IJNS
International Journal of Network Security

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

Vol. 19, No. 3 (May 2017)

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

International Journal of Network Security

Vol. 19, No. 3 (May 1, 2017)

A Secret Image Sharing with Deep-steganography and Two-stage Authentication Based on Matrix Encoding

Li Liu[1], Anhong Wang[1], Chin-Chen Chang[2] and Zhihong Li[1]

(Corresponding author: Anhong Wang, Chin-Chen Chang)

College of Electronic Information and Engineering, Taiyuan University of Science and Technology[1]
No. 66, Waliu Rd., Taiyuan, China
Department of Information Engineering and Computer Science, Feng Chia University[2]
No. 100, Wenhwa Rd., Seatwen, Taichung, Taiwan
(Email: wah_ty@163.com, alan3c@gmail.com)

(Received Mar. 26, 2016; revised and accepted May 10 & June 5, 2016)

Abstract

A secret image sharing with deep-steganography and two-stage authentication based on matrix encoding is proposed. This scheme firstly employed Wang and Su's sharing scheme to share a secret image into n shadows. Then, n shadows were embedded into the pre-selected cover images by using matrix encoding so as to generate n stego images. Different from the common schemes which directly use LSBs replacement, our scheme performs deep-steganography, which hides 3-bit secret data but only modifies at most 1-bit via matrix encoding. Benefiting from the comparatively slight modification to the cover image, our scheme obtains enhancement in both the security of secret data and the visual quality of the stego image. As far as the authentication ability is concerned, our scheme explores a two-stage authentication: the first uses a sub-key to pass identity authentication, and the second performs tamper authentication. The experimental results show that our scheme achieves better authentication ability than the referenced schemes.

Keywords: Deep-steganography, matrix encoding, secret sharing, two-stage authentication

1 Introduction

With the rapid development and wide application of Internet, information hiding technology has come to play an important role in the field of information security. As a new information hiding technology, steganography hides the secret information into the cover image and generates a stego image [15]. Since the secret information is invisible in the stego image, this ensures that any third party (i.e., aside from the communication parties) is not aware of the existence of secret information.

Compared with traditional encryption, steganography provides a new solution to avoid any suspicion of attackers when the data is transmitted. Many steganographic methods [1, 3, 9, 11, 19] have been proposed to produce meaningful stego image which hides secret information. However, this stego image is usually held by only one person without extra copies. Thus, if the stego image is lost accidentally or modified intentionally, the secret information will be destroyed. Hence, it is necessary to share certain secret information among several people [5]. The concept of (k, n)-threshold secret sharing proposed by Shamir [14] can solve this problem. This sharing scheme divides a secret message into n shadows. The secret can be restored if $k(k \leq n)$ of the n shadows are obtained, but any $k-1$ or fewer of them will obtain no information about the secret image.

In 1995, Naor and Shamir [13] introduced this sharing idea into image field, and later Thien and Lin [17] used this idea to generate n noise-like images (so called 'shadows') with size $1/k$ of the secret image. Because smaller shadows were needed for easy transmission, storage and hiding, Wang and Su [18] proposed an improved scheme using Huffman coding to compress the difference image of the secret image, with each generated shadow image being about 40% smaller than that of the method in [17]. However, these noise-like images tended to make attacker suspicious.

In order to further improve the security of secret image, steganography based on secret image sharing schemes [2, 6, 12, 16, 20, 21] have been proposed to hide the noise-like shadows inside pre-selected meaningful images (called 'cover images'), and consequently generate the meaningful stego images to represent shadows so that no one can detect the existence of the secret information. Here, the challenge is to create high-quality stego im-

ages so that the modifications are not visually perceptible. Moreover, it will be useful to check in advance the fidelity of all stego images before they are used to reconstruct the secret images. Since fake stego images that were accidentally or intentionally submitted by the participant will lead to unsuccessful secret reconstruction. The scheme of steganography and authentication based on secret image sharing was firstly proposed by Lin and Tsai [12] in 2004. However, this scheme has a lossy reconstructed secret image and a weak authentication mechanism. Afterwards, Yang et al. proposed a scheme [20] that overcomes the weakness of that proposed by Lin et al. and enhances the authentication to some degree. However, this scheme reduces the visual quality of the stego images because the lowest 4-bit of the pixel value in each block is modified.

In Chang et al.'s scheme [2], the Chinese remainder theorem (CRT) is used to generate four authentication bits in order to obtain better authentication ability. Meanwhile, they claimed that their scheme enhances the visual quality of stego images. In [6], the concept of linear cellular automata was employed instead of Shamir's (k, n) threshold image sharing scheme, and this scheme not only obtains better visual quality but also improves the ability of tamper detection. Nevertheless, this scheme needs to offer at least k consecutive shadows to reveal the secret image, and cannot accurately locate the tampered blocks.

In all of the aforementioned schemes, the cover images are first divided into fixed-size blocks (of size 2×2) and then the pixels of shadows are hidden orderly into the LSBs of each block. Note that there are two problems.

Firstly, it is possible that only a part of a cover image will be used to hide secret data, so the statistical property of stego images shows a great difference between the two parts of non-embedded and embedded data; secondly, the RS steganalysis [8, 10] can easily detect the existence of secret data that has been embedded by LSBs and can predict its length. Hence, if an attacker wanted to steal the embedding positions of secret data, he could obtain the secret data easily. In this paper, we propose a deep-steganography method using matrix encoding. Firstly, Wang and Su's sharing scheme was employed to share secret image into n shadows.

