1}} \rfloor + \left\lfloor \frac{b^{2k}}{p} \right\rfloor + 1}{b^{k+1}} \right\rfloor$$

Since $0 \leq z < b^{2k}$ and $b^{k-1} \leq p < b^k$, we have

$$\left\lfloor \frac{z}{b^{k-1}} \right\rfloor + \left\lfloor \frac{b^{2k}}{p} \right\rfloor + 1 \leq (b^{k+1} - 1) + b^{k+1} + 1 = 2b^{k+1}$$

$$q \leq \left\lfloor \frac{\lfloor \frac{z}{b^{k-1}} \rfloor \cdot \mu}{b^{k+1}} + 2 \right\rfloor = \hat{q} + 2.$$

Therefore, we obtain $\hat{q} \leq q \leq \hat{q} + 2$. Set $\hat{r} = z - \hat{q}\,p$. We get $r = \hat{r} + (\hat{q} - q)p$. That is, at most two subtractions are required to obtain r using $\hat{r}$.

In 2014, Cao and Wu [3] pointed out that the formula

$$\frac{z}{p} = \frac{z}{b^{k-1}} \cdot \frac{b^{2k}}{p} \cdot \frac{1}{b^{k+1}}$$

can be directly replaced with

$$\frac{z}{p} = \frac{z}{2^k} \cdot \frac{2^{2k}}{p} \cdot \frac{1}{2^k}$$

The adaption could further optimize the programming code and solve the data expansion problem in Barrett's reduction.

2.3 Lookup-Table Reduction

Suppose that z and n are two integers, $b^{k-1} \leq n < b^k$, $0 \leq z < b^{2k}$ where $b = 2^L$ is a suitable base. To compute $z \bmod n$, the usual lookup-table reduction computes

$$z = \sum_{j=0}^{k-1} z_j b^j + \sum_{i=0}^{k-1} z_{k+i} A[i] \bmod n, \tag{1}$$

where $0 \leq z_j < b, j = 0, \cdots, 2k-1$, $A[i] = b^{k+i} \bmod n\,(0 \leq i \leq k-1)$ are computed and stored in advance. In 1997, Lim *et al.* [10] suggested taking $b = 2^{32}$. In this method, it only requires a storage for 624 values of modulus size (e.g., about 78 Kbytes for $|n| = 1024$). They experimented on Montgomery's reduction, classical reduction, Barrett reduction and some lookup-table reduction algorithms. It reported that:

1) Modular reduction takes considerably more time than multiplication;
2) Montgomery's algorithm and the combined table lookup method give almost the same performance;
3) The proposed table lookup methods (L224, L624, L1696) run almost two to three times faster on a workstation than Montgomery's reduction. These methods, however, do not give much improvement on a PC.

3 Basic Lookup-Table Reduction

The idea behind the basic lookup-table modular reduction is naive, but useful in some cases. We now describe it as follows.

3.1 Pre-computed Table

Given a positive integer n, choose an integer k such that $2^{k-1} < n < 2^k$. The pre-computed table are constructed as following (see Table 1).

We can specify that $|r[\ell]| \leq \lfloor n/2 \rfloor$, $\ell = k, \cdots, 2k-1$. The size of the pre-computation table $\mathbb{T}$ can be further reduced because $r[i+1] = 2\,r[i]$ for some indexes i.

Table 1: Pre-computation table $\mathbb{T}$ for a modular n

ℓ	$2k-1$	$2k-2$	$\cdots$	k
$r[\ell]$	$2^{2k-1} \bmod n$	$2^{2k-2} \bmod n$	$\cdots$	$2^k \bmod n$

3.2 Basic Method (Method-1)

Denote the binary string of a positive integer z by Binary(z). Suppose that $0 \le z < 2^{2k}$. We directly set the base $b = 2$ in Equation (1). It follows that

$$z \equiv \sum_{i=0}^{k-1} z_{k+i}\, r[k+i] + \sum_{j=0}^{k-1} z_j\, 2^j \bmod n, \tag{2}$$

where $z_j \in \{0,1\}, j = 0, \cdots, 2k-1$, $r[k+i] = 2^{k+i} \bmod n\,(0 \le i \le k-1)$. Since $z_{k+i} \in \{0,1\}$, $0 \le i \le k-1$, we completely eliminated multiplications.

Example 1. $n = 97 = (1100001)_2, k = 7$ *(bit-length)*, $z = 3135 = (110000111111)_2, l = 12$. *Look up for the values* $r[11] = 2^{11} \bmod n = 11$ *and* $r[10] = 2^{10} \bmod n = 54$. *It gives* $z = 3135 \equiv r[11] + r[10] + (111111)_2 = 11 + 54 + 63 \equiv 31 \bmod 97$.

3.3 Cost Analysis

The number of additions in this method depends on the amount of 1's in the left segment of Binary(z). On average, there are about $\lfloor k/2 \rfloor$ 1's in the left segment if the bit-length of z is $2k$. That means it requires $\lfloor k/2 \rfloor$ additions of k-bit integers to compute $r = \sum_{j=0}^{k-1} z_j 2^j + \sum_{i=0}^{k-1} z_{k+i} r[k+i]$. It is expected that the absolute value $|r| < \frac{kn}{4}$, since $|r[\ell]| \le \lfloor n/2 \rfloor$. Hence, it requires $\lfloor k/4 \rfloor$ subtractions to compute $r \bmod n$. In total, Method-1 requires the cost of performing $\lfloor \frac{3k}{4} \rfloor$ additions of k-bit integers.

In the method, addition happened for all values $r[k+i]$ corresponding to $z_{k+i} = 1\,(0 \le i \le k-1)$. In the worst case, $z_k = z_{k+1} = \cdots = z_{2k-1} = 1$, it has to look up table and do addition k times. Clearly, Method-1 is inappropriate for this case.

4 Run-based Reduction

The Method-1 is not good for the worst case when there is only one run of 1's in the left segment of Binary(z), *i.e.*, all the positions are 1's. We now introduce a new reduction method based on lookup table which is much better for the above case.

4.1 The Basic Idea

Given two positive integers k, n, where $2^{k-1} < n < 2^k$, and a positive integer z satisfying $0 \le z < 2^{2k}$, set $\ell_0 = \lfloor \log_2 z \rfloor + 1$. Flipping all bits of z, we obtain the integer z_1 such that $z = (2^{\ell_0} - 1) - z_1$. Set $\ell_1 = \lfloor \log_2 z_1 \rfloor + 1$. Flipping all bits of z_1, we obtain the integer z_2 such that $z = (2^{\ell_0} - 1) - (2^{\ell_1} - 1) + z_2$. By the same procedure, we shall get

$$\begin{aligned} z =& (2^{\ell_0} - 1) - (2^{\ell_1} - 1) + (2^{\ell_2} - 1) + \cdots \\ &+ (-1)^{j-1}(2^{\ell_{j-1}} - 1) + (-1)^j z', \end{aligned} \tag{3}$$

where $\ell_{j-1} > k \ge \ell_j$, ℓ_j is the bit-length of z'. Clearly,

$$\ell_0 > \ell_1 > \cdots > \ell_j. \tag{4}$$

We then look up the pre-computed table for values $r[\ell_0], \cdots, r[\ell_{j-1}]$ using the indexes $\ell_0, \cdots, \ell_{j-1}$ and compute

$$\begin{aligned} r =& (r[\ell_0] - 1) - (r[\ell_1] - 1) + (r[\ell_2] - 1) + \cdots \\ &+ (-1)^{j-1}(r[\ell_{j-1}] - 1) + (-1)^j z'. \end{aligned} \tag{5}$$

Thus, $z \equiv r \bmod n$.

4.2 Description of Method-2

To obtain indexes $\ell_0, \cdots, \ell_{j-1}$ and z' in Equation (5), the above procedure requires to flip all bits of strings. In fact, these indexes and z' depend essentially on the runs in the left segment of Binary(z). Here a run means a maximal substring whose bit positions all contain the same digit 0 or 1. We can obtain them by counting the length of each run in the left segment. Suppose that

$$\text{Binary}(z) = \alpha_0 || \alpha_1 || \cdots || \alpha_{j-1} || \alpha'_j, \tag{6}$$

where the notation $a||b$ means that string a is concatenated with string b, and α_i $(0 \le i \le j-1)$ are runs with lengths d_i respectively, α'_j is the remaining string. We have

$$\begin{aligned} \ell_1 =& \ell_0 - d_0, \ \cdots, \ \ell_{j-1} = \ell_{j-2} - d_{j-2}, \\ \ell_j =& \ell_{j-1} - d_{j-1} \end{aligned} \tag{7}$$

where $\ell_j \le k < \ell_{j-1}$. Note that the length of string α'_j is ℓ_j. Hence, we get

$$z' = \begin{cases} (\alpha'_j)_2, & j \text{ is even}, \\ 2^{\ell_j} - 1 - (\alpha'_j)_2, & j \text{ is odd}, \end{cases}$$

Thus,

$$z \equiv \begin{cases} r[\ell_0] & -r[\ell_1] + r[\ell_2] + \cdots + (-1)^{j-1} r[\ell_{j-1}] \\ & +(\alpha'_j)_2, j \text{ is even}, \end{cases} \tag{8}$$

$$z \equiv \begin{cases} r[\ell_0] & -r[\ell_1] + r[\ell_2] + \cdots + (-1)^{j-1} r[\ell_{j-1}] \\ & +(\alpha'_j)_2 - 2^{\ell_j}, j \text{ is odd}, \end{cases} \tag{9}$$

Example 2. $n = 97 = (1100001)_2, k = 7; z = 3135 = (110000111111)_2, \ell_0 = 12$. *The runs in the left segment of Binary*(z) *are* $\alpha_0 = 11, \alpha_1 = 0000$. *Their lengthes are* $d_0 = 2, d_1 = 4$. *We have* $\ell_1 = \ell_0 - d_0 = 12 - 2 = 10, \ell_2 = \ell_1 - d_1 = 10 - 4 = 6$. *Since* $\ell_2 = 6 < 7 = k$, *we get* $j = 2$, $\alpha'_2 = 111111$. *Therefore,* $z' = (\alpha')_2 = (111111)_2 = 63$. *Thus,* $z = 2^{\ell_0} - 2^{\ell_1} + z' = 2^{12} - 2^{10} + 63 \equiv 22 - 54 + 63 = 31 \bmod 97$.

Table 2: The algorithm for Method-2

INPUT: $n, k = \text{BitLength}(n), 0 \le z < 2^{2k}$, and $\mathbb{T} = \{r[2k-1], r[2k-2], \cdots, r[k]\}$.
OUTPUT: $z \bmod n$.
If $z < n$, then return z.
If $\text{BitLength}(z) = k$, then return $z - n$.
$s \leftarrow \text{Binary}[z]$, $\ell \leftarrow \text{BitLength}[z]$, $y \leftarrow 1$, $r \leftarrow r[\ell]$, $d \leftarrow 0$, $t \leftarrow 0$.
For i from $\ell - 1$ downto 0 do
$b \leftarrow \text{StringTake}[s, \{i\}]$.
If $b = y$, then $d \leftarrow d + 1$.
$\ell \leftarrow \ell - d$, $t \leftarrow t + 1$, $r \leftarrow r + (-1)^t r[\ell]$.
If $\ell > k$, then $y \leftarrow \text{Mod}(y + 1, 2)$, $d \leftarrow 0$.
$\alpha \leftarrow \text{StringTake}\ [s, -\ell]$.
If $\text{Mod}\,(t, 2) = 0$, then $r \leftarrow r + (\alpha)_2$, else $r \leftarrow r + (\alpha)_2 - 2^\ell$.
Break.
While $r \ge n$ do: $r \leftarrow r - n$.
While $r < 0$ do: $r \leftarrow r + n$.
Return r.

4.3 Complexity Analysis

To obtain $\ell_0, \cdots, \ell_{j-1}, z'$, it requires only a handful of less-expensive bit operations. Since $\ell_0, \cdots, \ell_{j-1}$ is ordered, *i.e.*, $\ell_0 > \ell_1 > \cdots > \ell_{j-1}$, the cost of looking up $r[\ell_0], \cdots, r[\ell_{j-1}]$ in $\mathbb{T}$ is negligible. There are j additions for computing r. Since $|r[t]| \le \lfloor n/2 \rfloor, t \in \{\ell_0, \cdots, \ell_{j-1}\}$, we have

$$|r| \le (j+2)\lfloor n/2 \rfloor < \left(\left\lfloor \frac{j+2}{2} \right\rfloor + 1 \right) n.$$

That means it requires at most $\lfloor \frac{j+2}{2} \rfloor$ subtractions for computing $r \bmod n$. In total, the method needs to perform $\lfloor \frac{3j}{2} \rfloor$ additions of k-bit integers. We shall see that $j \approx \lfloor k/2 \rfloor$. That means Method-2 has the similar performance as Method-1.

We now give a comparison between Method-2 and Barrett's reduction. The computation of $\lfloor z/b^i \rfloor \cdot \mu$ dominates the cost of Barrett's reduction. It requires a multiplication. For convenience, we suppose that it is a multiplication of k-bit integers.

The Method-2 requires more cost for bit scans if the cost for one byte scan is considered to be approximately equal to that for one bit scan. But we here stress that the whole cost for bit scans is less than the cost for an addition of k-bit integers.

The quantity j is of great importance to the comparison. Clearly, $j \le k$. If the left segment of $\text{Binary}(z)$ is $\underbrace{1010\cdots10}_{k-\text{bit}}$, then $j = k$. Given a random $2k$-bit integer z, it is expected that there are about k runs and k 1's. Thus, we have $j = \lfloor k/2 \rfloor$. That means the new reduction is faster than Barrett's reduction at the expense of a little storage. The storage requirement in such case is acceptable to most devices at the time.

5 A Fast Reduction Method

As we mentioned previously, Method-1 is inappropriate for dealing with the string $11\cdots1$, whereas Method-2 can deal efficiently with such a string. Method-2 is not as efficient as Method-1 to deal with the string $1010\cdots10$. When hundreds of modular multiplications are required for modular exponentiation, it is better to use the two methods alternatively. Since they require a same pre-computed table, we can combine these two methods. We now present a description of such a combined reduction method.

5.1 A Combined Reduction Algorithm

Suppose that n is the modular, $0 \le z < 2^{2k}$, $k = \text{BitLength}(n)$ and $\mathbb{T}$ is the pre-computed table. To compute $z \bmod n$, the combined reduction method proceeds as follows.

1) Set Υ to be the left segment of $\text{Binary}(z)$ such that the length of the right segment equals to k.
2) Count the amount of 1's in Υ and denote it by ϕ.
3) Count the amount of runs in Υ and denote it by ψ.
4) If $\phi \le \psi$ then use Algorithm-1. Otherwise, use Algorithm-2.

5.2 Refined Algorithm

It is possible to refine the above algorithm. For example, considering a segment of $(101010111101)_2$. For this string, $\phi = 8$ and $\psi = 9$. So Algorithm-1 will be used. However, it is easy to see that the right part of the string is better to use Algorithm-2. So it is better to use Algorithm-1 for first 6 bits and use Algorithm-2 for last 6 bits. In general, if we have a long run of 1, then we should use Algorithm-2 for that run.

The following algorithm can be used to calculate $z \bmod n$, where $n < z < n^2$.

Table 3: Comparison between Barrett's reduction and Method-2

	arithmetic operation (k-bit integers)	pre-computation	byte/bit scans
Barrett's reduction	1 multiplication, 3 additions	value μ	$k/8$ byte
Method-2	$\lfloor \frac{3k}{4} \rfloor$ additions	table $\mathbb{T}$ (k items)	k bit

1) Set $\ell_0 = \text{BitLength}[z]$. Set Υ to be the left segment of Binary(z) such that the length of the right segment equals to k. Count the amount of 1's in Υ and denote it by ϕ. If $\phi \geq \lfloor k/2 \rfloor$, then flip all bits of Binary(z). Denote the new number by $\hat{z}$. Here $z = (2^{\ell_0} - 1) - \hat{z}$. In such case, the number of 1's in the corresponding left segment of $\hat{z}$ is less than $\lfloor k/2 \rfloor$. So, we consider $\hat{z} \bmod n$. For convenience, we now assume that $\phi \leq \lfloor k/2 \rfloor$.