Secondly, n shadows were embedded into the preselected cover images by using matrix encoding to generate the n stego images. Note that, in order to reduce the change in the statistical property of cover images, the full capacity of the cover images was used to realize the process of randomly embedded data. Since secret data does not directly appear in the pixels of the cover image, we call it 'deep-steganography'. Benefiting from the comparatively slight modification to the cover image, our scheme obtains enhancement in both the security of secret data and the visual quality of stego image. Furthermore, the authentication ability of our scheme can be well expressed by using two-stage authentication.

2 Related Works

2.1 Wang and Su's Image Sharing Scheme

Wang and Su [18] used a Huffman coding for a (k, n)-threshold secret image sharing scheme in order to generate smaller shadows and reconstruct the secret image completely.

The difference image DIFF=$\{diff_{ij}\}$ is calculated using Equation (1) and then encoded using the Huffman coding scheme.

$$diff_{ij} = \begin{cases} se_{ij}, & if \;\; i = 0 \;\; and \;\; j = 0 \\ se_{ij} - se_{(i-1)j}, & if \;\; i \neq 0 \;\; and \;\; j = 0 \\ se_{ij} - se_{i(j-1)}, & otherwise \end{cases} \quad (1)$$

where se_{ij} is the pixel value of the original secret image (sized $m \times n$) in the i-th row and j-th column, $1 \leq i \leq m, 1 \leq j \leq n$. Every t bits that obtained from the Huffman output stream are converted into the decimal sharing coefficient $(a_0, a_1, \ldots, a_{k-1})$, and every k coefficients are used to generate the $(k-1)$ degree polynomial sharing function.

$$f_l(x) = (a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}) \mod 2^t. \quad (2)$$

Each shadow S_i can be derived as

$$S_1 = f_l(1), \ldots, S_i = f_l(i), \ldots, S_n = f_l(n). \quad (3)$$

where the number of l depends on the number of decimal sharing coefficients. Obviously, if fewer than k shadows are received, the original secret image cannot be revealed.

In the revealing phase, the sharing coefficients $a_0, a_1, \ldots, a_{k-1}$ of $f_l(x)$ can be calculated using Lagrange's interpolation when giving any k of n shadows, and then all of the sharing coefficients are transformed into binary bit stream. This bit stream was decoded by using the Huffman codes table to reveal the difference image, and then the inverse-differencing process was used to reveal the whole secret image.

2.2 Matrix Encoding

Matrix encoding, which was introduced by Crandall [4], reduces the number of changes required and improves the embedding efficiency to a great extent in comparison with the classic LSB replacement methods. Assume the embedding cell is expressed in the form of vector $m = \{m_1, m_2, \ldots, m_q\}$ with the length of q. The coding implemented on each embedding cell is denoted by an ordered triple (d_{max}, q, t), where q is the number of modifiable bit positions in an embedding cell, and t is the bit length of secret information $s = \{s_1, s_2, \ldots, s_t\}$. Usually, we set $d_{max} = 1$, namely, an embedding cell with q positions will be changed in no more than d_{max} positions to embed t bits of secret information [7].

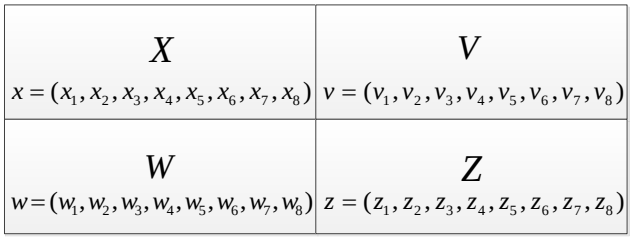


Figure 1: Structure of the four-pixel block in the cover image

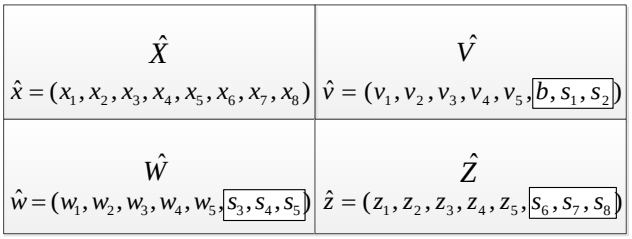


Figure 2: Block structure of the stego image in Lin and Tsai's scheme

A hash function $f(\cdot)$ is defined as Equation (4) to map q bits embedding cell m into t bits binary string.

$$f(m) = \bigoplus_{i=1}^{q} m_i \cdot i. \tag{4}$$

Subsequently, XOR operation is implemented on the value of the hash function and secret information s to find the position P of the changes required.

$$P = s \oplus f(m). \tag{5}$$

Finally, an embedded stego data m' is generated by the following rule:

$$m' = \begin{cases} m, & P = 0 \\ m_1, \cdots, \bar{m}_i, \cdots, m_q & P = i \end{cases}, \tag{6}$$

where $\bar{m}_i$ is the negation of m_i. In the secret extraction phase, the receiver would retrieve the secret information s by directly putting the stego data m' into the same hash function, i.e., $s = f(m')$.

2.3 Review of Existing Schemes

The scheme of the steganography and authentication based on secret image sharing was first proposed by Lin and Tsai [12] in 2004 as follows. Firstly, select n meaningful images as the cover images and divide them into non-overlapping 4-pixel blocks. Secondly, use each pixel of the secret image as sharing coefficient a_0 and choose $(k-1)$ random numbers as sharing coefficients $a_1, a_2, \ldots, a_{k-1}$ in Equation (2) to generate n shadows. Note that there is modulo 251 in this scheme instead of modulo 2^t and the top-left pixel value x of each cover image block in Figure 1 is used as the value in Equation (2) for computing all of the shared pixels $f(x)$. Thirdly, convert each shared pixel into a binary representation $s_1, s_2, \ldots, s_8$, and meanwhile, one parity bit b is generated as an authentication bit by the secret key K. Finally, embed the nine bits $(s_1, s_2, \ldots, s_8, b)$ into the cover image block as in Figure 2 to form the stego-block. However, the calculation of modulo 251 damages the quality of the reconstructed secret image, and the authentication ability of 1 bit is very weak.