2) Count runs in Υ to obtain $R = (l_0, r_0;\ l_1, r_1; \ldots; l_j, r_j)$, where l_0 is the length of the first run of 1 in Υ and r_0 is the length of the first run of 0 in Υ, $\ldots$, l_j is the length of the last run of 1 in Υ and r_j is the length of the last run of 0 in Υ. Here $l_i \geq 1$ for $0 \leq i \leq j$ and $r_i \geq 1$ for $0 \leq i \leq j-1$ while $r_j \geq 0$.

3) Let $\ell_t = k + \sum_{i=t}^{j}(l_i + r_i)$, $0 \leq t \leq j$. For t from 0 to j calculate S_t: if $l_t \leq 2$, $S_t = \sum_{m=\ell_t - l_t + 1}^{\ell_t} r[m-1]$; if $l_t > 2$, $S_t = r[\ell_t] - r[\ell_t - l_t]$.

4) Compute $LS = \sum_{t=0}^{j} S_t$ which can be used to calculate $z \bmod n$.

Note that the refined algorithm only needs to look up the pre-computation table $1 + \lfloor k/2 \rfloor$ times at most, *i.e.*, it requires about $\lfloor k/2 \rfloor$ additions of k-bit integers *at worst*. Since Barrett's reduction requires one multiplication of k-bit integers, the method is expected to be almost twice as fast as the Barrett's reduction.

Example 3. *Suppose* $z = 58809 = (1110010110111001)_2$, $n = 267 = (100001011)_2$. *Then* $k = 9$, $\Upsilon = (1110010)$, $R = (3, 2; 1, 1)$. *Therefore* $S_0 = r[16] - r[13] = 121 - 182 = -61, S_1 = r[10] = -44, LS = -61 - 44 = -105$. *So* $z = -105 + (110111001)_2 = -105 + 441 = 69 \bmod 267$.

6 Implementation Tips

Some experiments on modular reduction algorithms have been implemented, including the common lookup table reduction, the refined run-based reduction, Montgomery's reduction, Barret's reduction, the improved Barret reduction (see [3]) and the general repeated square reduction for the computation $c^d \bmod n$, where

$c =$ 55103280959622143570402130367663431846883890024225365746631236013125897340714776982730249289966488343996755920163957112016132956975401238007039707639868810208777108408089829058605678271696502129955757569123179449702471331787304364959839519775265074084061593327434500118603083495853207768231485190054148583981,

$d =$ 17970154009029862760662344073406083538245587958989134228820996621710832903953558853778906950976745158065143728393505657901184045798332028289815093774137325178448521127388065678503478658724581654937781809973937551742257916140835853898828972640247878231859992853336005115520383724262443403384025327820646467533,

$n =$ 13506641086599522334960321627880596993888147560566702752448514385152651060485953383394028715057190944179820728216447155137368041970396419174304649658927425623934102086438320211037295872576235850964311056407350150818751067659462920556368552947521350085287941637732853390610975054433499981115005697723689092756 3.

n is just the RSA-1024 number. The programming codes are written in Wolfram language. Nevertheless, their performances were not as expected strictly. It means the current high level languages cannot make the most of bit, byte or run scanning. That is to say, the underlying assembly language should be exploited for Montgomery's reduction, Barret's reduction and run-based reduction.

7 Conclusion

A new modular reduction method based on lookup table is introduced, which requires less arithmetic operations at the expense of a little storage. We show that the new reduction is almost twice as fast as Barrett's reduction. Interestingly, the method scans bit-by-bit. This feature makes it more portable and more suitable for small devices.

Acknowledgements

We thank the National Natural Science Foundation of China (Project 61411146001). The authors gratefully acknowledge the reviewers for their valuable suggestions.

References

[1] P. Barrett, "Implementing the rivest shamir and adleman public key encryption algorithm on a standard digital signal processor," in *Proceedings of 6th Annual Cryptology Conference, Advances in Cryptology (CRYPTO'86)*, pp. 311–323, Aug. 1987.

[2] A. Bosselaers, R. Govaerts, and J. Vandewalle, "Comparison of three modular reduction functions," in *Proceedings of 13th Annual Cryptology Conference, Advances in Cryptology (CRYPTO'93)*, pp. 175–186, Aug. 1993.

[3] Z. J. Cao and X. J. Wu, "An improvement of the barrett modular reduction algorithm," *International Journal of Computer Mathematics*, vol. 91, no. 9, pp. 1874–1879, 2014.

[4] V. Dupaquis and A. Venelli, "Redundant modular reduction algorithms," in *Proceedings of 10th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Applications (CARDIS'11)*, pp. 102–114, Sep. 2011.

[5] C. Guo, C. C. Chang, and S. C. Chang, "A secure and efficient mutual authentication and key agreement protocol with smart cards for wireless communications," *International Journal of Network Security*, vol. 20, no. 2, pp. 323–331, 2018.

[6] D. Hankerson., A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, 2004. (`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.394.3037&rep=rep1&type=pdf`)

[7] S. Hong, S. Oh, and H. Yoon, "New modular multiplication algorithms for fast modular exponentiation," in *Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT'96)*, pp. 166–177, May 1996.

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

[9] S. Kawamura and K. Hirano, "A fast modular arithmetic algorithm using a residue table," in *Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT'88)*, pp. 245–250, May 1988.

[10] C. Lim, H. Hwang, and P. Lee, "Fast modular reduction with precomputation," in *Proceedings of Korea-Japan Joint Workshop on Information Security and Cryptology (JW-ISC'97)*, pp. 65–79, Oct. 1997.

[11] Y. J. Liu, C. C. Chang, and S. C. Chang, "An efficient and secure smart card based password authentication scheme," *International Journal of Network Security*, vol. 19, no. 1, pp. 1–10, 2017.

[12] D. Mahto and D. K. Yadav, "Performance analysis of rsa and elliptic curve cryptography," *International Journal of Network Security*, vol. 20, no. 4, pp. 625–635, 2018.

[13] P. Montgomery, "Modular multiplication without trial division," *Mathematics of Computation*, no. 44, pp. 519–521, 1985.

[14] B. Parhami, "Analysis of tabular methods for modular reduction," in *Proceedings of 28th Asilomar Conference Signals, Systems, and Computers*, pp. 526–530, Nov. 1994.

[15] B. Parhami, "Modular reduction by multi-level table lookup," in *Proceedings of Midwest Symposium on Circuits and Systems (MWSCAS'97)*, pp. 381–384, Aug. 1997.

[16] C. Y. Tsai, C. Y. Yang, I. C. Lin, and M. S. Hwang, "A survey of e-book digital right management," *International Journal of Network Security*, vol. 20, no. 5, pp. 998–1004, 2018.

[17] C. Walter, "Faster modular multiplication by operand scaling," in *Proceedings of 11th Annual Cryptology Conference, Advances in Cryptology (CRYPTO'91)*, pp. 313–323, Aug. 1991.

[18] E. Win, S. Mister, B. Preneel, and M. Wiener, "On the performance of signature schemes based on elliptic curves," in *Proceedings of Algorithmic Number Theory*, pp. 252–266, June 1998.

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

Zhen Chen is currently pursuing his M.S. degree from Department of Mathematics, Shanghai university. His research interests include information security and cryptography.

Ruizhong Wei is a professor with the Department of Computer Science, Lakehead University, Canada. He received his Ph.D. degree in applied mathematics from Waterloo University. His research interests include combinatorics, algebraic code, algorithm design and analysis.

Lihua Liu is an associate professor with the Department of Mathematics, Shanghai Maritime University. She received her Ph.D. degree in applied mathematics from Shanghai Jiao Tong University. Her research interests include combinatorics and cryptography.

Anomaly Detection for Network Flow Using Immune Network and Density Peak

Yuanquan Shi[1,2] and Hong Shen[2]
(Corresponding author: Yuanquan Shi)
School of Computer Science and Engineering, Huaihua University[1]
Jinhai Road, Huaihua 418008, China
School of Computer Science, The University of Adelaide[2]
North Terrace, SA 5005, Australia
(Email: syuanquan@163.com)
(Received Oct. 29, 2018; Revised and Accepted May 17, 2019; First Online June 15, 2019)

Abstract

To identify effectively unknown malicious attack behaviors from massive network flows in Internet environment, an Anomaly Detection approach for network flow using Artificial Immune network and Density peak (ADAID) is proposed in this paper. In ADAID, we present an unsupervised clustering algorithm aiNet_DP combining artificial immune network (aiNet) and the clustering algorithm based on density peaks (CDP), where aiNet denotes a coarse-grained clustering algorithm to extract abstract internal images of network flows, CDP denotes a fine-grained clustering algorithm to obtain more precise cluster number and cluster centroids according to the clustering results of aiNet. The clustering labeling algorithm (CLA) and the flow anomaly detection algorithm (FAD) are introduced in ADAID to detect malicious attack behaviors of network flows, where CLA is used for labeling each cluster whether is malicious or not, and the labeled cluster is viewed as detector to identify anomaly network flows by using FAD. To evaluate the effectiveness of ADAID, the ISCX 2012 IDS dataset is used for simulating experiments. Compared with the anomaly detection approach which is based on the aiNet clustering and the aiNet based hierarchical clustering (aiNet_HC), respectively, the results show that ADAID is a radical anomaly detection approach and can achieve higher accuracy rates.

Keywords: Anomaly Detection; aiNet; Clustering Algorithm; Density Peak; Network Flow

1 Introduction

With the rapid development of information technologies and the universal application of electronic productions, network security problem has become severe society focus in our daily life. Nowadays, there are millions of network viruses and malicious attacks in different network environments, and many updated versions of them or novel attacks are produced constantly. The targets of network attacks mainly include network nodes, terminal computers, and smart devices, especially smartphone providing network admission and payment function [9]. To evaluate effectively cyberspace security, many security strategies are employed, such as private protection, firewall mechanism, virus defense, intrusion detection and risk evaluation *etc.*

Anomaly detection is one key component part of the intrusion detection system [11]. Up to now, anomaly detection strategy has been applied to many application areas, such as network security system, industrial control system, and Internet of Things *etc.* The merits of anomaly detection [2,25,27] can detect unknown malicious attacks from the captured network packets real-timely in network system environments. In traditional anomaly detection system, administrators firstly need define the legitimate profiles for the protected network system, the anomaly detection system will alarm if the detected network behaviors aren't normal. Due to the misuse detection strategy, another important intrusion detection method, holding the known malicious attack characteristic and the higher detection rates, some researchers have proposed the improved anomaly detection system combining with misuse detection technique to raise the detection rates (DRs) of known malicious attacks and decrease false alarm rates (FARs) of unknown attacks [3].

Compared with the packet anomaly detection, the flow anomaly detection analyzes network security problem by network flows, and it can solve some problems which are processing time and data reduction [23]. Network flow is viewed as a description approach of network behaviors based on the connections of network terminals and records high-level description of network connections, but network flow isn't real network packet [14]. Network flow is a bidirectional or unidirectional sequence

of packets traveling between two network terminals using network protocols (e.g. TCP/UDP) with common features [18]. The most important features of network flow include duration time, source/destination IP address, source/destination port number, and the transferred source/destination packets *etc.* The inherent rules of network flows can be analyzed by the common features of the sending/receiving protocol packets, especially TCP flows. At present, the flow anomaly detection has become a research hotspot, and meanwhile it is regarded as an effective complement of packet inspection [7, 8, 14].

As an important machine learning method, clustering analysis is applied widely to solve network security problem, especially detecting malicious attack behaviors from the massive network flows. Clustering analysis is aimed at classifying the given data elements into categories based on their similarity [22]. Clustering, an unsupervised classification approach, doesn't provide available labeled elements during training phase. The procedure of clustering analysis involves four basic stages [30]: Feature selection and extraction, clustering algorithm design and selection, clustering validation, results interpretation. Many researchers think that clustering holds the internal homogeneity and the external separation, *i.e.* elements in a cluster possessing similar pattern. The representative clustering techniques [30] include hierarchical clustering, partitional clustering, and evolutionary clustering *etc.* As one type of the most difficult and challenging problems in machine learning fields, many evolutionary clustering algorithms, such as artificial immune system, genetic algorithm and artificial neural network, are proposed successively to analyze the unsupervised nature problem, and the relevant data spatial distribution is unknown [4, 16, 31].

In this paper, an Anomaly Detection approach for network flow using Artificial Immune network and Density peak (ADAID) is proposed. To obtain more precise samples and cluster number from network flows, the aiNet [4] is used for coarse-grained clustering, and CDP [22] is adopted for fine-grained clustering according to the output results of coarse-grained clustering. To raise detection rates and decrease false alarm rates, we devise the CLA algorithm in this paper to label normal/abnormal clusters, and ISCX 2012 IDS dataset [26] is adopted to detect anomaly network flows. The mainly contributions of this paper include:

1) Propose an anomaly detection framework (ADAID), to detect malicious attack behaviors of network flows;
2) Propose an unsupervised clustering algorithm (aiNet_DP) combining artificial immune network and density peaks;
3) Propose a cluster labeling algorithm (CLA) to distinguish effectively benign and malicious behaviors of network flows.

The remainder of this paper is organized as follows. We describe a review of the prior researches on the unsupervised anomaly detection based on clustering algorithm and artificial immune network in Section 2. Section 3 describes the proposed ADAID approach based on artificial immune network and density peak for the anomaly detection of network flows. Section 4 illustrates the performance evaluations of ADAID on ISCX IDS dataset. The conclusion is finally given in the last Section.

2 Related Works

The clustering algorithms have been proposed to solve anomaly detection problems of network flow [2]. Portnoy *et al.* [20] proposed a variant of single-linkage clustering based on distance to classify data instances. Leung *et al.* [13] proposed the density-based and grid-based high dimensional clustering algorithm for unsupervised anomaly detection of large datasets. Petrovic *et al.* [19] combined the Davies-Bouldin index of clustering and the centroid diameters of clusters to detect massive network anomaly attacks. Syarif *et al.* [28] investigated the performances of five different clustering algorithms for anomaly detection problem, namely, k-means, improved k-means, k-mediods, expectation maximization (EM) and distance-based outlier detection algorithm. The experimental results show that the distance-based outlier detection algorithm outperform other clustering algorithms, and some researchers have obtained remarkable outcomes by using the clustering-based anomaly detection for network flows. Erman *et al.* [5] proposed a semi-supervised clustering method, which consists of a learner and a classifier, to classify network flows. Munz *et al.* [17] proposed flow anomaly detection approach based on K-means clustering algorithm. The training data used in this approach, which are unlabeled network flows, are separated into clusters of normal and malicious network flows, and the obtained cluster centroids can be used for detecting anomaly behaviors from on-line monitoring data. Ahmed *et al.* [1] used X-means clustering to detect collective anomaly flows. The X-means clustering is a variant of K-means algorithm, and provide an effective strategy to select the number of clusters k. Sheikhan *et al.* [23] proposed NIDS based on artificial neural network for detecting anomaly attacks of network flows. This system identifies malicious and benign flows using multi-layer perceptron neural classifier, and uses the gravitational search algorithm to optimize the interconnection weights of neural anomaly detector. Winter *et al.* [29] presented network intrusion detection approach to analyze anomaly flows, and used One-Class Support Vector Machines to identify malicious network flow. Therefore, the advantages of the anomaly detection approach based on clustering algorithm mainly include:

1) Generate anomaly detectors by self-learning approach;
2) Extract common features from the given dataset;

3) Detect unknown malicious attack behaviors from the changeable network environment.

Artificial immune network is one of important theories of artificial immune system inspired by vertebrate immune system, and holds some merits of artificial immune system, such as self-learning, self-adaption, self-organization and immune memory *etc.* [24]. According to immune network theory [10], the binding between idiotopes (molecular portions of an antibody) located on B cells and paratopes (other molecular portions of an antibody) located on B cells has a stimulation effect for B cells, and the interaction of B cells within a network will produce to a stable memory structure and account for the retainment of memory cells. For clustering algorithm inspired by immune network theory, the antibodies in immune network will be suppressed when similarity between antibodies is higher, conversely, they will be stimulated [4]. As a result, the expected network will be generated and its redundant antibodies will be eliminated. In recent years, artificial immune network has been employed by intrusion detection system to cluster anomaly malicious behaviors. Liu *et al.* [6] proposed an unsupervised anomaly detection algorithm based on artificial immune network, and the hierarchical agglomerative clustering is employed to help clustering analysis. Shi *et al.* [25] proposed an unsupervised UADINK approach based on K-means improved by immune network theory to detect anomaly behaviors of network flows. Lau *et al.* [12] proposed an unsupervised anomaly detection architecture which is capable of online adaptation inspired by immune network theory. Rassam *et al.* [21] investigated artificial immune network for clustering malicious attacks of intrusion detection system, and the rough set principle is employed to get the key element features of the given dataset so as to enhance detection rate of this system. These mentioned anomaly detection approaches show that artificial immune network can be used effectively for clustering network flows and refining detectors of anomaly detection system.