Later, Yang et al.'s scheme [20] overcame the weakness of Lin and Tsai's scheme. The main improvements were as follows: (1) The modulus value is set to 2^8, so there is no loss of data when the secret image is reconstructed; and (2) One authentication bit b, which is calculated by a hash function, offers higher authentication ability than that of Lin and Tsai's scheme. The embedding strategy of this scheme is shown in Figure 3. However, because the bits x_7, x_8 of pixel X need to be recorded in the pixel W as shown in Figure 3, the visual quality of the stego images may be greatly reduced.

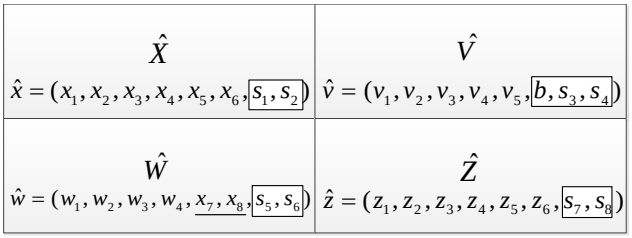


Figure 3: Block structure of the stego image in Yang et al.'s scheme

Chang et al.'s scheme [2] has better stego image quality and authentication ability than previous schemes. This scheme uses k consecutive pixels of the secret image as sharing coefficients $a_0, a_1, \ldots, a_{k-1}$ in Equation (2) to generate smaller shadows, so the visual quality of the stego image is clearly improved. Meanwhile, the use of four authentication bits (b_1, b_2, b_3, b_4) in each stego-block, which is calculated by CRT, also decrease the possibility of successful tampering. The embedding strategy of this scheme is shown in Figure 4.

Different from the above mentioned three schemes, Es-

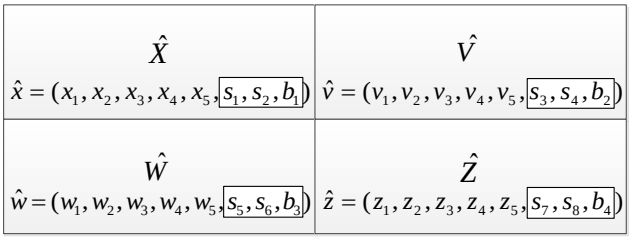


Figure 4: Block structure of the stego image in Chang et al.'s scheme

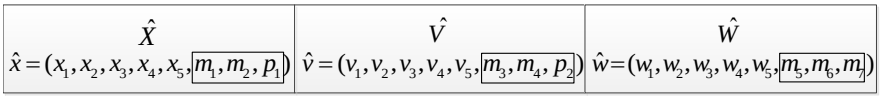


Figure 5: Block structure of the stego image in the proposed scheme

lami et al. [6] employ the concept of linear cellular automata instead of Shamir's (k,n) threshold image sharing scheme. Although this scheme also achieves good visual quality and authentication ability, it needs to offer at least k consecutive shadows to reveal the secret image, and cannot accurately locate the tampered blocks.

3 Proposed Scheme

The proposed scheme includes the following two phases: (1) the sharing and embedding phase; and (2) the authentication and revealing phase.

3.1 Sharing and Embedding Phase

Based on the details described in previous sections, we can now describe a complete algorithm to implement the proposed sharing and embedding procedure. Assume that the secret image SE is a $m \times m$ gray-scale image. Select n meaningful images like photographs of famous people or beautiful landscapes as cover images with a size of $w \times w$.

Wang and Su's scheme [18] as described in Sub-section 2.1 was first adopted to generate n shadows. Notice that the sharing function in our scheme is slightly different from that in Wang and Su's scheme. In order to meet the needs of the matrix encoding, we use the parameter $t = 3$ instead of $t = 8$ in Equation (2); that is, every 3 bits that are obtained from the Huffman output stream are converted into the decimal sharing coefficient, and the generation polynomial used in GF(2^3) is $x^3 + x + 1$, as shown in Equation (7). Apparently, all pixel values in each shadow are limited to the range from 0 to 7.

$$f_l(x) = (a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}) \mod 2^3. \quad (7)$$

To achieve a high embedding efficiency which indicates the average bit number of embedded secret data per change, the ordered triple (1,7,3) is used in our scheme as an example of matrix encoding. Accordingly, the structure of the four-pixel block in the cover image of previous schemes also needs to be changed. In the proposed scheme, as shown in Figure 5, every three pixels form a block so as to implement the matrix encoding. Suppose that we choose $m_1, m_2, m_3, m_4, m_5, m_6, m_7$ from the three pixels $\hat{X}, \hat{V}, \hat{W}$ in each block as an embedding cell. And each shadow pixel is converted into binary bits s_1, s_2, s_3 as secret data to be embedded. The following is a basic algorithm that reveals the principle of the proposed scheme for embedding.

Embedding algorithm.

Input. n shadow images $S^{(j)}$, n cover images $C^{(j)}, j \in [1,n]$, and a secret key K.

Output. n stego images $\hat{C}^{(j)}, j \in [1,n]$, and n sub-keys $K_1, \ldots, K_j, \ldots, K_n$.

Steps.