3 The Proposed ADAID

The proposed ADAID approach is an unsupervised anomaly detection strategy, and provides an automatic mechanisms to detect anomaly behaviors of network flows, therefore, it doesn't need the samples labeled by experts in order to cluster network flows. The framework of ADAID is shown in Figure.1. ADAID mainly includes four aspects:

1) Obtain network flows. They can be generated by replaying network packets of the given benchmark dataset or captured by real network world.

2) Select common features of network flows. We need select typical features of each network flow which can identify easily network behaviors in order to effectively distinguish malicious attack behaviors.

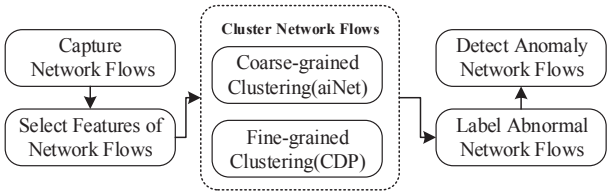


Figure 1: The framework of ADAID

3) Cluster network flows. It relates to two stages, namely the coarse-grained stage and the fine-grained stage. In the coarse-grained clustering stage, the aiNet model is introduced firstly for clustering samples from the given dataset [4]. The CDP algorithm [22],which is the fine-grained clustering, is used for clustering the output results of the coarse-grained clustering, and the aim that employ the CDP algorithm is to refine the cluster centroids from the previous stage and improve the attack detection accuracy of network flows.

4) Label abnormal network flows. After the final cluster centroids are obtained, each cluster centroid represents one of class network flows. Therefore, these cluster centroids need be labeled as abnormal/normal network flows so that ADAID can detect easily anomaly attacks of network flows. The relevant models and algorithms that compose ADAID are described as the following subsections, namely, artificial immune network (aiNet), clustering algorithm based on density peaks (CDP), clustering labeling algorithm (CLA) and flow anomaly detection algorithm (FAD).

3.1 The aiNet Model

The artificial immune network (aiNet) model is inspired by the clone selection principle and immune network theory of vertebrate immune system. The aiNet model [4] is firstly used for analyzing and filtering the crude dataset, and an internal image of all data samples in dataset, namely a refined relationship map, is constructed by immune evolution mechanisms, such as self-organizing, self-adaptive and self-learning *etc.* Therefore, the aiNet model is regarded as a coarse-grained method to refine some important features from complex information data. At present, the aiNet model has been introduced in pattern recognition, clustering data, and data compression *etc.*

The aiNet model is given in Figure 2. Its mainly aim is to search optimal memory antibodies of antigen ag_j by immune optimization strategies. This model may generate a memory antibody subset M_j in terms of the given antigen ag_j. After all antigens are travelled, the memory antibody set M will aggregate and storage the optimal antibodies. The antibody of M will be suppressed in each iterative operation of this model in order to avoid similar antibodies entering next generation. The memory antibody set

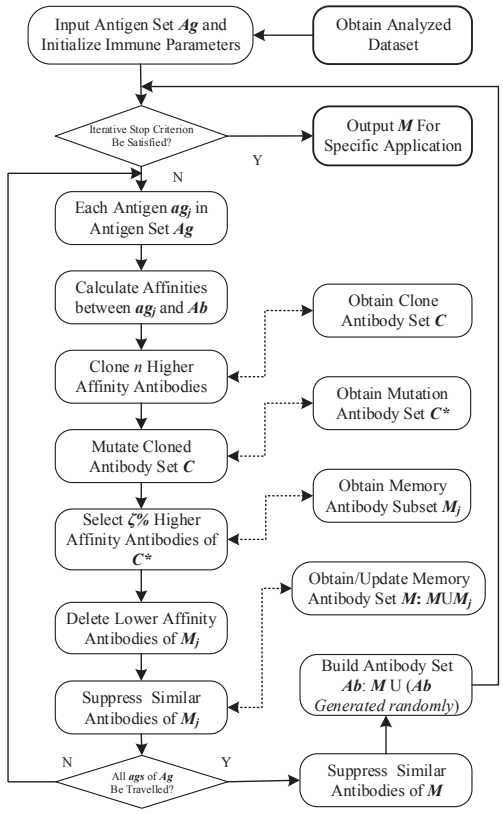


Figure 2: The flowchart of the aiNet model

M will be outputted as the final results or preprocessing data of the specific application system if the iterative stop criterion of this model is satisfied, for example, obtaining the cluster number/centroids of the relevant clustering algorithms. Therefore, the design of immune optimization strategies is a vital phase to improve the evolution learning capabilities of aiNet [4], such as clonal selection, immune mutation, and antibody suppression *etc.*

3.2 The CDP Algorithm

The clustering algorithm based on density peaks (CDP) [22] mainly includes three aspects:

1) Compute the local density ρ_i for each data point i of the given dataset, and the minimum distance δ_i between the data point i and any other data points with higher density.

2) Obtain cluster centroids by the drawn decision graph in terms of the local density and the minimum distance of each data of dataset, the cluster centroids possess both wider distance and higher density.

3) Assign each remaining data point of dataset to the same cluster centroid as its nearest neighbor of high density. The CDP algorithm can fast search and find density peaks by the specific functions which are used for calculating local density and distance of each data point of dataset.

For the CDP algorithm [22], Equations (1) and (2) are used for calculating ρ_i of each data point i, where d_c represents a cutoff distance, Equation (3) is used for calculating δ_i between each data point i and any other points with higher density, Equation (4) is used for discovering the power law distribution of all data points, and some data points that possess higher γ can be selected as cluster centroids.

$$\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c) \tag{1}$$

$$\chi = \begin{cases} 1, if(d_{ij} - d_c) < 0 \\ 0, \quad otherwise \end{cases} \tag{2}$$

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ij}) \tag{3}$$

$$\gamma_i = \rho_i \cdot \delta_i \tag{4}$$

According to the idea of ADAID, the CDP algorithm is viewed as a fine-grained clustering algorithm to classify effectively network flows, and the clustered data in CDP are the refined network flows that are learned by aiNet. The CDP algorithm is described by Algorithm 1.

Algorithm 1 The CDP Algorithm

1: **Input**: Memory antibody set M refined by aiNet
2: **Output**: Cluster number set T of M
3: **Start**
4: Calculate the distance d between each data point and any other data points in M, and find a cutoff distance d_c according to d of each data point in M
5: Calculate ρ_i,δ_i,γ_i by Equation (1), Equation (3) and Equation (4), respectively
6: Determine cluster centroids according to the power law distribution γ of all data points
7: Assign the rest of data points in M to the corresponding cluster centroid according to ρ_i, and finally obtain cluster number set T
8: **End**

3.3 The CLA Algorithm

The Cluster Labeling Algorithm (CLA) is used for labeling each cluster as normal/abnormal detector of network flows, and then these generated detectors are used for distinguishing malicious/benign network flows. In CLA, the labeled results for the corresponding clusters will influence anomaly detection performance of ADAID. The CLA algorithm is described by Algorithm 2.

3.4 The FAD Algorithm

The aim of the flow anomaly detection algorithm (FAD) is that provides an anomaly detection function for network

Algorithm 2 The CLA Algorithm

Input: Memory antibody set M, Cluster number set T, Training dataset Ag, Recognition threshold Rt
Output: Label set Nal of clusters
Start
Determine size of Nal, preprocess Ag
for each antigen of Ag **do**
  Calculate affinity of each antibody in M
  Find an antibody with maximum affinity, and accumulate the appeared times of this antibody
end for
for each different cluster number in T **do**
  Accumulate the matched times of different antibodies of M with antigens of Ag, and the cluster number of each antibody should keep same with T
  Calculate percent ratio Pr that each different cluster has recognized antigens of Ag
  if Pr is not less than Rt **then**
    Storage the number of this cluster and label this cluster as normal cluster in Nal
  else
    Storage the number of this cluster and label this cluster as abnormal cluster in Nal
  end if
end for
End

flows. Therefore, administrators can obtain network security situation by using FAD, and then some security strategies can be deployed timely. The FAD algorithm is described by Algorithm 3.

Algorithm 3 The FAD Algorithm

Input: Memory antibody set M, Cluster number set T, Label set Nal, Test dataset Tag
Output: Alarmed network flows which can match abnormal clusters of Nal
Start
Preprocess the test dataset Tag
for each antigen of Tag **do**
  Calculate affinities between each antibody in M and this antigen
  Choose an antibody with maximum affinity, and identify its cluster number in T
  if cluster number of this chosen antibody in T is equal to abnormal cluster in Nal **then**
    Alarm and Output this antigen, namely find an abnormal network flow
  end if
end for
End

4 Experimental Results

4.1 Dataset Description

To verify the effectiveness of the proposed ADAID, the ISCX 2012 IDS dataset [26] is adopted as benchmark dataset to detect malicious behaviors of network flows. This dataset includes seven days capturing data with overall 2,450,324 network flows, and is designed by the University of New Brunswick. In our evaluation experiments, the Tuesday's sub-dataset (23.4GB) of the ISCX 2012 IDS dataset is considered, and its brief statistics is listed by Table 1. Due to existing only a few malicious network flows from the 1^{st} flow to the $375{,}664^{th}$ flow in the Tuesday's sub-dataset, we select 196034 network flows from the $375{,}665^{th}$ flow to the last flow in this sub-dataset to demonstrate the effectiveness of the proposed ADAID. The trained/tested network flows consist of 158576 benign flows and 37458 malicious attack flows, and Table 2 shows the distribution of malicious attack flows of the selected network flows. The 10 percent flows of the selected network flows are viewed as training samples in order to generate detectors, and the rest network flows of that are viewed as test samples in order to verify the detection capability of ADAID.

4.2 Dataset Preprocessing

The preprocessing operation for data samples of the given dataset plays an important role in the machine learning fields, and it mainly relates to feature selection and dimension reduction. Considering the common features of network flows, we extracted 10 typical features of the ISCX 2012 IDS dataset listed by Table 3 to analyze malicious behaviors of network flows in terms of the empirical methods of the existed literatures [15, 23]. The aim of the preprocessing operation for network flows is that it may not only improve the anomaly detection precision but save the running costs both times and spaces in anomaly detection system.

As a key part of the preprocessing operation for the selected data sample, it's necessary that the key features of network flows are processed numerically. The minimum/maximum values of each selected feature is listed by Table 3. For the numeric range of these listed features, their default values are assigned according to the definitions and specifications of TCP/IP protocols. For instance, the fifth flag option of TCP header, SourceTCPFlags, is set to [0, 63]. For the rest features listed by Table 3, their maximum values aren't be limited, but they should be greater than the real values of any selected network flows. Take the third feature as an example, it is set to [0, 40,000] because the largest value of the transferred destination packets in any flows is not greater than 38,685.

Table 1: Tuesday's network flow statistics in the ISCX 2012 IDS dataset

Feature	Value	Feature	Value
Flows	571,698	Destination Bytes	22,842,855,364
Attack Flows	37,460	Source Bytes	1,905,193,956
Normal Flows	534,238	Destination Packets	21,746,115
ICMP Flows	6,073	Source Packets	13,254,945
TCP Flows	441,563	Destination IPs	26,780
UDP Flows	124,023	Source IPs	2,196

Table 2: Distribution of malicious network flows in the selected network flows

Network Flows	Attacks of	Network Flows	Attacks of
19,603 (10%)	79	117,620 (60%)	7,054
39,207 (20%)	82	137,224 (70%)	18,511
58,810 (30%)	83	156,827 (80%)	29,363
78,414 (40%)	84	176,431 (90%)	37,421
98,017 (50%)	85	196,034 (100%)	37,458

4.3 Evaluation Matrices

Anomaly detection is viewed as one kind of two-class problems. Network flow behaviors can be classified as benign behaviors or malicious behaviors by using anomaly detection algorithms. In this paper, we introduce three metrics to evaluate the performance of ADAID [25]:

1) Accuracy Rate (AR) that indicates the clustered correctly portion for all test samples of network flows, and its formal definition is shown in Equation (5);

2) Detection Rate (DR) that indicates the malicious attack flows which may be recognized correctly from test samples, and its formal definition is shown in Equation (6);

3) False Alarm Rate (FAR) that indicates the real benign flows which have been recognized as malicious attack flows from test samples, and its formal definition is shown in Equation (7).

In Equations (5) ,(6) and (7), TP(True Positive) indicates the cumulative number for the malicious attack flows which are labeled as real attack flows in test samples, FP(False Positive) indicates the cumulative number for the malicious attack flows which are labeled as benign flows in test samples, TN(True Negative) indicates the cumulative number for the benign flows which are labeled as normal network flows in test samples, and FN(False Negative) indicates the cumulative number for the benign flows which are labeled as malicious attack flows in test samples. To avoid bias, the final results of these evaluation metrics are given by the average results of Nr (=10) independent trials.

$$AR = \frac{TP+TN}{TP+FP+TN+FN} \quad (5)$$

$$DR = \frac{TP}{TP+FP} \quad (6)$$

$$FAR = \frac{FP}{TN+FP} \quad (7)$$

4.4 Parameter Settings

4.4.1 Evolution Parameters of aiNet

To demonstrate the effectiveness of ADAID, three clustering algorithms, namely aiNet model, aiNet based hierarchical clustering (aiNet_HC), and the proposed clustering algorithm combining aiNet with CDP (aiNet_DP), use same evolution parameter values listed by Table 4.

4.4.2 Parameter Settings of CDP

The cutoff distance dc and the cluster number nc are two key parameters of CDP, and can improve the clustering precision of network flows. The parameter dc represents a border region of each cluster. For the cluster centroid of each cluster, if the distance between this cluster centroid and one of data/vector points of the clustered dataset is not greater than dc, this data/vector point will be assigned to this cluster. Therefore, dc is an important parameter to discriminate correctly different clusters. Known from Reference [22], supposing nd represents the number of data/vector points of the clustered dataset, $n = \lceil (0.5 * (nd-1) * nd) \rceil$ represents the total number of points by calculating distance between any two different data/vector points of the clustered dataset, and the value of dc can be chosen any one point around the former 1-2% of the total number of points after these points are sorted in ascending order. The larger dc is, the lesser the number of clusters are; conversely, the smaller dc is, the more the number of clusters are. The dc in this paper is obtained from one point around 1.5% of the total number of points in the clustered dataset.

To obtained reasonable nc of dataset, we firstly need calculate $ri = pi * di$ in Equation (4) after choosing a suitable dc, and ri is used for exhibiting a power law distribution of all data points, and then all elements in r are re-sorted in descend order, where pi denotes the local

Table 3: Flow feature description for the ISCX 2012 IDS dataset

Feature name	Description	Minimum Value	Maximum Value
TotalDestinationBytes	Transferred destination octets	0	60,000,000
TotalSourceBytes	Transferred source octets	0	2,000,000
TotalDestinationPackets	Transferred destination packets	0	40,000
TotalSourcePackets	Transferred source packets	0	20,000
DestinationTCPFlags	Destination TCP flags	0	63
SourceTCPFlags	Source TCP flags	0	63
DestinationPort	Destination port number	0	65,535
SourcePort	Source port number	0	65,535
ProtocolName	IP protocol number	0	255
Duration	Duration of flow (in seconds)	0	864,000

Table 4: Evolution parameters of the aiNet model

Parameter	Value	Parameter	Value
Number of Runs Nr	10	Re-selection Rate Rr	0.2
Number of Generations Ng	10	Hypermutation Rate Hr	4
Population Size Ps	10	Natural Death Threshold Nt	1
Taken Best-matching Cells Tbc	4	Suppression Threshold St	0.1

density of each data point i, and di denotes its distance from points with higher density. The i-th data point with corresponding to ri has more chance as a cluster centroid if ri is more bigger [22]. The nc will be set to 35% of the total number of r in this paper, and the total number of r depend on the output results of the coarse-grained clustering stage.