Step 1. Split the secret key K into n sub-keys $K_1, \ldots, K_j, \ldots, K_n$ by using the polynomial sharing function (Note that here we do not need the modulo operator).

Step 2. Divide each cover image $C^{(j)}$ into non-overlapping blocks with size 1×3.

Step 3. Select random location for the embedded block through the following steps:

- Assume that the length of the j-th shadow is l, and the number of all blocks in the j-th cover image is L, then calculate

$$t = \lfloor \frac{L}{l} \rfloor, \quad (8)$$

where t means that at least one block during t continuous blocks can be used to embed a shadow pixel.

- Use the secret sub-key K_j to generate the random sequence $\{r_1^{(j)}, r_2^{(j)}, \ldots, r_l^{(j)}\}$.
- The $i^{(j)}$-th location of the embedded block is calculated by Equation (9).

$$Loc(i^{(j)}) = i^{(j)} \times t + r_i^{(j)} \bmod t, \quad 1 \le i^{(j)} \le l. \quad (9)$$

Here, we randomly select one block to embed secret data during t continuous blocks, which can realize uniform random embedding.

Step 4. Embed binary bits s_1, s_2, s_3, which are converted from the $i^{(j)}$-th shadow pixel, into the $i^{(j)}$-th embedded block by using the matrix encoding. Here, at most one bit was changed in the $i^{(j)}$-th embedded block.

Step 5. Calculate two authentication bits p_1, p_2 and embed them into the relevant position in Figure 5.

$$p_1 p_2 = XOR\langle H_{K_j}(TH^{(j)} || B_{ID}^{(j)}) \rangle_2, \quad (10)$$

where $H_{K_j}(\cdot)$ is a hash function with the secret key K_j, $TH^{(j)}$ is 15-bit exclusive

the check bits p_1, p_2 and an embedding cell $m_1, m_2, m_3, m_4, m_5, m_6, m_7$ from each block of j-th cover image, $||$ represents the concatenation operator for strings, $B_{ID}^{(j)}$ is the block ID of j-th cover image. Then, the generated output of hash function is executed XOR operation to obtain two authentication bits p_1, p_2.

From the description above, we can observe three facts.

1) Secret data was evenly and randomly distributed throughout the whole cover image in order to reduce changes in the visual and statistical properties of the stego images.

2) No more than one bit is changed when embedding the secret data. In other words, no more than three bits were changed in each block of the cover images, so higher visual quality of the stego images can be acquired.

3) Sub-keys K_j used to calculate authentication bits can prevent the other participants from using their own authentication bits to counterfeit.

3.2 Authentication and Revealing Phase

Our proposed scheme has two-stage authentication ability.

1) Identity authentication, that is, any k or more participants should provide their sub-keys K_j to the administrator. Afterwards, these sub-keys K_j can be used to recover the secret key K by Lagrange's interpolation. If the recovered secret key K is different from the original secret key K, then identity authentication failed; otherwise, further tamper authentication will be implemented.

2) Tamper authentication. Stego images which pass the identity authentication are divided into a set of blocks with three pixels, and then the corresponding sub-keys K_j and the information of block are used to generate the authentication bits. Authentication bits are taken out from the provided stego images, and then compared with the generated authentication bits. If they differ, then the stego image has been tampered with; otherwise, authentication is successful.

After successful authentication, the secret data s can be retrieved by using the inverse processing of data embedding. The basic steps for data extraction are as following:

Step 1. Collect any k stego images which have been certified.

Step 2. Divide each stego image into non-overlapping blocks with size 1×3;

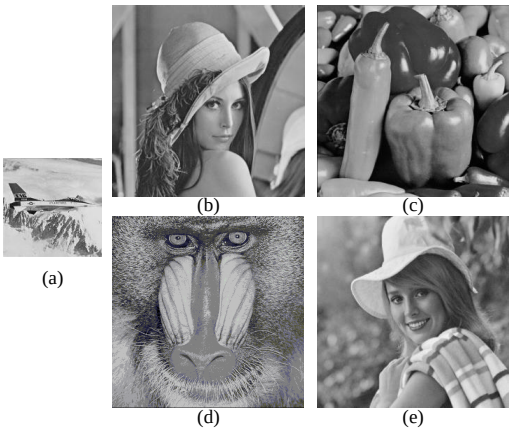


Figure 6: The five test images, (a) Jet-F16, (b) Lena, (c) Peppers, (d) Baboon, (e) Elaine

Step 3. Confirm location of the embedded block using Step 3 in the embedding process.

Step 4. Extract an embedding cell $m_1, m_2, m_3, m_4, m_5, m_6, m_7$ from every confirmed embedded block. And Use the matrix encoding described in Section 2.2 to retrieve the binary bits s_1, s_2, s_3.

Step 5. Convert each binary bits s_1, s_2, s_3 into the decimal number, and these decimal numbers are rearranged to recover the shadow image.

When all k shadows corresponding to all k stego images are obtained, the secret image can be recovered by using the method as described in Section 2.1. Note that we use Shamir's (k, n)-threshold scheme to generate the shadows, so less than k stego images is not enough to recover any information about the secret image.

4 Experimental Results and Analysis

Five test images from the USC-SIPI image database as shown in Figure 6 contain a 256×256 secret image Jet-F16 (Figure 6(a)) and four 512×512 cover images Lena (Figure 6(b)), Peppers (Figure 6(c)), Baboon (Figure 6(d)) and Elaine (Figure 6(e)). Three groups of experiments are performed in this section: (1) to estimate the visual quality of the stego images; and (2) to test the authentication ability.

4.1 Estimate the Visual Quality of the Stego Images

The objective quality of the stego images is measured by the peak-signal-to-noise ratio (PSNR), which is defined as

$$PSNR = 10\log_{10}(\frac{255^2}{\text{MSE}})\text{dB}, \quad (11)$$

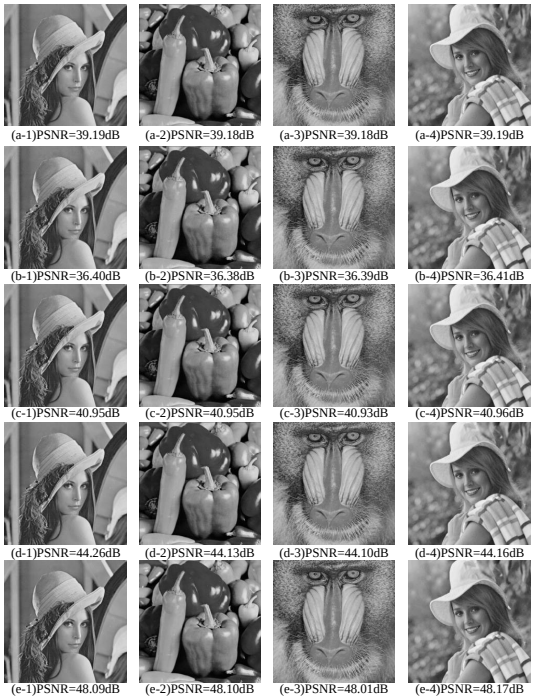


Figure 7: The experimental results of different schemes when (a) stego images in Lin and Tsai 's scheme, (b) stego images in Yang et al.'s scheme, (c) stego images in Chang et al.'s scheme, (d) stego images in Eslami et al.'s scheme, (e) stego images in the proposed scheme

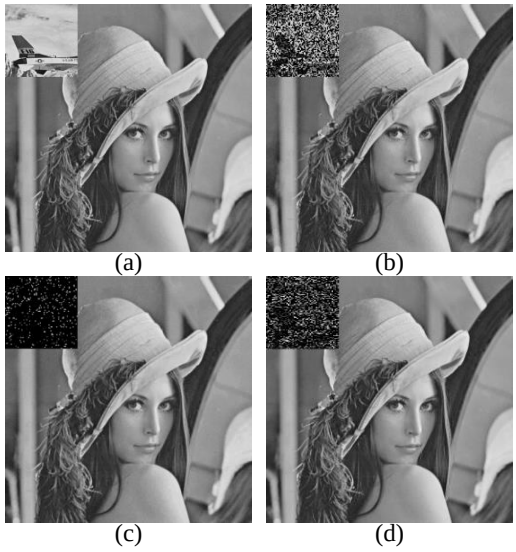


Figure 8: Experimental results for three different schemes, (a) tampered Lena stego image, (b) Yang et al.'s scheme with $DR \approx 1/2$, (c) Chang et al.'s scheme with $DR \approx 15/16$, (d) our scheme with $DR \approx 3/4$

where MSE is the mean square error between the cover images and the stego images. In this experiment, we choose two groups of different threshold parameters: the first group is $k = 2, n = 3$; the second is $k = 3, n = 4$. Figure 7 demonstrates the experimental results of different schemes when $k = 2, n = 3$. As can be seen from Figure 7, the proposed scheme obtains the highest PSNR value among the compared schemes. Note that all of the experiments are conducted under the same conditions, i.e., the same secret image, the same cover image and the same k, n.

Table 1 summarizes the experimental results of different schemes when $k = 3, n = 4$. Elaine is used as the fourth cover image in all mentioned schemes. Under the same experimental conditions, the proposed scheme still obtains the highest PSNR value. The reason for this objective quality is that our scheme changes relatively fewer bits, that is to say, not more than one bit in each pixel of the embedding block is modified by hidden secret and authentication bits.

In order to further illustrate the effectiveness of the proposed scheme, we calculated average number of modified bits per pixel in different schemes when different k because fewer modifications can obtain higher quality of the stego images. Table 2 summarizes the results of this comparison. It can be seen from the comparison that, the average number of modified bits per pixel in our scheme is smallest when different k, so we also obtain the best visual quality of stego images.

4.2 Testing the Authentication Ability

The detection ratio is defined as $DR = NTPD/NTP$ to test the authentication ability, where $NTPD$ is the number of tampered pixels that are detected, and NTP is the number of tampered pixels [6].

In this sub-section, we make the tampered Lena stego image by adding a shrunk version of the Jet-F16 in Figure 8(a). Herein, Lin and Tsai's scheme and Eslami et al.'s scheme are not considered to compare the authentication ability because they cannot effectively prevent tampering or locate the tampered position. Figures 8(b), (c) and (d) show the experimental results for three different schemes. Figures 8 (b) and (c) use the same size of stego-block (4 pixels) to achieve tamper detection, but the detection ratio of Yang et al.'s scheme is lower than that of Chang et al. In our scheme, we choose the smaller stego-block (3 pixels) to achieve tamper detection.

Although the detection ratio is slightly lower than that of Chang et al.'s scheme, we adopt a two-stage authentication mechanism, that is to say, the participant firstly must provide the right sub-key K_j to pass identity authentication before tamper detection. It is very difficult to accurately guess the random sub-key K_j. Therefore, our scheme has higher authentication ability.