4.4.3 Parameter Settings of CLA

The recognition threshold Rt is an important parameter of CLA, and it is used for labeling normal/abnormal clusters. A reasonable selected Rt can increase the DRs and decrease the $FARs$ of anomaly detection system. There are two strategies to obtain the reasonable value of Rt. The first strategy is that the ratio, which is 10 percent of all samples of training dataset, may be considered as the value of Rt. The second strategy is that the ratio between the existing real attacks and the total amount samples in training dataset also may be considered as the value of Rt. Known from Table 2, there are 79 real attack flows in all 19603 network flows of training dataset, so the highest attack ratio in training dataset is about 0.004. According to the first strategy, one kind of network flows is regarded as normal if its amount of network flows isn't less than 10 percent of all samples in training dataset, namely Rt=0.1. Therefore, the value of Rt may be defined from 0.0040 to 0.1 according to the above-mentioned two strategies, but the reasonable value of Rt should close to 0.0040 in order to detect effectively anomaly network flows. The experimental results, which are AR, DR, and FAR, of the proposed ADAID are shown by Figure 3. Known from Figure 3, AR, DR and FAR of ADAID have got different results according to the change of Rt that ranges from 0.0040 to 0.0051. Rt in ADAID is set to 0.0046 in this paper, and the corresponding AR, DR, and FAR are 85.93%, 100% and 14.64%, respectively.

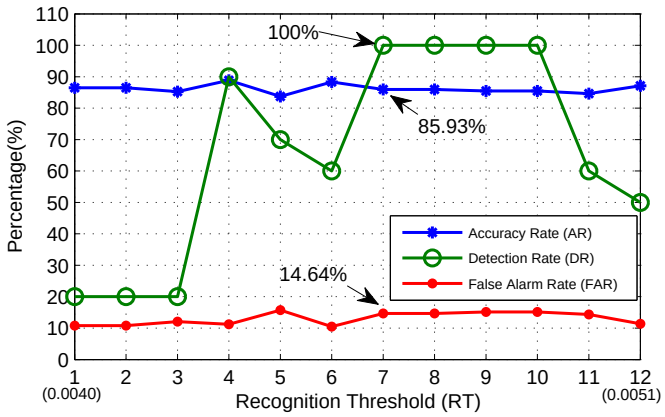


Figure 3: Performance comparison of ADAID with different Rt

4.4.4 Performance Evaluation of ADAID

Known from the proposed ADAID, the clustering algorithm is reviewed as a vital part of anomaly detection strategy. In this paper, we discuss the performances of three different clustering algorithms, which are aiNet, aiNet_HC and the proposed aiNet_DP, to detect anomaly behaviors of network flows. After running clustering operation for network flows, CLA and FAD are used for rec-

ognizing malicious clusters of network flows and detecting anomaly behaviors of network flows, respectively. Table 5 shows the experimental results of three different anomaly detection approaches.

Known from Table 5, compared with the aiNet based anomaly detection approach, the accuracy rates (ARs) of the aiNet_DP based anomaly detection approach in training stage and test stage are reach to 85.93% and 85.78%, respectively. And the corresponding false alarm rates ($FARs$) are only 14.64% and 15.28%, respectively. Therefore, the aiNet_DP based anomaly detection approach possesses higher ARs and lower $FARs$ than the aiNet based anomaly detection approach. Although the aiNet_HC based anomaly detection approach possesses higher ARs and lower $FARs$ than ADAID, its detection rates (DRs) in training stage and test stage are only reach to 70% and 70.75%, respectively. Obviously, the DRs of ADAID are about 30% higher than the aiNet_HC based anomaly detection approach. The deviation of the ARs between ADAID and the aiNet_HC based anomaly detection approach in training stage and test stage do not exceed 5%, and meanwhile the deviation of $FARs$ of them do not exceed 6%.

The aiNet based unsupervised clustering is regarded as an effective strategy for detecting network anomaly behaviors in anomaly detection system. The experimental results show that the aiNet based anomaly detection approach has more improvement space to enhance its ARs and reduce its $FARs$. Therefore, the improved clustering algorithm combining aiNet with other clustering algorithm is considered as more radical method to improve the effectiveness of clustering algorithm, such as aiNet_HC and aiNet_DP listed by Table 5. Compared with the aiNet based anomaly detection approach, the DRs of the aiNet_HC based anomaly detection approach decline even if its ARs and $FARs$ are improved. However, compared with two anomaly detection approaches which are respectively based on aiNet and aiNet_HC, the proposed ADAID combining aiNet with density peaks is more ideal approach for detecting anomaly behaviors of network flows because it possesses precise DRs, higher ARs and reasonable $FARs$.

5 Conclusions

An anomaly detection approach for network flow using artificial immune network and density peak (ADAID) in this paper is proposed to detect malicious attack behaviors and benign activities of network flows. In ADAID, its clustering algorithm consists of aiNet and CDP, where aiNet and CDP are viewed as coarse-grained clustering and fine-grained clustering, respectively. The aim of this clustering algorithm is to cluster similar values of common features from massive network flows and finish the classification of network flows. The anomaly detection of ADAID comprises of CLA and FAD, where CLA is to label clusters as abnormal or normal by learning network flows of training dataset, and the identified clusters are viewed as detectors; FAD can be used for detecting malicious attack behaviors from network flows of test dataset.

To demonstrate the effectiveness of ADAID, we firstly introduce three different clustering algorithms, namely, aiNet, aiNet_HC and the proposed aiNet_DP, to classify network flows of training dataset, respectively. The output clusters generated by three clustering algorithms all are labeled by CLA. And then the labeled clusters use FAD to detect network flows of test dataset. To improve the performance of ADAID, we analyzed the parameters of CDP, namely cutoff distance dc and cluster number nc, to obtain more precise clusters of network flows, and meanwhile we discussed the recognition threshold Rt of CLA to distinguish reasonably malicious flows and benign flows. In our experiments, the ISCX 2012 IDS dataset is adopted to evaluate ADAID. To avoid bias, the final experimental results are given by the average experimental results of Nr independent trials, and show that ADAID is a radical anomaly detection approach for network flows.

We will further improve ADAID in our future works that relates to unsupervised clustering, automatic detection, running costs *etc.* We will try to adopt more efficient immune optimizing strategies and parallel computing approaches to improve ADAID for detecting anomalies of network flows.

Acknowledgments

This work was funded by China Scholarship Council, Australian Research Council Discovery Project DP150104871, the China Postdoctoral Science Foundation under Grant No.2014M562102, Hunan Provincial Natural Science Foundation of China under Grant No.2015JJ2112, the Scientific Research Fund of Hunan Provincial Education Department of China under Grant No.18A449. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. Ahmed and A. N. Mahmood, "Network traffic analysis based on collective anomaly detection," in *The 9th IEEE Conference on Industrial Electronics and Applications*, vol. 2014, pp. 1141–1146, 2014.

[2] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[4] L. N. de Castro and F. J. Von Zuben, "aiNet: An artificial immune network for data analysis," *Data Mining: A Heuristic Approach*, vol. 2001, no. 1, pp. 231–259, 2001.

Table 5: Accuracy comparison for clustering algorithm based anomaly detection

Anomaly Detection	Training phase			Test phase		
	AR(%)	DR(%)	FAR(%)	AR(%)	DR(%)	FAR(%)
aiNet Based	76.43	100	24.53	76.39	100	23.71
aiNet_HC Based	90.47	70	8.70	89.37	70.75	10.55
ADAID	85.93	100	14.64	84.78	100	15.28

[5] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Offline/realtime traffic classification using semi-supervised learning," *Performance Evaluation*, vol. 64, no. 9-12, pp. 1194–1213, 2007.

[6] L. Fang and L. Le-Ping, "Unsupervised anomaly detection based on an evolutionary artificial immune network," in *Workshops on Applications of Evolutionary Computation*, pp. 166–174, 2005.

[7] Y. Hamid, V. R. Balasaraswathi, L. Journaux, and M. Sugumaran, "Benchmark datasets for network intrusion detection: A review," *International Journal of Network Security*, vol. 20, no. 4, pp. 645–654, 2018.

[8] Y. He, "Identification and processing of network abnormal events based on network intrusion detection algorithm," *International Journal of Network Security*, vol. 21, no. 1, pp. 153–159, 2019.

[9] M. S. Hwang, S. K. Chong, and H. H. Ou, "On the security of an enhanced umts authentication and key agreement protocol," *European Transactions on Telecommunications*, vol. 22, no. 3, pp. 99–112, 2011.

[10] N. K. Jerne, "Towards a network theory of the immune system," in *Annales d'immunologie*, vol. 125, pp. 373–389, 1974.

[11] D. Kwon, H. Kim, J. Kim, C. S. Sang, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, no. 5, pp. 1–13, 2017.

[12] H. Lau, J. Timmis, and I. Bate, "Anomaly detection inspired by immune network theory: A proposal," in *IEEE Congress on Evolutionary Computation*, pp. 3045–3051, 2009.

[13] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," *Proceedings of the Twenty-eighth Australasian conference on Computer Science*, vol. 8, pp. 333–342, 2005.

[14] B. Li, J. Springer, G. Bebis, and M. H. Gunes, "A survey of network flow applications," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 567–581, 2013.

[15] W. Li, M. Canini, A. W. Moore, and R. Bolla, "Efficient application identification and the temporal and spatial stability of classification schema," *Computer Networks*, vol. 53, no. 6, pp. 790–809, 2009.

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

[17] G. Munz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," *GI/ITG Workshop MMBnet*, 2007. (`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.323.6870&rep=rep1&type=pdf`)

[18] A. W. Moore, Z. Denis, and M. L. Crogan, "Discriminators for use in flow-based classification," *Queen Mary and Westfield College, Department of Computer Science*, 2005. (`https://www.cl.cam.ac.uk/~awm22/publications/RR-05-13.pdf`)

[19] S. Petrovic, G. Alvarez, A. Orfila, and J. Carbo, "Labelling clusters in an intrusion detection system using a combination of clustering evaluation techniques," *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, vol. 6, pp. 129b–129b, 2006.

[20] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in *Proceedings of ACM CSS Workshop on Data Mining Applied to Security*, 2001. (`http://citeseerx.ist.psu.edu/viewdoc/citations?doi=10.1.1.126.2131`)

[21] M. A. Rassam and M. A. Maarof, "Artificial immune network clustering approach for anomaly intrusion detection," *Journal of Advances in Information Technology*, vol. 3, no. 3, pp. 147–154, 2012.

[22] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[23] M. Sheikhan and Z. Jadidi, "Flow-based anomaly detection in high-speed links using modified gsa-optimized neural network," *Neural Computing and Applications*, vol. 24, no. 3-4, pp. 599–611, 2014.

[24] Y. Shi, R. Li, X. Peng, and G. Yue, "Network security situation prediction approach based on clonal selection and scgm(1,1)c model," *Journal of Internet Technology*, vol. 17, no. 3, pp. 421–429, 2016.

[25] Y. Shi, X. Peng, R. Li, and Y. Zhang, "Unsupervised anomaly detection for network flow using immune network based k-means clustering," in *International Conference of Pioneering Computer Scientists, Engineers and Educators*, pp. 386–399, 2017.

[26] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, 2012.

[27] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[28] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in *International Conference on Networked Digital Technologies*, pp. 135–145, 2012.

[29] P. Winter, E. Hermann, and M. Zeilinger, "Inductive intrusion detection in flow-based network data using one-class support vector machines," *IEEE 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS'11)*, vol. 2011, pp. 1–5, 2011.

[30] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[31] J. Zhang, "Application of artificial intelligence technology in computer network security," *International Journal of Network Security*, vol. 20, no. 6, pp. 1016–1021, 2018.

Biography

Yuanquan Shi received the B. S. degree from Hunan Normal University, Changsha, China, in 2000, the M.E. degree from National University of Defense Technology, Changsha, China, in 2005, the Ph.D. Degree from Sichuan University, Chengdu, China, in 2011, all in computer science. Currently, he is Professor in the School of Computer Science and Engineering, Huaihua University, China, and also a visiting scholar in the School of Computer Sciences at the University of Adelaide, Australia. His research interests include network security, intelligent computing, time series prediction, and parallel computing.

Hong Shen is Professor (Chair) of Computer Science in University of Adelaide, Australia. He received the B.Eng. degree from Beijing University of Science and Technology, M.Eng. degree from University of Science and Technology of China, Ph.Lic. and Ph.D. degrees from Abo Akademi University, Finland, all in Computer Science. He was Professor and Chair of the Computer Networks Laboratory in Japan Advanced Institute of Science and Technology (JAIST) during 2001∼2006, and Professor (Chair) of Compute Science at Griffith University, Australia, where he taught 9 years since 1992. With main research interests in parallel and distributed computing, algorithms, data mining, privacy preserving computing, network security, high performance networks and multimedia systems, he has published more than 300 papers including over 100 papers in international journals such as a variety of IEEE and ACM transactions. Prof. Shen received many honours/awards including China National Endowed Expert of Thousand Talents, Chinese Academy of Sciences Hundred Talents, National Education Commission Science and Technology Progress Award, and Chinese Academy of Sciences Natural Sciences Award. He served on the editorial board of numerous journals and chaired several conferences.

Adaptive Access Control Model of Vehicular Network Big Data Based on XACML and Security Risk

Peng-Shou Xie[1,2], Hong-Jin Fan[1], Tao Feng[1], Yan Yan[1], Guo-qiang Ma[1], and Xue-Ming Han[1]

(Corresponding author: Hong-jin Fan)

School of Computer and Communications, Lanzhou University of Technology[1]

Research Center of Engineering and Technology for Manufacturing Informatization of Gansu Province[2]

No. 287, Lan-gong-ping Road, Lanzhou,Gansu 730050, China

(Email: fan_hjin@163.com)

(Received Oct. 17, 2018; Revised and Accepted Mar. 23, 2019; First Online June 22, 2019)

Abstract

With the constant expansion of vehiclescale and the continuous development of Internet of Vehicles, the network environment of the data resource of Internet of Vehicles is becoming more and more complex. Traditional access control models have been difficult to meet the requirements of various access control conditions and dynamic adaptive adjustment of access control strategy. Aimed at the problem of adaptive access control model of vehicular network big data environment, XACML powerful ability of expressing access strategy is used in the paper, and we conduct the risk quantification based on 10 counts of risk factors, risk threshold and risk quota mechanism are also used for risk management. Experimental verification indicated that the risk adaptive access control model is effective, the research results will have great significance for promoting the application research of Internet of Vehicles and its safety technology and improving people's quality of life.

Keywords: Access Control; Big Data Security; Internet of Vehicles; Risk Adaptive; XACML

1 Introduction

The Internet of Vehicles is a large interactive network that contains information about vehicular location, speed, route, etc. Based on mobile communication and the information science technology, the Internet of Vehicles uses wireless communication technology, automotive sensors technology, global-positioning technology and automobile data recorder technology to complete the data collection of vehicular information and the surrounding environment, data transmission and processing, etc, in order to achieve effective intelligent monitoring, planning and management of vehicles, people, roads and locations [6]. It can be seen from the generation of big data in the Internet of vehicles,vehicular network big data has the characteristics of 4V, that is, Volume, high Velocity, Variety and high Value.In addition, it also has the following characteristics: Spatial and temporal scales span , large dynamic variability, high randomness, locality and finite life cycle. These characteristics of big data in the Internet of vehicles require us to provide more convenient services, such as data sharing and efficient computing to improve the processing efficiency of access control. In addition, when users enjoy the service, if do not provide reliable protection to these data with a large number of ownership characteristics, it will bring huge losses.

Access control technology according to the pre-defined access control policy ensures that resources can only be operated legally by legitimate visitors thus preventing unauthorized access to information. With the emergence of new computing environments such as cloud computing, Internet of things, some characteristics of those have brought great challenges to the application of access control technology, which makes the traditional access control model for closed environments such as Discretionary Access Control (DAC), Mandatory Access Control (MAC) and role-based Access Control (RBAC) difficult to apply directly to the new computing environment [10]. Subsequently, many related research work began to emerge, most of which focused on how to extend the access control of traditional models and how to introduce risks in the extended model. Hui Zhen *et al.* [9] proposed a risk-based access control model for medical big data, which can adaptively adjust doctors' access ability and protect patients' privacy. Chen Aiguo *et al.* [4] proposed a dynamic risk-based access control model, which emphasizes the risk measurement as an auxiliary decision indicator. The model uses the sliding window calculation method based on data stream, and the comprehensive final decision is affected by the policy, risk measurement and dynamic threshold. Xu Jing *et al.* [16] proposed a dynamic

access control model, which introduced both the times of threat behavior and risk threshold into the trust model, the dynamic authorization was achieved by mapping trust level and permission. Almehmadi Abdulaziz *et al.* [1] proposed intent-based access control model, which uses the intent and intent motivation level to compute the access risk and greatly reduce the damages caused by internal threats. Chattopadhyay, Arup Kumar *et al.* [5] proposed a scheme uses simple Boolean based encryption and decryption of the data files which is low in computational cost, it reduced the risk of highly sensitive data from internal or external attacks.

Amghar,Sara *et al.* [14] proposed a new hybrid model, which uses KP-ABE and authentication system scheme to enhance the security and privacy of shared big data in the cloud.This model realized flexible and fine-grained access control for storing big data. Kibiwott, Kittur Philemon *et al.* [11] proposed a Cloudlet-Based eHealth Big Data System with Outsourced Decryption. It overcomes so many problems, such as confidentiality of data outsourced to the cloud, integrity of stored data, wide area network latency delays, and the resource constraints of the mobile devices. Lee, Ki Young *et al.* [12] proposed spatio-temporal XACML which could accept not only geospatial information but also temporal information and it compensated for the lack of Geo-XACML. Arunkumar [3] demonstrated the ability of the current OASIS standard to control access to XACML policies,described some confusing methods, and made specific suggestions on which elements should be involved in the process of access control. By combining XACML framework with the attribute based on encryption mechanism, Yang Yafeng [17] designed and realized a kind of attribute-based security enhanced cloud storage access control system applicable to cloud storage environment. Hou Shuchen [8] proposed a security access model for strengthening web services-based business system based on XACML system. Some progress has been made in the solution to security risk access control, but there is still a problem that is insufficient adaptive adjustment capability. Importantly, there are relatively few researches on risk adaptive access control methods specific to the vehicular network big data environment.

Considering the shortcomings of the above researches, we propose a security risk adaptive access control model. By the model, data security can be better protected. Based on the full use of XACML's powerful access policy expression capabilities, the introduction of quantitative risk control functions extends the XACML architecture, enabling dynamic adjustment of access policies based on visitor access, greatly improving vehicle network access control flexibility and applicability in complex network environments. The rest of this article is organized as follows. Section 2 introduces the XACML extension framework and the basic structure of the policy set. Section 3 describes the quantification process for the big data security risk of the Internet of Vehicles. Section 4 describes the decision and execution process of the strategy. The effectiveness of the model was tested by simulation experiments in Section 5. Section 6 concludes the solutions.

2 XACML Extended Framework and Policy Set Infrastructure

2.1 XACML Extended Framework

Figure 1 shows the XACML extended framework: the left side represents the XACML module, and the right side is the newly added module.

The functions of each module are as follows:

- Policy Administration Point (PAP): Create and maintain policies,policy sets and use files for storage.
- Policy Decision Point (PDP): Determine whether access requests are allowed, evaluate available policies, and provide authorization decisions.
- Policy Enforcement Point (PEP): Receive and send messages, interact with external applications according to results and obligations.
- Policy Information Point (PIP): Provides attributes information about subject, resources and environment.
- Context Handler (CH): Convert the access request to the XACML format and send it to the PDP.
- Subject (S): A visitor that performs an action on a resource.
- Resource (R): The data, services, and system components that the system provides to the visitor.
- Environment (E): A set of attributes that are related to authorization decisions and that are not related to specific property, resources or actions.
- Risk Engine (RE): PDP is invoked to handle risk-based access control. It mainly analyzes or solves these resource related risk policies; RE gets the attributes and request information from PDP, and these parameters would be substituted into a specific algorithm to calculate the risk value about the whole access request.
- Risk Quantification Function (RQF): Execute the risk measurement, they play a role inside the risk engine and make the use of risk policies more convenient.
- Risk Policies (RP): Define how each risk-based access control policy evaluates each resource. Using XACML's strong access strategy expression capabilities, we can does not change the original policy structure, just by setting the parameters of the rules in the strategy, the strategy for authorizing by risk value can be implemented. And risk management department can adjust risk strategy as needed. It can give

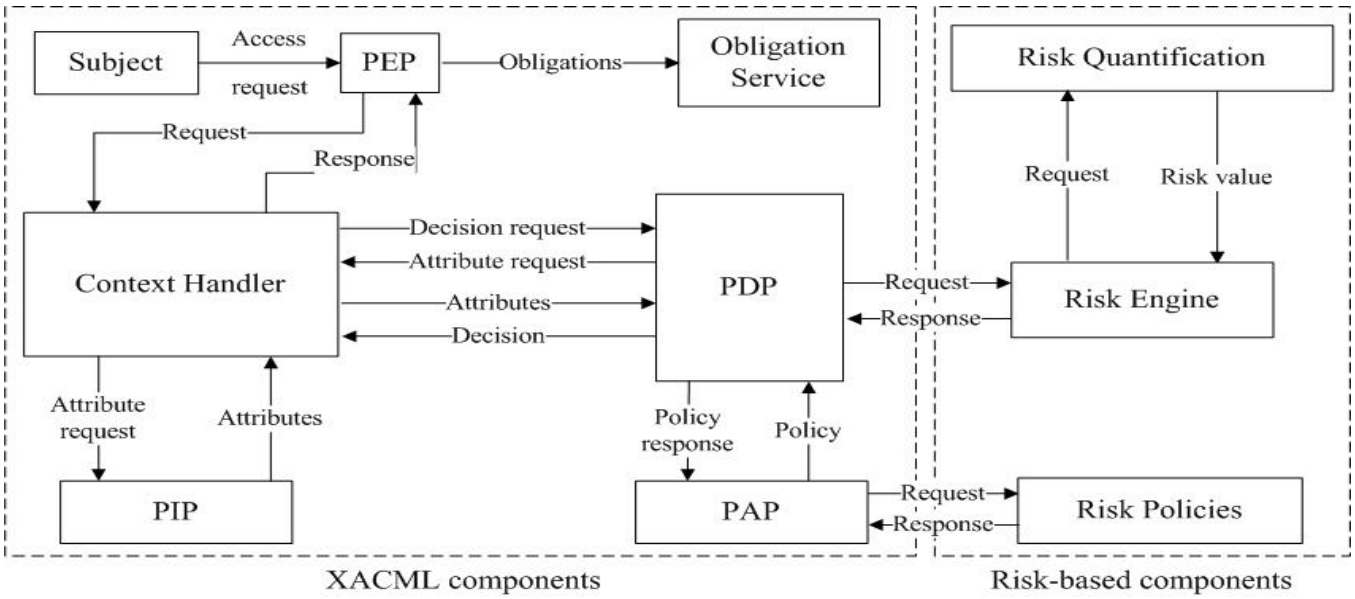


Figure 1: XACML extension framework

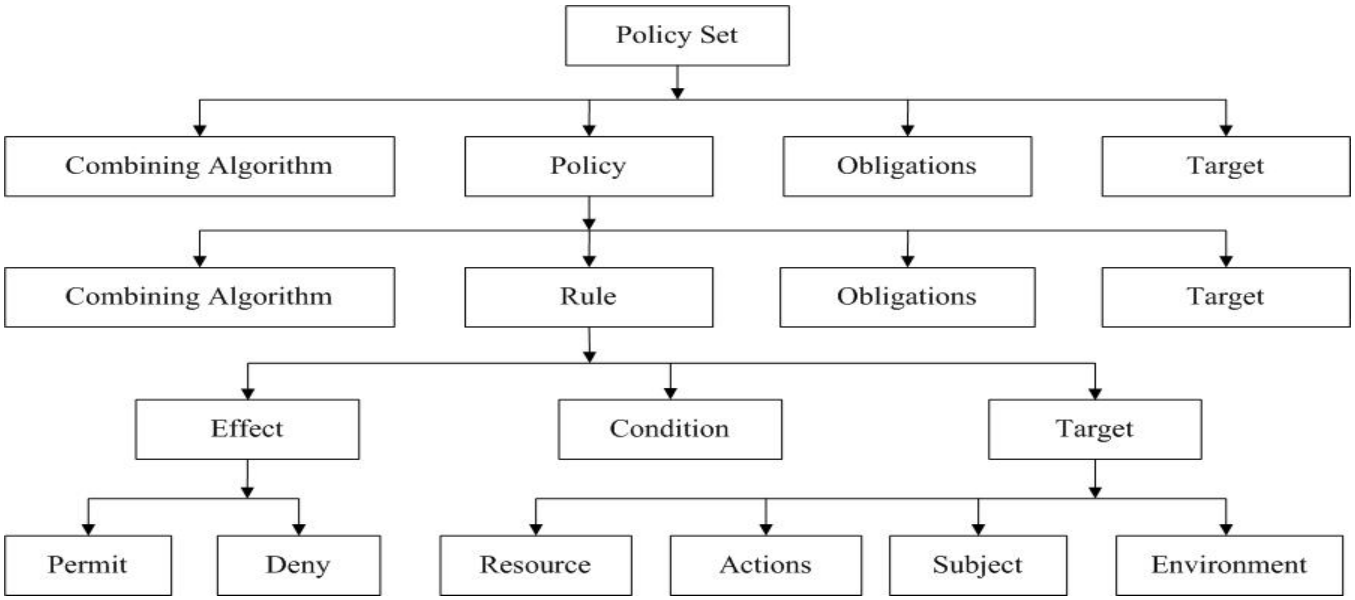


Figure 2: XACML basic structure of strategy set

Table 1: Merge algorithm

Algorithm	**Description**
Reject priority algorithm	If any assessment returns a refusal, the result must be a refusal, even if other assessments have returned permission.
Apply the algorithm first	Rules are evaluated in the order in which they are listed.
Unique application algorithm	For all policies in the policy set, if there is no applicable policy, the result is not applicable.If multiple strategies are applied, the result is indeterminate. If only one policy is applied, the result is the result of evaluating the policy.
License priority algorithm	If any assessment returns a license, the result must be a license, even if other reviews have returned a denial.

the judgment result by comprehensively judging the risk value of the access request and the allowable risk value defined by the policy.

The process framework of XACML after adding the risk point: When the subject issues an accessing request, authorization request will be send to PEP. Then PEP sends it to CH which will standardize the description of the attributes of visitors. In the meanwhile, create an XACML request and send it to PDP to decide whether the accessing authorization can be allowable. The practical policies are stored in PAP. PDP does not use all the policies in each accessing request. Instead, it searches applicative policies to evaluate the request and return the authorization decision back. For risk decisions, PDP will examine whether the resources used this kind of assessment method or not based on the instructions of relevant risk policies. If such a strategy does not exist, the result will be the traditional accessing decisions, conversely, PDP will send the request to RE to check the basic risk policies in the first place. If the basic policy is permitted, RE will Quantitative risk, and the result of risk measurement will be aggregated into a single value before return to PDP. PDP will decide whether to allow requests and send those to PEP and finally fulfil corresponding obligations, which are based on XACML policy and risk policy and merge algorithm.

2.2 The Basic Structure of Strategy Set

As shown in Figure 2, rule is the smallest unit of evaluating access requests which are consisted of three parts: Ondition, target and effectiveness. Logical judgement of accessing request is realized by conditional implementation. The decision result of rule determination is obtained by matching the subject, resource, action, environment in the target and the corresponding attributes in the accessing request. The upper layer of the rule is strategy, which is composed of target, merge algorithm, responsibility set and rule set. Responsibility indicates the tasks to be completed at the stage of strategy implementation. Strategy set is the most top-level structure of policy. XACML implements hierarchical policy management mechanism by this kind of nested structure. The merge algorithm is used to define the merging logic of results decided by multiple rules. According to the combination logic, the results of all rules are merged to get the final decision.

2.3 Merge Algorithm

A policy set may contain multiple policies and multiple rules. Different decision rules may result in conflicts. In order to obtain a unified decision result, a suitable merge algorithm is needed to resolve the conflict. The standard merge logic used by XACML has Deny-overrides, First-applicable, Only-one-applicable, Permit-overrides. Specific description as shown in Table 1.

3 Quantification of Vehicular Network Big Data Security Risks

Quantifying risk is to estimate each visit behavior and classify it into risk levels. Each risk level represents an access decision and action behavior. The top risk level and an access decision reject the contact, meaning the risk is high. Called that the boundary is a hard boundary; The lowest risk level contact with an access decision "allow" that means the risk is low, called that the boundary is a soft boundary; Between the hard and soft boundaries, and also an access decision associate with multiple operational actions. Traditional risk access control, which is static access control, just is allow or deny. But the risk adaptive access control described in this paper is a dynamic, multi-decision access control, that is allow-deny [13].

For risk points, the stage of risk assessment needs to input some factors (denotes subject, denotes object, denotes action, denotes context) to determine whether the accessing request was granted or rejected. This output function is based on a risk threshold and the mechanism of risk quotas to be managed. Specifically, our model determines the risk associated with access requests (visitor trust level and requested object security level and so on.) and then judging such requests according to the risk threshold of situational conditions. And if the quantified risk is below the risk threshold, the access request will be allowed, otherwise it will be denied. Another parallel condition of can Access is that the risk of quantification is less than risk threshold [2]. As shown in Formula (1).

$$canAccess(s,o,a,c) = \begin{cases} 1 \text{ if } risk < riskThreshold \\ \qquad \text{and } riskQuotas > 0 \\ 0 \text{ otherwise} \end{cases} \tag{1}$$

Risk value(s, o, a, c)denotes that the risk comes from when subjects perform operations on objects according to context. Result 1 indicates the right to be granted, while the result of 0 is denied.

3.1 Risk Quantification Base On C, I, A

Table 2 shows the influences on data from different types of accessing behavior, this model means that accessing behaviors do the risk quantification with Confidentiality(C), Integrity (I) and Availability (A). When behaviors include risk attributes, it is designated as 1 otherwise as 0.

CiaRisk can be calculated by Formula (2) and Formula (3):

$$ciaRisk = C * P_b + I * P_b + A * P_b \tag{2}$$

$$P_b = \frac{N_b}{N_{all}}. \tag{3}$$

Among them, P_b is the probability of occurrence of behavior, N_b is the number of the behavior occurs, N_{all} is the total number of occurrences of all behaviors, and the

Table 2: Risk value from Santos *et al.* [7]

Behavior	Data attribute	C	I	A
Create	sensitive/insensitive	0	1	1
View	sensitive	1	0	0
View	insensitive	0	0	1
Modify	sensitive/insensitive	0	1	1
Delete	sensitive/insensitive	0	1	1

probability P_b of each behavior can be calculated by using the statistical history of Formula (3). If the probability of the visitor modifying the data is 0.6, then ciaRisk = (0 * 0.6) + (1 * 0.6) + (1 * 0.6) = 1.2.

3.2 Risk Quantification Base on 6 Risk Factors of Internet of Vehicles

According to the results of researchers such as Santos *et al.*, Table 3 presents 6 risk factors under the Internet of Vehicles environment which are the index of risk quantification evaluation. The first group (Charact.of Visitor) shows the relevant resource information of visitors. The second group (Characteristics of Information and Requirements) shows the relevant risk of resource itself. It enumerates 2 groups total 6 risk factors and their weights. The total weight of each group is 0.5(1/2), the weight of each factor in each group is 0.5 / n. N is the number of factors in this group.

ContexRisk can be calculated by Formula (4):

$$contextRisk = \sum_{n=1}^{6} f_n * r_n \tag{4}$$

$$Risk_{Role} = \begin{cases} 1 & \text{R} \in \text{SuperAdmin} \\ 5 & \text{R} \in \text{Admin} \\ 10 & \text{R} \in \text{User} \\ 15 & \text{R} \in \text{Otherwise} \end{cases} \tag{5}$$

Among them, f_n is the weight of the risk factor, r_n is the risk value of the risk factor. The risk value of each risk factor is defined in advance, as Formula (5) defines the risk value for the role factor.