Table 1: Experimental results of different schemes when $k = 3, n = 4$

Scheme	PSNR			
	Lena	Pepper	Baboon	Elaine
Lin and Tsai's scheme	39.22	39.18	39.21	39.22
Yang et al.'s scheme	36.41	36.38	36.35	36.40
Chang et al.'s scheme	42.70	42.77	42.73	42.69
Eslami et al.'s scheme	47.12	47.16	47.10	47.15
Our scheme	49.47	49.47	49.50	49.51

Table 2: Comparison of average number of modified bits per pixel with different k

Different k	average number of modified bits per pixel				
	Lin et al.	Yang et al.	Chang et al.	Eslami et al.	Ours
k	–	–	$1 + (2/k)$	$2/(k-1)$	–
2	2.25	2.75	2	2	0.85
3	2.25	2.75	1.67	1	0.74
4	2.25	2.75	1.5	0.67	0.59

5 Conclusions

In this paper, a secret image sharing with deep-steganography and two-stage authentication was proposed. This scheme is based on matrix encoding to embed secret shadows into cover images and at most one bit was changed in the embedded block, so secret data does not directly appear in the pixels of the cover image. This ensures the security of secret data and obtains higher visual quality of the stego image. As far as authentication ability is concerned, our scheme increases the process of the identity authentication, that is to say, participants must first provide the right sub-key to pass identity authentication, and then pass tamper authentication. The final experimental results also demonstrate that our scheme achieves better authentication ability than those referenced in the literature review.

Acknowledgments

This work has been supported in part by National Natural Science Foundation of China (No. 61272262), Program for New Century Excellent Talent in Universities (NCET-12-1037), International Cooperative Program of Shanxi Province (No. 2015081015).

References

[1] S. Chakraborty and S. K. Bandyopadhyay, "Steganography method based on data embedding by sudoku solution matrix," *International Journal of Engineering Science Invention*, vol. 2, no. 7, pp. 38–42, 2013.

[2] C. C. Chang, Y. P. Hsieh, and C. H. Lin, "Sharing secrets in stego images with authentication," *Pattern Recognition*, vol. 41, no. 10, pp. 3130–3137, 2008.

[3] C. C Chang, N. T. Huynh, and T. F. Chung, "Efficient searching strategy for secret image sharing with meaningful shadows," *International Journal of Machine Learning and Computing*, vol. 3, no. 5, pp. 342–352, 2014.

[4] R. Crandall, "Some notes on steganography," *Posted on Steganography Mailing List*, pp. 1–6, 1998.

[5] X. Dong, "A multi-secret sharing scheme based on the CRT and RSA," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 47–51, 2015.

[6] Z. Eslami, S. H. Razzaghi, and J. Z. Ahmadabadi, "Secret image sharing based on cellular automata and steganography," *Pattern Recognition*, vol. 43, no. 1, pp. 397–404, 2010.

[7] L. Fan, T. Gao, Q. Yang, and Y. Cao, "An extended matrix encoding algorithm for steganography of high embedding efficiency," *Computers & Electrical Engineering,*, vol. 37, no. 6, pp. 973–981, 2011.

[8] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of lsb steganography in color and grayscale images," *Proceedings of The International Conference on Multimedia and Security: New Challenges*, pp. 27–30, 2001.

[9] H. Gupta, A. P. R. Kumar, and S. Changlani, "Steganography using lsb bit substitution for data hiding," *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 2, no. 10, pp. 676–680, 2013.

[10] C. Jianping and U. Jauhari, "Analysis on regular-singular steganalysis algorithm for multimedia,"

Journal of Network & Information Security, vol. 4, no. 4, pp. 395–403, 2013.

[11] L. Li, A. El-Latif, X. Yan, S. Wang, and X. Niu, "A lossless secret image sharing scheme based on steganography," in *Proceedings of The International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC'12)*, pp. 1247–1250, Beijing, China, Dec. 2012.

[12] C. C. Lin and W. H. Tsai, "Secret image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 73, no. 3, pp. 405–414, 2004.

[13] M. Naor and A. Shamir, "Visual cryptography," *Advances in Cryptology (EUROCRYPT'94)*, pp. 1–12, Springer, 1995.

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

[15] M. Shobana, "Efficient X-box mapping in stego-image using four-bit concatenation," *International Journal of Electronics and Information Engineering*, vol. 1, no. 1, pp. 29–33, 2014.

[16] A. Singh and U. Jauhari, "A symmetric steganography with secret sharing and psnr analysis for image steganography," *International Journal of Scientific & Engineering Research*, vol. 3, no. 6, pp. 2–8, 2012.

[17] C. C. Thien and J. C. Lin, "Secret image sharing," *Computers & Graphics*, vol. 26, no. 1, pp. 765–770, 2002.

[18] R. Z. Wang and C. H. Su, "Secret image sharing with smaller shadow images," *Pattern Recognition Letters*, vol. 27, no. 6, pp. 551–555, 2006.

[19] D. C. Wu and W. H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9, pp. 1613–1626, 2003.

[20] C. N. Yang, T. S. Chen, K. H. Yu, and C. C. Wang, "Improvements of image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 80, no. 7, pp. 1070–1076, 2007.

[21] C. N. Yang, J. F. Ouyang, and L. Harn, "Steganography and authentication in image sharing without parity bits," *Optics Communications*, vol. 285, no. 7, pp. 1725–1735, 2012.

Li Liu received her B.E. degree in communication engineering in 2002, from Lanzhou Railway University and M.E. degree in communication and information system in 2006, from Lanzhou Jiaotong University. Now, she is a Ph. D student in Northwestern Polytechnical University. Her current research interests include information hiding and secret sharing. She has published more than 10 papers.