3.3 Risk Quantification Based on C, I, A, H and 6 Risk Factors

Ten risk factors are used for risk quantification in the paper, including 6 contextual factors, C,I,A risk factors and H historical records,that is:

- Safety features of behavior: security impact of confidentiality, integrity and availability behavior on resources.
- Contextual factor: visitor features, information features.
- History record: historical risk associated with the visitor.

The ultimate risk is:

$$aggregatedRisk = w_1 * ciaRisk + w_2 * contextRisk + w_3 * hisRisk \tag{6}$$

The H is the past risk value(hisRisk), which can be obtained by reading the past risk value from the database. If the visitor is first visit, the visitor's hisRisk is 0, w_1, w_2, w_3 are the weights of each metric category.

3.4 Vehicular Network Big Data Risk Threshold and Risk Quota Mechanism

The risk quota indicates how much the system is tolerant of the risk posed by each visitor. For access control, the system periodically assigns each visitor a certain number of risk quotas. Each visitor's visit behavior poses a certain risk and consumes the same amount of risk quota. If the visitor's risk quota is greater than zero, they can continue to access; Otherwise their access request will be denied until a new risk quota is obtained. The allocation of quotas is regular. The risk quota allocated each time should satisfy the normal visitors and will not be exhausted before the next allocation, that is, the request of normal visitors can be successfully passed.

For the formal description, the following symbols will be used.

V: A collection of visitors;

D: A collection of access data;

R: A collection of access records;

T: A collection of the same type of data.

This model periodically analyzes data visitor access records and calculates risk values. In the analysis of the history of the data visitor V_i, the same data access section visited by each visitor is integrated and recorded as D(V_i, D_j), where D_j is the data access section of the visitor, and $D_j \epsilon$ D . The label of one of the types of data is represented by T_k, and $T_k \epsilon$ T, the number of data accesses of the data block D_j and the data type T_k is represented by $F_{Vi}(D_j , T_k)$, and T_a represents all data types in the data block D_j. Through this number we can calculate the probability of data visitors accessing T_k data.

Using the calculation formula of information entropy [15], the amount of information obtained by the visitor V_i in the data section D_j.

$$P_{Vi}(T_k|D_j) = \frac{F_{Vi}(D_j, T_k)}{\sum_{T_a \epsilon T} F_{Vi}(D_j, T_a)} \tag{7}$$

$$H_{Vi} = -\sum_{k=1}^{T} P_{Vi}(T_k|D_j) \ln P_{Vi}(T_k|D_j). \tag{8}$$

Similarly, according to the historical access record, the average amount of information of all visitors V_{all} who ac-

Table 3: 6 risk factors of Vehicular Network Big Data

Risk factor	Weight
1. Characteristics of Visitor	
1.1 Role	$n_1 = 0.12$
1.2 Access Level	$n_2 = 0.12$
1.3 Previous Violations	$n_3 = 0.12$
1.4 Risk Quotas	$n_4 = 0.12$
2. Characteristics of Information and Requirements	
2.1 Sensitive level	$n_5 = 0.25$
2.2 Permission Level	$n_6 = 0.25$

cessed the data section D_j can be obtained.

$$\overline{H}(D_j) = \frac{H_{all}(D_j)}{C(V_{all})} \tag{9}$$

Among them, H_{all} (D_j) represents the total amount of information of V_{all}, and C (V_{all}) represents the total number of visitors. By comparing the information amount of the visitor Vi and V_{all}, the difference $Risk_{Vi}$ accessing the same data section D_j can be obtained, and then all the access section differences of the visitor V_i can be summed to obtain the risk threshold.

$$Risk_{Vi} = max\left\{0, H_{Vi}(D_j) - \overline{H}(D_j)\right\} \tag{10}$$

$$Risk_{Threshold} = \sum_{T_a \epsilon T} Risk_{Vi}(T_a). \tag{11}$$

A_m is the kth risk quota allocation phase, $Q_{Vd}(A_m)$ is the access quota used by visitor d at this stage, V(A_m) is the total number of visitors in stage A_m, and Formula (12) is the average. In the m + 1 risk quota allocation phase, the quota to be allocated is determined by the average of the quota consumption of the previous m stages. It is considered that the average of the quota consumption of the first m stages is a sample of a normal distribution, and then the mean and the variance s of the distribution can be obtained. The quota to be allocated is in the range of [?- ns, ?+ ns], where n is selected according to the system. Then set the probability ?= [0, 1] as the risk tolerance threshold of the risk adaptive access control system. If the probability of the visitor exhausting the quota in the next stage is less than the visitor can be assigned a new quota.

$$E(A_m) = \frac{Q_{Vd}(A_m)}{V(A_m)} \tag{12}$$

4 Policy Determination and Execution

4.1 Policy Determination

The access request process introduces risk quantification mechanism and policy decision function, the code is Algorithm 1:

Algorithm 1 Introduce Risk and Policy Decision

```
1: < RuleRuleId = ""Effect = "Permit" >
2:  < Target > ··· < /Target >
3: <Condition FunctionId = http://research.
   sun.com/Projects/xacml/names/function\
   #Risk-quantification?
4:   <Apply FunctionId =rn:oasis:names:tc:
   xacml: 1.0: function:integer-one-and-only?
5:    <EnvironmentAttributeDEsignator
6:        DataType = http:www.w3.org/2001/
   XMLSchema\#integer?
7:   AttributeId =rn:oasis:names:tc:xacml:1.0:
   environment:riskThreshold?>
8:   </Apply>
9:    <AttributeValue = http:www.w3.org/2001/
   XMLSchema\#integer?
10:  </AttributeValue>
11:   </Condition>
12: </Rule>
```

This rule will be added to every strategy that requires risk determination. Its role is to quantify the access request by calling the method of risk assessment. When its condition is satisfied, the decision effect of the access in this rule is allowed. The Apply function gets the current system riskThreshold provided by the risk strategy.

Algorithm 2 Rule Quantification

```
1: Input: request
2: Output: ruleDecision
3: Begin
4: requestAttributes = PIP. requestAttributes
5: riskQuotas = PIP.riskQuotas
6: requestRisk = RG. quantify
7: riskThreshold = RP. riskThreshold
8: if (requestRisk < riskThreshold and riskQuotas >
   0) then
9:       ruleDecision = permit
10:      return ruleDecision.
11: end if
12:      Return noEffect
13: End
```

Table 4: Dataset metadata

	Heading	Type of data	Data Format
1	medallion	string	Text string format
2	hack license	string	Text string format
3	IDvendor id	string	Text string format
4	rate code	string	Text string format
5	store and forward flag	string	Text string format
6	pickup datetime	string	Time format YYYY/MM/dd
7	dropoff datetime	string	Time format YYYY/MM/dd
8	passenger count	int	Normal integer format
9	trip time in seconds	int	Normal integer format
10	trip distance	float	Normal floating point format
11	longitude coordinates for the pickup location	float	Normal floating point format
12	latitude coordinates for the pickup location	float	Normal floating point format
13	longitude coordinates for the dropoff location	float	Normal floating point format
14	latitude coordinates for the dropoff location	float	Normal floating point format

The specific method of determination is as follows: First, the parameters related to the attributes provided by PIP are passed to RE to calculate the risk value of the access request. Then determine whether the risk value is less than the riskThreshold and the risk quota is greater than zero. Finally decide whether the access request is allowed. Each risk determination rule will be judged by reference to the risk threshold.In other words, the risk threshold manages the acceptable risk level of the entire system. If the administrator wants information to flow more smoothly, that is, the system can accept a larger risk value, you can increase the value; If the system

Algorithm 3 Algorithm Decision

```
Input: request
Output: ruleDecision
Begin
policySet = PAP.match + RP.match
policy[] = policySet.match
for i = 1 to policy. quantity do
    // rule[] = policy [i] . match
    for  j = 1 to rule . quantity do
        // rule Decision [j] = rule[j].rulequantify.combine
        policy Result [i] = policy [i] . policyquantify.combine
        result = policySet. policyResult[i].combine
        return result
    end for
end for
End
```

administrator wants to be more careful about the flow of information, you can turn this value down.

The code is Algorithm 2.

The judgment result of each rule is merged by the merge logic preset by the policy.Finally, the judgment result of the strategy is obtained, and the corresponding obligation is added according to the judgment result.If there is a policy set at the top level, merge the decision results of each strategy with the merge logic preset by the policy set. Get the final judgment result and total obligation.

Table 5: Data visitor access log table

Heading	Type of data
name	string
age	int
gender	string
accountID	int
departmentID	longint
position	string
permission	string
risk quota	int
hisRisk	int
previous Violations	string
actionTime	string
action	string
path	string

The pseudo-code for the entire quantization process is Algorithm 3.

After the judgment phase is completed, the result information containing the judgment result and all the obligation are returned to the PEP. The PEP enters the next policy execution phase based on the content of the result information.

4.2 Policy Execution

Figure 3 shows the access control decision process.Similarly, the left side represents the XACML component and the right side represents the risk module. First, the principal issues an access request. Then the external

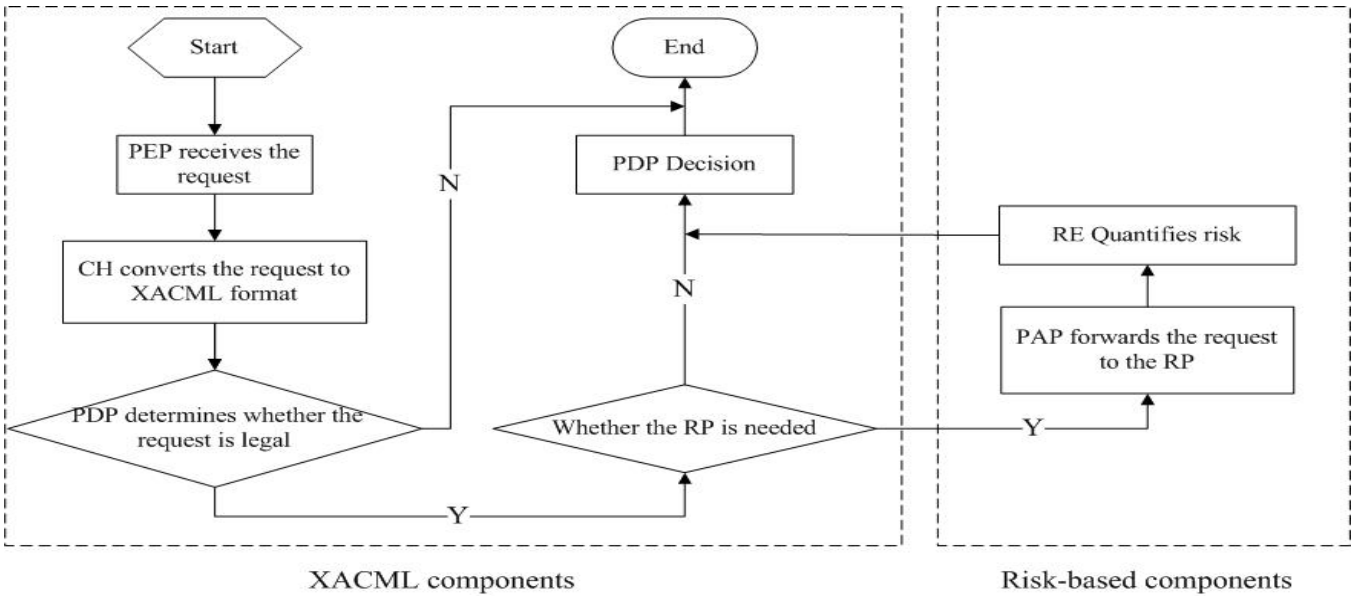


Figure 3: XACML extension framework

application passes the access request to the PEP.Finally, the PEP interacts with the external application. PEP sends an access request to CH. CH converts the access request format to XACML format and sends it to the PDP. The PDP is used to determine whether the access request is legal. The policy or policy set provided by the PAP is required in the decision, and the attribute information provided by the PIP is required. If the access request is illegal, it ends directly; If it is legal, it determines whether the access requires a risk policy. If a risk strategy is not required, the PDP evaluates directly; If required, the PAP makes a request to the risk policy. First, RE quantifies the risk and sends the result to the PDP. Then the PDP makes the decision and sends the result to the PEP. Finally, the PEP performs the relevant obligations.

5 Simulation

5.1 Experiment Setup

In the experiment, the taxi driving position record is used in this model to verify the privacy protection of the big data of the Internet of Vehicles. The data here comes from the real taxi detailed driving position data, including medallion, hack license , vendor id, rate code, store and for ward flag, pick up datetime, drop off datetime, passenger count, trip time in seconds, trip distance, latitude and longitude coordinates for the pickup location, latitude and longitude coordinates for the dropoff location and so on, the specific information is shown in Table 4 . We simulate access requests from two types of visitors, including each visitor's role, access rights, historical violations, risk quotas, et.al for each access record. The specific information is shown in Table 5. The visitor holds the access requirement to access the data, and finally, the risk value is calculated by the visitor's access record through the risk access control model.

5.2 Experimental Result

In the experiment, simulated the access history of 600 visitors as the experimental data, the information included in the history is shown in Table 5, it is about abnormal visitors, and the rests are normal visitors.Calculating the risk value for each visitor and sorting by risk.To test the effectiveness of the method, two indicators were examined.Accuracy rate represents the proportion of abnormal visitors among the top K visitors with the highest risk. Recall rate is the proportion of abnormal visitors in the top K visitors at all abnormal visitors. In each component module based on the XACML access control mechanism, the program is implemented in the Java language based on the Eclipse development platform. Important third-party development kits are based on SunXACML and the University of Murcia (UMU). The API of SunXACML implements the parsing and decision calculation of xacml.UMU uses the Java language to develop a UML-XACML-Editor V1.3.2 policy editor that supports the XACML 2.0 specification, which can be used to edit its own policy documents.

1) Experimental results under different visits. The experiment is mainly used to test the effect ofthe model on the number of different accesses requests. According to Formulas (2)-(11) risk value and risk threshold calculation method, the risk value and risk threshold of the visitor access data are calculated separately. As shown in Table 6, ESAV indicates that

Table 6: Results of 600 visitors

Visits	index	Quantity	Risk value	Risk threshold	Accuracy	Arain2017 Recall rate
5	ESAV	60		3.39	49/60	49/60
	AIAV	49	[4.27, 6.29]			
	AINV	551	[0.90, 3.28]			
10	ESAV	60		3.41	51/60	51/60
	AIAV	51	[4.09, 6.23]			
	AINV	549	[0.86, 3.26]			
15	ESAV	60		3.43	53/60	53/60
	AIAV	53	[3.77, 6.14]			
	AINV	547	[0.82, 3.23]			
20	ESAV	60		3.49	54/60	54/60
	AIAV	54	[3.68, 6.12]			
	AINV	546	[0.79, 3.10]			

Table 7: Risk Threshold calculation process when the number of visits is 5

	Sensitive data access times	Insensitive data access times	Total visits	entropy	amount of information	Risk threshold
D_1	189	267	456	0.00219298	0.67844549	3.393949
D_2	343	215	558	0.00179212	0.666601401	
D_3	467	511	978	0.0010225	0.692134799	
D_4	267	337	604	0.00165563	0.686416351	
D_5	165	230	395	0.00253165	0.679545906	

the number of abnormal visitors is set by the laboratory, AIAV indicates that the number of abnormal visitors identified by the algorithm, AINV indicates that the number of normal visitor identified by the algorithm.The experiment counts the identification of abnormal visitors under different access times. For example, when the number of visits is 5, the risk threshold is 3.39, and the risk value of abnormal visitors is between [4.27, 6.29]. The normal visitor's risk value is between [0.90, 3.28], the accuracy rate and the recall rate also reach 82% (49/60), and the accuracy and recall rate increase with the number of visits. It shows that the model in this paper can clearly distinguish two types of visitors, that is, the model is effective.

In the case of 5 visits, among the 5 data blocks, the sensitive data of the data block D_1 is accessed 189 times, the insensitive data is 267 times; The sensitive data of the data block D_2 is accessed 343 times, and the insensitive data is accessed 215 times; The sensitive data of the data block D_3 is accessed 476 times, and the insensitive data is accessed 511 times; The sensitive data of the data block D_4 is accessed 267 times, and the insensitive data is accessed 337 times; The sensitive data of the data block D_5 is accessed 165 times, and the insensitive data is accessed 230 times; According to Formulas (5)-(9) risk thresholds can be obtained when the number of visits is 5. The specific information is shown in Table 7.

In addition, this experiment also carried out extended statistics, which respectively counted the identification of abnormal visitors in the top 10, top 20, top 30, top 40 and top 50 highest risks. In Table 8 of the risk ranking results, the proportion of abnormal visitors in Top 10 is 100% (10/10), and in Top 50, our accuracy rate is also above 88% (44/50); In the case of recall rate, when the number of access log records of the system is 20 and K is 50, the recall rate is also above 78% (47/60), and the accuracy and recall rate both increase with the number of visits increase, which is because more visits can be more thorough understanding of the behavior and impact of visitors, and the calculated risk value is more accurate.

2) Experimental results under different abnormal visitor proportion. In this experiment, the number of visitors were still 600, mainly testing the identification of abnormal visitors at 5% (30 people), 10% (60 people), 15% (90 people) and 20% (120 people). And set the number of visits is 15 for per visitor, the test results are shown in Table 9. As can be seen from the table, the risk value of abnormal visitors is significantly higher than that of normal visitors. In this experiment, only the number of abnormal visitors is compared, so the accuracy and recall rate is the same in the same proportion. Moreover, as the propor-

Table 8: Accuracy and recall rate under different access times

Measure	Visits	K(Top K visitors with the highest risk value)				
		10	20	30	40	50
Accuracy	5	10/10	19/20	28/30	36/40	44/50
	10	10/10	20/20	29/30	37/40	45/50
	15	10/10	20/20	29/30	38/40	46/50
	20	10/10	20/20	29/30	39/40	47/50
Recall rate	5	10/60	19/60	28/60	36/60	44/60
	10	10/60	20/60	29/60	37/60	45/60
	15	10/60	20/60	29/60	38/60	46/60
	20	10/60	20/60	30/60	39/60	47/60

Table 9: Experimental results for different abnormal visitor ratios

Measure	The proportion of abnormal visitors to all visitors			
	5%(30)	10%(60)	15%(90)	20%(120)
Normal visitor risk value	[0.86,3.31]	[0.82,3.23]	[0.79,3.21]	[0.75,3.18]
Abnormal visitor risk value	[3.83,6.15]	[3.77,6.14]	[3.76,6.09]	[3.72,6.01]
Accuracy	25/30	53/60	81/90	110/120
Recall rate	25/30	53/60	81/90	110/120

tion of abnormal visitors increases, the accuracy and recall rate also increases from 83% (25/30) to 92% (110/120), and the overall performance of the model increases. Experiments show that this model is valid for different proportion of abnormal visitors.

6 Conclusion

Security risks adaptive access control for vehicular network big data is the theme of the paper, it combines the characteristics of XACML's powerful access policy expression capabilities to introduce risk extension XACML framework. It mainly introduces the process of determining and executing the risk quantification process and strategy. Finally, the effectiveness of the model is verified by simulation experiments. In a distributed environment, different enterprises or departments may have different requirements for authorization management, and they use different access control methods.The compatibility of multiple access control technologies must be considered during the development of dynamic authorization decision center.Later, we also need to test the time that takes for the visitor's request from the browser to the fully loaded and the delay in the number of risk metrics for the entire decision.

7 Acknowledgement

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

References

[1] A. Abdulaziz and El-Khatib Khalil, "On the possibility of insider threat prevention using intent-based access control (ibac)," *IEEE Systems Journal*, vol. 11, no. 2, pp. 373–384, 2017.

[2] M. Abomhara, G. M. Køien, V. A. Oleshchuk, and M. Hamid, "Towards risk-aware access control framework for healthcare information sharing," in *The 4th International Conference on Information Systems Security and Privacy (ICISSP'18)*, pp. 312–321, 2018.

[3] S. Arunkumar, M. Srivatsa, B. Soyluoglu, M Sensoy, and F. Cerutti, "Privacy enforcement through policy extension," in *The 35th IEEE Military Communications Conference*, pp. 1096–1100, 2016.

[4] A. Chen, H. Xing, K. She, and G. Duan, "A dynamic risk-based access control model for cloud computing," in *The 6th IEEE International Conference on Big Data and Cloud Computing*, pp. 579–584, 2016.

[5] A. Chattopadhyay, A. Nag, and K. Majumder, "Secure data outsourcing on cloud using secret sharing scheme," *International Journal of Network Security*, vol. 19, no. 6, pp. 912–921, 2017.

[6] Z. Chunfeng, *Research and Application on Unstructured Big Data Storage and Processing for Vehicle Network*, University of Science and Technology of China, Master Thesis, 2018.

[7] D. R. dos Santos, R. Marinho, G. R. Schmitt, C. M. Westphall, and C. B. Westphall, "A framework and risk assessment approaches for risk-based access control in the cloud," *Journal of Network and Computer Applications*, vol. 74, pp. 86–97, 2016.

[8] S. Hou, *The Research and Application of Access Control Model Based on Attribute in the Web Services*, Beijing University Of Technology, Master Thesis, 2017.

[9] Z. Hui, H. Li, M. Zhang, and D. Feng, "Risk-adaptive access control model for big data in health care," *Journal on Commu- nications*, vol. 36, no. 12, pp. 190–199, 2015.

[10] M. S. Hwang, T. H. Sun, and C. C. Lee, "Achieving dynamic data guarantee and data confidentiality of pub- lic auditing in cloud storage service," *Jou- rnal of Circuits Systems and Computers*, vol. 26, no. 5, pp. 1–17, 2017.

[11] K. P. Kibiwott, Z. Fengli, O. A. Anyembe, and D. Adu-Gyamfi, "Secure cloudlet- based ehealth big data system with fine-grained access control and outsourcing decryption from ABE," *International Journal of Network Security*, vol. 20, no. 6, pp. 1149–1162, 2018.

[12] K. Y. Lee, A. Kim, Y. E. Jeon, J. J. Kim, Y. S. Im, G. S. Choi, S. B. Park, Y. S. Lim, and J. J. Kang, "Spatio-temporal xacml: The expansion of xacml for access control," *International Journal of Security and Networks*, vol. 10, no. 1, pp. 56–63, 2015.

[13] J. Li, C. Peng, Y. Zhu, and H. Ma, "Risk access control model for hadoop," *Chinese Journal of Network and Information Security*, vol. 2, no. 1, pp. 46–52, 2016.

[14] A. Sara, T. Yassine, and M. Abdellatif, "Secure confidential big data sharing in cloud computing using KP-ABE," in *The 2nd International Conference on Big Data Cloud and Applications (BDCA'17)*, 2017. ISBN: 978-1-4503-4852-2.

[15] Z. Song, H. Wang, H. Zhao, Y. and Chen, "Method and application for multi-scenario hybrid risk decision making based on utility-risk entropy," *Systems Engineering and Electronics*, vol. 40, no. 12, pp. 2751–2757, 2018.

[16] J. Xu, Z. Liu, S. Li, B. Qiao, and G. Tan, "A cloud-user behavior assessment based dynamic access control model," *Inter- national Journal of Systems Assurance Engi- neering and Management*, vol. 8, pp. 1966–1975, 2017.

[17] Y. Yang, *Design and Implementation of Security Enhanced Cloud Storage Access Control System Based on Attributes*, Beijing Institute of Technology, Master Thesis, 2016.

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.

Hong-jin Fan was born in Mar. 1993. He is a master student at Lanzhou University of Technology. His major research field is Security on big data of vehicular network. E-mail:fan_hjin@163.com.

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

Yan Yan was born in Oct. 1980. She is a associate professor and a supervisor of master student at Lanzhou University of Technology. Her major research field is privacy protection, multimedia information security. E-mail: yanyan@lut.cn.

Guo-qiang Ma was born in Jun. 1992. He is a master student at Lanzhou University of Technology. His major research field is network and information security. Hwang2005ijnsaesE-mail: magq1514@163.com.

Xue-ming Han was born in Jan. 1990. He is a master student at Lanzhou University of Technology. His major research field is network and information security. E-mail: hxmhan@163.com.

An Enhanced Secure Smart Card-based Password Authentication Scheme

Hsieh-Tsen Pan[1], Hung-Wei Yang[1], and Min-Shiang Hwang[1,2]
(Corresponding author: Min-Shiang Hwang)
Department of Computer Science & Information Engineering, Asia University, Taichung, Taiwan[1]
Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan[2]
(Email: mshwang@asia.edu.tw)
(Received Mar. 3, 2019; Revised and Accepted Dec. 12, 2019; First Online Feb. 28, 2020)

Abstract

The development of world communication and information technology is very advanced. The use of the Internet and smart cards makes it easier for users to conduct remote transactions, and security factors are the key to successful remote users' transactions. In this case, the authentication process is critical to maintaining confidentiality when transactions use public channels. Recently, Moon et al. proposed an efficient and secure smart card based password authentication scheme. They claimed that their scheme is more secure and practical as a remote user authentication scheme. However, Irawan and Hwang found that the Moon et al.'s scheme was still unable to withstand gussing identity attacks and user impersonation attacks. To address this security hole, we propose a new authentication scheme and a key with a smart card in this article. In addition, we show that the proposed authentication scheme is highly resistant to various attacks. Finally, we compare the performance and functionality of the proposed scheme with other related schemes.

Keywords: Password; Smart Card; User Authentication

1 Introduction

Everyone needs security at home, in the office, on the street, and everywhere, because it enables people to use security systems safely and prevent things that shouldn't happen. The safety system should be flexible, cheap, and work continuously without being limited by working hours. With the rapid development of cloud computing, more and more applications and services have been provided, such as cloud storage services, cloud resources, shared computing, and so on [1, 2, 9, 12, 20, 21, 24, 28]. Smart card RFID is an advanced information technology embedded in a card as an information storage medium [8, 25, 26]. At present, the implementation of smart cards has spread to almost all fields, whether it is used in the attendance of hotels, homes, offices and educational institutions, or strict data security.

A user authentication scheme is a mechanism by which a server authenticates users before allowing them to access resources or services provided by the server [14]. To date, many user authentication schemes have been proposed [3, 4, 6, 17, 27]. However, most of these schemes have advantages and disadvantages. In 2012, Yoon et al. proposed a remote user authentication scheme [30], which is an improvement on the scheme of Liaw et al. [19]. However, Chen et al. found that their scheme was not secure enough [7]. In 2012, Li et al. proposed a YS-like user authentication scheme using smart cards [18]. However, Feng et al. found the security of their scheme was vulnerable to the password guessing attack [10]. In 2014, Huang et al. proposed a timestamp-based user authentication with smart card [13]. However, Feng et al. showed that their scheme is vulnerable to the password guessing attack [11]. In 2014, Zhuang et al. proposed a password authentication scheme based on geometric hash function without using smart card [31]. However, Chen showed that their scheme is also vulnerable to the password guessing attack [5].

In 2017, Liu et al. proposed a more secure and practical remote user authentication scheme [22]. However, Moon et al. found that their scheme was still unable to withstand external attacks and offline password guessing attacks [23]. To overcome these security loopholes, Moon et al. also proposed an ECC-based authentication and key agreement scheme using smart cards. Utilizing the lightweight calculation of ECC (Elliptic Curve Cryptography System) [15, 29], Moon et al.'s scheme is both practical and easy to implement. However, in 2018, Irawan and Hwang discovered a security hole in Moon et al.'s two-factor authentication scheme [16]. They showed that Moon et al.'s scheme was actually unable to resist anonymous interception and user impersonation attacks. To overcome these security loopholes, we propose an improved biometric-based authentication and key agreement scheme using smart cards. In addition, we will prove that the proposed authentication scheme is more resistant to

various attacks than other related schemes.

For more details, we divide this article into the following five sections: In Section 1, we briefly introduce our research motivations. In Section 2, we briefly reviewed the weaknesses of Moon et al.'s password authentication scheme. In Section 3, we propose a new authentication scheme. The security and performance analysis of the proposed scheme is given in Section 4, and the conclusions of this paper are given in Section 5.

2 The Weaknesses of Moon et al.'s Scheme

In 2018, Irawan and Hwang found that the Moon et al.'s scheme was unable to withstand gussing identity attacks and user impersonation attacks [16]. In this section, we briefly review the attacks proposed by Irawan and Hwang as follows:

Gussing Identity Attack:

Moon et al.'s scheme [23] did not hide the identity ID of user U_i during the login and authentication phases, User $\rightarrow$ Server: $\{AID_i, D_i, E_i, F_i, T_i\}$, Server $\rightarrow$ User: $\{F_i, G_i, T_s\}$. Attackers can easily guess or steal it from unsecured public channels. The attacker can then check $h(ID_i'||(AID_i \oplus ID_i')||F_i||T_s) \stackrel{?}{=} G_i$.

User Impersonation:

Knowing the user ID_i (guest identity) of the first attack, the attacker will send the user ID_i to the server S through a public channel. During the login phase of Attacker $(ID) \rightarrow$ Server: $\{AID_i, D_i, E_i', F_i', T_i\}$, the server will calculate $F_i' = h(ID_i||h(A_i)||E_i'||T_i)$, it is considered a legitimate user.

3 The Proposed Scheme

In this section, we propose a scheme to improve Moon et al., called a new biometric-based password authentication scheme using smart cards [23]. We modify some procedures during registration, login, and authentication phases. In the improved scheme, there are also two participants, namely the i^{th} user U_i and server S.

3.1 Registration Phase

At the beginning of the improved Moon et al.'s scheme, the server S selects x, E, P, and $h(\cdot)$. Here, x denotes a master secret key stored in S; P denotes a base point of the elliptic curve E; and $h(\cdot)$ denotes a collision-resistant hash function. The user U_i then registers with the server S by the following steps and Figure 1:

Step 1. U_i prints personal biometric information BIO_i on the device sensor. Then, the device sensor scans BIO_i, extracts (R_i, P_i) from $Gen(BIO_i) \rightarrow (R_i, P_i)$, and stores P_i in memory. Here, R_i and P_i denote almost random binary strings and U_i's auxiliary binary strings, respectively. Next, U_i selects identity ID_i and password PW_i, and calculates $RPW_i = h(PW_i||R_i)$. Finally, U_i sends a registration request message $\{ID_i, RPW_i\}$ to S over the secure channel.

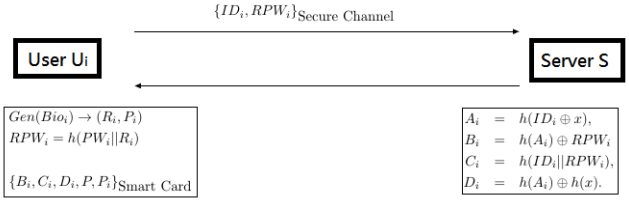


Figure 1: The registration phase of the proposed scheme

Step 2. After receiving the registration request message from U_i, the server S verifies whether ID_i is valid and calculates the following parameters:

$$\begin{aligned} A_i &= h(ID_i \oplus x), \\ B_i &= h(A_i) \oplus RPW_i \\ C_i &= h(ID_i||RPW_i), \\ D_i &= h(A_i) \oplus h(x). \end{aligned}$$

Here, $\oplus$ denotes an exclusive-or operation; and $\|$ denotes a concatenation operation.

Step 3. The server S stores the data $\{B_i, C_i, D_i, h(\cdot), P\}$ on the new smart card, and issues the smart card to the user U_i through a secure channel.

Step 4. The user U_i stores the random number P_i into the smart card.

3.2 Login Phase

After the registration phase is performed, the user will proceed with the login phase to invoke the U_i user to log in to the server S. The steps in this phase are described below and Figure 2.

Step 1. U_i inserts his/her smart card into the card reader, enters ID_i and password PW_i, and then prints biometric information BIO_i^* on the sensor. The sensor then sketches the BIO_i^* and recovers R_i from $Rep(BIO_i^*, P_i) \rightarrow (R_i, BIO_i^*)$.

Step 2. The smart card first calculates two parameters: $RPW_i = h(PW_i||R_i)$ and $C_i' = h(ID_i||RPW_i)$. The smart card then checks if C_i' is equal to the stored C_i. If it is true, the smart card proceeds to Step 3; otherwise, Step 3 is performed. Otherwise, the smart card will terminate this session.