Anhong Wang was born in Shanxi Province in 1972. She received B.E and M.E. degrees from Taiyuan University of Science and Technology (TYUST) respectively in 1994 and 2002, and PHD degree in Institute of Information Science, Beijing Jiaotong University in 2009. She became an associate professor with TYUST in 2005 and became a professor in 2009. She is now the director of Institute of Digital Media and Communication, Taiyuan University of Science and Technology. Her research interest includes image/video coding, compressed sensing, and secret image sharing. She has published more than 40 papers. Now she is leading two national research projects from National Science Foundation of China.

Chin-Chen Chang received his B.E. degree in applied mathematics in 1977 and the M.E. degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph. D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. Since February 2005, he has been a Chair Professor of Feng Chia University. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression and data structures.

Zhihong Li was born in Shanxi Province in 1970. He is currently an associate professor in Taiyuan University of Science and Technology (TYUST). He received the B.Eng. degrees in electronic information engineering from Taiyuan University of Science and Technology (TYUST) in 1994. His research interest includes compressed sensing and secret image sharing. He has participated in the projects on distributed video coding and now is leading one research project on image secret from Shanxi Natural Science Foundation.

Behavioral and Security Study of the OHFGC Hash Function

Ahmed Drissi and Ahmed Asimi
(Corresponding author: Ahmed Drissi)

Department of Mathematics, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco
B.P 8106, Agadir, Morocco
(Email: idrissi2006@yahoo.fr)

(Received Mar. 10, 2016; revised and accepted May 22 & June 10, 2016)

Abstract

The designs of several hash functions (SB, FSB, RFSB, SFSB, OHFGC, $\cdots$) are based on the error-correcting codes properties. The hash function based on the classical Goppa code "OHFGC" [7] is distinguished by the possibility that an user selects certain parameters to achieve a level of performance and security corresponds to its needs. The objective of this article examines the security features of the hash function "OHFGC" and its behavior in order to propose relevant parameters for different user situations. We also propose both a method to summarize all parameters in one and a method that links the size of the hashed to the document.

Keywords: Classical Goppa code, one way hash function, syndrome decoding

1 Introduction

Several hash functions (SB, FSB, RFSB, SFSB, OHFGC, $\cdots$) [2, 3, 4, 9] are based on the error-correcting codes properties. The hash function based on the classical Goppa code "OHFGC" [7] is distinguished by the ability that an user selects certain parameters to achieve a level of performance and security corresponds to its needs. In the next section, we recall the algorithms components of the OHFGC. Section Three is devoted to the security study by the design model and the hashed size. In section four, we study the performance, the behavior of the OHFGC and its sensitivity to initial conditions. Our proposed method of choosing a single parameter from the others is presented in section five. It ends with a conclusion. Table 1 is the notations used in this paper.

Let the finite field $F_{2^m} = \{0, 1, \alpha, \alpha^2, \cdots, \alpha^{2^m-2}\}$ and its primitive element α which is the root of a primitive polynomial of the degree m on F_2 [10]. There is a biunivocal correspondence between the elements of F_{2^m}, as a F_2 vector space its base is $(1, \alpha, \alpha^2, \cdots, \alpha^{m-1})$, and the elements of F_{2^m} is defined by:

$$\begin{array}{rcl} \varphi : F_{2^m} & \longrightarrow & F_{2^m} \\ x = \sum_{i=0}^{i=m-1} a_i \alpha^i & \longrightarrow & (a_0, \cdots, a_{m-1})^T \end{array}$$

2 Recall on the OHFGC Hash Function

The hash of a message M by $OHFGC$ is according to the MERKLE and DAMGARAD model [6, 11], in the heart of this model there is a compression function. The compression function of the $OHFGC$ [7] is composed of the following algorithms:

- A compression function CF.

  A compression function CF, of the input size n and of the output r, based on H (a parity check matrix of a classical Goppa code), and is defined as follows:

$$\begin{array}{rcl} CF : F_{2^n} & \rightarrow & F_{2^r} \\ x & \rightarrow & x^{(1)} + H\phi(x)^t, \end{array}$$

  with $x = (x^{(2)}, x^{(1)})$, $x^{(1)} \in F_{2^{n-r}}$, $x^{(2)} \in F_{2^r}$ and

$$\begin{cases} \phi(x) = x & \text{if } w(x) \leq \frac{n}{2} \\ \phi(x) = x \oplus 1^n & \text{if } w(x) \succ \frac{n}{2} \end{cases}$$

- The generation of a parity check matrix.

  The generation of a parity check matrix ($H \in M_{r,n}(F_2)$) from a primitive element of a field F_{2^m} and an integer n with $(2m \prec n \prec 2^m - 1)$.

The generation of H is done as follows:

1) Choose an integer n such as $2m \prec n \prec 2^m - 1$ and a primitive element α of F_{2^m}.

2) Calculate $(i_j)_{j=1}^n$ with $i_j = n^j \bmod (2^m - 1)$ and $t = E(\frac{n}{2m})$.

Table 1: Notations

N	The set of integers.
$F_2 = \{0, 1\}$	A finite field of the two elements.
n	an integer.
m	an integer.
F_{2^m}	The finite field of 2^m elements, with m an integer.
${F_{2^m}}^*$	The multiplicative group of the nonzero elements of F_{2^m}.
$M_{rxn}(K)$	The set of rxn matrices with coefficients in an abelian field K.
${F_2}^n$	The set of the vectors that components 0 or 1 and their length is n.
$1^n = (1, 1, ..., 1)$	The vector of n components equal to 1.
$OHFGC$	One-way Hash function synchronized based on Goppa Codes.
CF	Compression function.
$E(x)$	The integer part of x.
t	An integer.
$\Gamma(L, x^t)$	A classical Goppa code with L its support and x^t its polynomial.
$OHFGC(m)$	One-way Hash function based on Goppa Code with his principle parameter m.
$w(x)$	The sum of the components of x.
$\oplus$	An XOR operation.