Step 3. The smart card randomly generates a number α

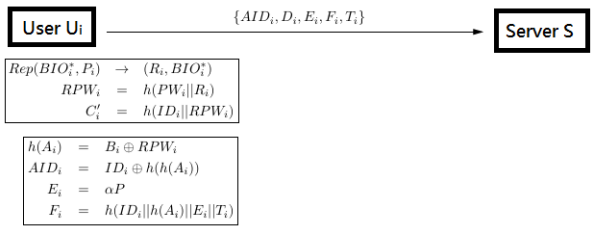


Figure 2: The login phase of the proposed scheme

and calculates the following parameters:

$$\begin{aligned} h(A_i) &= B_i \oplus RPW_i \\ AID_i &= ID_i \oplus h(h(A_i)) \\ E_i &= \alpha P \\ F_i &= h(ID_i||h(A_i)||E_i||T_i), \end{aligned}$$

where T_i is the current timestamp of user U_i.

Step 4. The smart card sends a login request message $\{AID_i, D_i, E_i, F_i, T_i\}$ to the server S.

3.3 Authentication Phase

After completing this phase, the user U_i and the server S can authenticate each other and establish a shared session key for subsequent secret communication. The steps in the certification phase are as follows and Figure 3:

Step 1. The server S verifies $T_i' - T_i \leq \Delta T$, where T_i' is the time to receive the login request message, and ΔT is the valid time threshold. If both conditions are true, the server S proceeds to Step 2; otherwise, the server S proceeds to Step 2. Otherwise, the server S rejects the login request.

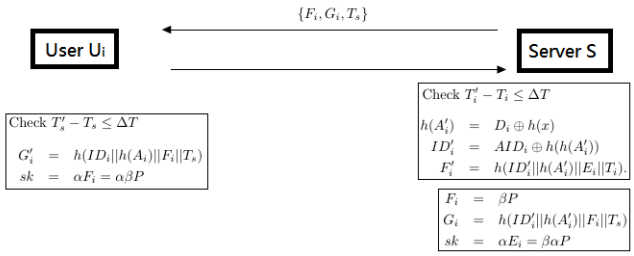


Figure 3: The authentication phase of the proposed scheme

Step 2. The server S calculates the following parameters:

$$\begin{aligned} h(A_i') &= D_i \oplus h(x) \\ ID_i' &= AID_i \oplus h(h(A_i')) \\ F_i' &= h(ID_i'||h(A_i')||E_i||T_i). \end{aligned}$$

The server S then compares whether F_i' is equal to F_i. If it is true, the server S confirms that the user U_i is valid and the login request is accepted; otherwise, the server S confirms that the user U_i is valid. Otherwise, the server S rejects the login request.

Step 3. Next, server S randomly generates a number β and calculates the following parameters: $F_i = \beta P$, $G_i = h(ID_i'||h(A_i')||F_i||T_s)$, where T_s is server S The current timestamp.

Step 4. The server S sends a mutual authentication message $\{F_i, G_i, T_s\}$ to the user U_i.

Step 5. Upon receiving the message $\{F_i, G_i, T_s\}$ from S, the user U_i checks the validity of T_s. If $T_s' - T_s \leq \Delta T$, where T_s' is the time to receive the mutual authentication message, the user U_i proceeds to Step 6; otherwise, the user U_i proceeds to Step 6. Otherwise, the user U_i terminates the connection.

Step 6. The user U_i calculates $G_i' = h(ID_i||h(A_i)||F_i||T_s)$, and then checks whether G_i' is equal to the received G_i. If it is true, the validity of the server S is verified; otherwise, the session is terminated.

Step 7. Finally, the user U_i and the server S construct a shared session key $sk = \alpha\beta P$ to ensure secret communication.

3.4 Password Change Phase

During the password change phase, U_i can update the password without any help from server S. This phase includes the following steps:

Step 1. U_i enters his/her identity ID_i and password PW_i, and print biometric information BIO_i^* on the sensor. The sensor then scans BIO_i^* and recovers R_i from $Rep(BIO_i^*, P_i) \rightarrow R_i$.

Step 2. Next, SC_i calculates $RPW_i = h(PW_i||R_i)$ and checks if $h(ID_i||RPW_i)$ is equal to the stored C_i. If it does, the smart card will ask U_i for the new password; otherwise, SC_i terminates the password change phase immediately.

Step 3. U_i enters a new password PW_i^{new}, smart card further calculates $RPW_i^{new} = h(PW_i^{new}||R_i)$, $B_i^{new} = B_i \oplus RPW_i \oplus RPW_i^{new}$ and $C_i^{new} = C_i \oplus RPW_i \oplus RPW_i^{new}$.

Step 4. Finally, the smart card replaces B_i with B_i^{new} and C_i with C_i^{new} in memory.

4 Security and Performance Analysis of the Proposed Scheme

The improved scheme retains the advantages of the Moon et al.'s scheme [23] and can withstand many types

Table 1: Functionality comparison of the proposed scheme and Moon et al.'s scheme

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Moon et al. [23]	○	○	○	○	○	○	○	○	×	×
The proposed	○	○	○	○	○	○	○	○	○	○

F1: Mutual authentication; F2: Session key agreement; F3: Freely chosen and exchanged password; F4: Withstanding man in the middle attack; F5: Withstanding insider attack; F6: Withstanding replay attack; F7: Providing perfect forward secrecy; F8: Satisfying known-key security; F9: Guessing identity attack; F10: User impersonation attack.

of possible attacks, such as resistance to outsider attacks, insider attacks, user impersonation attacks, and perfect forward secrecy. In this section, we show that the improved scheme can resist gussing identity attacks and user impersonation attacks discovered by Irawan and Hwang [16] and described in Section 2.

4.1 Resisting Gussing Identity Attack

In Moon et al.'s scheme [23], the attacker could intercept $\{AID_i, D_i, E_i, F_i, T_i\}$ in login phase and $\{F_i, G_i, T_s\}$ in authentication phase. The attacker can guess an identity ID_1' and check $h(ID_i'||(AID_i \oplus ID_i')||F_i||T_s) \stackrel{?}{=} G_i$. If the equation holds, the attacker has already guessed the identity ID_i of the user, otherwise, the attacker will repeatedly guess and check other possible identities ID_i'. The main problem is

$$\begin{aligned} G_i &= h(ID_i||h(A_i)||F_i||T_s) \\ &= h(ID_i||(AID_i \oplus ID_i)||F_i||T_s). \end{aligned} \quad (1)$$

Once the attacker knows G_i, AID_i, F_i, and T_s, the attacker can guess ID_i' to satisfy Equation (1).

In the improved scheme, AID_i and ID_i are

$$\begin{aligned} AID_i &= ID_i \oplus h(h(A_i)) \\ h(A_i') &= D_i \oplus h(x) \\ ID_i' &= AID_i \oplus h(h(A_i')). \end{aligned}$$

In the proposed scheme,

$$\begin{aligned} G_i &= h(ID_i||h(A_i)||F_i||T_s) \\ &\neq h(ID_i||(AID_i \oplus ID_i)||F_i||T_s). \end{aligned}$$

It is difficult to obtain $h(A_i)$ from the intercepted $\{AID_i, D_i, E_i, F_i, T_i\}$ during the login phase and $\{F_i, G_i, T_s\}$ during the authentication phase. Thus, the proposed scheme can resist the guessing identity attacks.

4.2 Resisting User Impersonation Attack

In Section 2, we describe that Moon et al.'s scheme cannot resist this guessing identity attack. If the attacker can guess the identity of the legitimate user ID_i, the attacker will impersonate the legitimate user by guessing the identity. In Moon et al.'s scheme, the attacker knows the user ID_i by guessing identity attack, the attacker will impersonate the user ID_i to the server S. During the login phase, the attacker sends $\{AID_i, D_i, E_i', F_i', T_i\}$ to the server. The server will check $F_i' = h(ID_i||h(A_i)||E_i'||T_i)$, so it will be treated as a legitimate user.

Since the proposed scheme can resist identity guessing attacks, the proposed scheme does not have the weakness of the user impersonation attack discovered by Irawan and Hwang [16].

4.3 Performance Analysis

In this section, we compare the functionality between the proposed scheme and the Moon et al.'s scheme in Table 1. If you are interested in comparison with other latest solutions, please refer to [23].

We compare the computational cost between the proposed scheme and the Moon et al.'s scheme in Table 2. If you are interested in comparison with other latest solutions, please refer to [23]. It can be seen from the comparison that the hashing cost of this scheme is slightly higher than that of Moon et al. scheme. Because the coputational cost of ECC operation is much larger than the coputational cost of hash functions and XOR operations. Therefore, we can ignore the computational cost of hash functions and XOR operations. In other words, the computational cost of the proposed scheme is almost equal to that of Moon et al.

5 Conclusion

In this paper, we have proposed an improved Moon et al.'s scheme. We also show that the proposed scheme can against the guessing identity attack and the user impersonation attack.

Acknowledgments

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

Table 2: Computational cost comparison of the proposed scheme and other related schemes

	C1	C2	C3	C4	C5	C6	Total
Moon et al. [23]	1H+1F	4H+4X	5H+1F+2P+2X	4H+1F+2P+3X	3H+1F+4X	-	17H+4F+4P+13X
The proposed	1H+1F	5H+3X	6H+1F+2P+2X	5H+1F+2P+2X	3H+1F+4X	-	20H+4F+4P+11X

C1: Computational cost of the user in registration phase; C2: Computational cost of the server in registration phase; C3: Computational cost of the user in login and authentication phases; C4: Computational cost of the server in login and authentication phases; C5: Computational cost of the user in password change phase; C6: Computational cost of the server in password change phase; H: Hashing operation; E: Modulus exponential operation; S: Symmetric encryption/decryption operation; M: Multiplication/division operation; Null: P: ECC operations; X: XOR operations; F: Fuzzy extraction; Null: Cannot provide this functionality.

References

[1] D. S. AbdElminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40–48, 2018.

[2] M. H. R. Al-Shaikhly, H. M. El-Bakry, and A. A. Saleh, "Cloud security using Markov chain and genetic algorithm," *International Journal of Electronics and Information Engineering*, vol. 8, no. 2, pp. 96–106, 2018.

[3] M. Bayat, M. B. Atashgah, M. Barari, and M. R. Aref, "Cryptanalysis and improvement of a user authentication scheme for internet of things using elliptic curve cryptography," *International Journal of Network Security*, vol. 21, no. 6, pp. 897–911, 2019.

[4] S. Q. Cao, Q. Sun, and L. L. Cao, "Security analysis and enhancements of a remote user authentication scheme," *International Journal of Network Security*, vol. 21, no. 4, pp. 661–669, 2019.

[5] S. M. Chen, C. S. Pan, M. S. Hwang, "Cryptanalysis and improvement of Zhuang-Chang-Wang-Zhu password authentication scheme", in *The 2nd Congress on Computer Science and Application (CCSA'14)*, pp. 118–123, 2014.

[6] T. Y. Chen, C. C. Lee, M. S. Hwang, J. K. Jan, "Towards secure and efficient user authentication scheme using smart card for multi-server environments," *Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, Nov. 2013.

[7] T. Y. Chen, C. H. Ling, M. S. Hwang, "Weaknesses of the Yoon-Kim-Yoo remote user authentication scheme using smart cards", in *IEEE Workshop on Electronics, Computer and Applications*, pp. 771–774, 2014.

[8] 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.

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

[10] T. H. Feng, W. Y. Chao, and M. S. Hwang, "Cryptanalysis and improvement of the Li-Liu-Wu user authentication scheme", in *International Conference on Future Communication Technology and Engineering (FCTE'14)*, pp. 103–106, 2014.

[11] T. H. Feng, C. H. Ling, M. S. Hwang, "An improved timestamp-based user authentication scheme with smart card", in *The 2nd Congress on Computer Science and Application (CCSA'14)*, pp. 111–117, 2014.

[12] W. F. Hsien, C. C. Yang and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133-142, 2016.

[13] H. F. Huang, H. W. Chang, P. K. Yu, "Enhancement of timestamp-based user authentication scheme with smart card," *International Journal of Network Security*, vol. 16, pp. 463–467, 2014.

[14] M. S. Hwang, Li-Hua Li, "A New Remote User Authentication Scheme Using Smart Cards", *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, Feb. 2000.

[15] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves", *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 73–84, 2004.

[16] B. Irawan, M. S. Hwang, "The weakness of Moon et al.'s password authentication scheme", in *3rd Annual International Conference on Information System and Artificial Intelligence (ISAI'18)*, Journal of Physics: Conference Series, vol. 1069(1), pp. 012070, 2018.

[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] J. Li, S. Liu, S. Wu, "Cryptanalysis and improvement of a YS-like user authentication scheme", *International Journal of Digital Content Technology and its Applications*, vol. 7, no. 1, pp. 828–836, 2012.

[19] H. T. Liaw, J. F. Lin, and W. C. Wu, "An efficient and complete remote user authentication scheme us-

ing smart cards," *Mathematical and Computer Modelling*, vol. 44, no. 1-2, July 2006.
[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] L. Liu, Z. Cao, C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, pp. 29–35, 2018.
[22] Y. Liu, C. C. Chang and S. C. Chang, "An efficient and secure smart card based password authentication scheme," *International Journal of Network Security*, vol. 19, no. 1, pp. 1–10, 2017.
[23] J. Moon, D. Lee, J. Jung, D. Won, "Improvement of efficient and secure smart card based password authentication scheme," *International Journal of Network Security*, vol. 19, no. 6, pp. 1053-1061, 2017.
[24] S. Rezaei, M. A. Doostari, M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115–131, 2018.
[25] C. H. Wei, M. S. Hwang, A. Y. H. Chin, "A mutual authentication protocol for RFID", *IEEE IT Professional*, vol. 13, no. 2, pp. 20–24, Mar. 2011.
[26] C. H. Wei, M. S. Hwang, Augustin Y. H. Chin, "A secure privacy and authentication protocol for passive RFID tags," *International Journal of Mobile Communications*, vol. 15, no. 3, pp. 266–277, 2017.
[27] H. Wijayanto, M. S. Hwang, "Improvement on timestamp-based user authentication scheme with smart card lost attack resistance," *International Journal of Network Security*, vol. 17, no. 2, pp. 160–164, 2015.
[28] C. Yang, Q. Chen, Y. Liu, "Fine-grained outsourced data deletion scheme in cloud computing," *International Journal of Electronics and Information Engineering*, vol. 11, no. 2, pp. 81–98, 2019.
[29] C. C. Yang, T. Y. Chang, M. S. Hwang, "A new anonymous conference key distribution system based on the elliptic curve discrete logarithm problem", *Computer Standards and Interfaces*, vol. 25, no. 2, pp. 141-145, 2003.
[30] E. J. Yoon, S. H. Kim, and K. Y. Yoo, "A security enhanced remote user authentication scheme using smart cards," *International Journal of Network Security*, vol. 8, no. 5, pp. 3661–3675, 2012.
[31] X. Zhuang, C.C. Chang, Z.H. Wang, Y. Zhu, "A simple password authentication scheme based on geometric hashing function," *International Journal of Network Security*, vol. 16, pp. 271–277, 2014.

Biography

Hsien-Tsen Pan received B.S. in Business Administration From Soochow University, Taiwan in 1999; M.S. in Information Engineering, Asia University, Taiwan in 2015; Doctoral Program of Information Engineering, Asia University, Taiwan from 2015 till now. From 2011 to 2014, he was the manager in Enterprise Service Chunghwa Telecom South Branch Taichung Taiwan. From 2014 to 2017, he was the operation manager in Medium division Taiwan Ricoh Co., Ltd. Taichung Taiwan From 2017 Sep 20 he is the Apple MDM Server Service VP in Get Technology Co.Ltd. Taipei Taiwan.

Hung-Wei Yang received B.S. in Industry Engineer From Da-Yeh University, Taiwan in 2001; M.S. in Information Management, Chao Yang University, Taiwan in 2009; Doctoral Program of Information Engineering, Asia University, Taiwan from 2016 till now. From 2012 to 2014, he was the manager in International Business Machine. From 2014 to 2015, he was the manager in Cisco Systems, Inc. Taiwan branch. From 2016 to 2019 he is the sales director of China branch in Syntron Technology Co. Ltd. Taipei Taiwan .From 2020 he is channel director in M-Power Co. Ltd., Taipei Taiwan.

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

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.