3) Calculate $K' = (\varphi(\alpha_j^{i-t-1}))_{i=1,\cdots,t;j=1,\cdots,n}$.

4) The parity control matrix H is composed of lines in K' without repetition and in the same order. This is the parity check matrix of $\Gamma(L, x^t)$ in F_2 of rxn type.

5) r is the output size of $OHFGC$ and the compression function CF.

Remark 1. *We cannot predict the value of the hashed size r before the construction of H, this is due to a particular property of the parity check matrix of a classical Goppa code. We have to recourse to implementation.*

3 The OHFGC Security Study

The security of the entire hash functions depends mainly on its design model and the hashed size. The first ensures resistance against structural attacks and the second guarantees its resistance to generic attacks. In the two following paragraphs we discuss these principles in the case of the OHFGC.

1) The OHFGC security based on design model.

   The OHFGC is built according to the model MERKLE and DAMAGARAD [6, 11]. MERKLE [6] showed that the security of any hash function is designed according to the model is summarized in compression function of the resistance, constructed with, at the three security criteria (resistance to pre-image, second pre-image and collisions).

   For hash functions were based on code, including the OHFGC, the security is easily linked to the difficulty of the problem by decoding syndromes [4, 8, 12].

   The following two issues proved hard [7], provide the security for the OHFGC.

   Given H a matrix of the type rxn of elements of the F_2 and $s \in F_{2^r}$.

   Find $x = (x^{(2)}, x^{(1)}) \in F_{2^{n-r}} x F_{2^r}$ such as $x^{(1)} + Hx^t = s$.

   Given H a matrix of the type of rxn of elements of the F_2 and $s \in F_{2^r}$.

   Find $x = (x^{(2)}, x^{(1)}) \in F_{2^{n-r}} x F_{2^r}$ and $y = (y^{(2)}, y^{(1)}) \in F_{2^{n-r}} x F_{2^r}$ such as $x^{(1)} + y^{(1)} = H(x + y)^t$.

2) The OHFGC security based on its hashed size.

   Generic attacks [5] (see Table 2) depend on the number of the possible hashed 2^r of the size r. As to ensure safe of some functions hash, simply increase the size of hashed (at the moment the sizes 256 and 512 are considered acceptable). For the OHFGC, we propose to give varying sizes included in intervals depending on its primary endpoint: primitive polynomial. In addition, it is distinguished by the possibility of extending these intervals by increasing the degree of the primitive polynomial. This property gives the complexity of the OHFGC for a longer time.

Table 2: Complexity of the best generic attacks

Generic attack	Complexity
Search pre-image	2^r
Research of second pre-image	2^r
Research of collisions	$2^{\frac{r}{2}}$

4 The Behavior Study and the Performance of OHFGC Function

The parameters of the OHFGC function are n, m, α (α is a root of a primitive polynomial $p(x)$ of degree m) [1] and hashed size r. Tables 3, 4, 5, 6, 7, provides examples of the parameters that can be used. These examples give us an idea of the possible choices.

Table 3: The hashed size for m=8

m	$p(x)$	n	Hashed size r	2^m
8	$x^8+x^5+x^3+x+1$	254	4	256
8	$x^8+x^5+x^3+x+1$	253	90	256
8	$x^8+x^5+x^3+x+1$	252	120	256
8	$x^8+x^5+x^3+x+1$	251	16	256
8	$x^8+x^5+x^3+x+1$	250	120	256
8	$x^8+x^5+x^3+x+1$	249	120	256
8	$x^8+x^5+x^3+x+1$	248	120	256
8	$x^8+x^5+x^3+x+1$	247	90	256
8	$x^8+x^5+x^3+x+1$	100	43	256

Table 4: The hashed size for m=9

m	$p(x)$	n	Hashed size r	2^m
9	x^9+x^5+1	254	4	512
9	x^9+x^5+1	253	90	512
9	x^9+x^5+1	510	4	512
9	x^9+x^5+1	509	251	512
9	x^9+x^5+1	508	246	512
9	x^9+x^5+1	507	251	512
9	x^9+x^5+1	506	252	512
9	x^9+x^5+1	504	252	512
9	x^9+x^5+1	503	59	512
9	x^9+x^5+1	502	60	512
9	x^9+x^5+1	501	243	512
9	x^9+x^5+1	500	243	512
9	x^9+x^5+1	400	198	512
9	x^9+x^5+1	300	52	512
9	x^9+x^5+1	200	99	512

These data lead us to seek to have a OHFGC function of the variable hashed size and summarize the parameters in one.

1) The behavior study of the OHFGC.

   Any modification of the hashed document leads a variation on the hashed. The variation on the hashed is measured by the Hamming distance between the two vectors (hashed). The graphs (Figures 1, 2, 3, 4) represent the Hamming distance between the

Table 5: The hashed size for m=10

m	$p(x)$	n	Hashed size r	2^m
10	$x^{10}+x^3+1$	1022	495	1024
10	$x^{10}+x^3+1$	1021	373	1024
10	$x^{10}+x^3+1$	1020	510	1024
10	$x^{10}+x^3+1$	1019	364	1024
10	$x^{10}+x^3+1$	1018	500	1024
10	$x^{10}+x^3+1$	1017	500	1024
10	$x^{10}+x^3+1$	1016	500	1024
10	$x^{10}+x^3+1$	1015	365	1024
10	$x^{10}+x^3+1$	1014	500	1024
10	$x^{10}+x^3+1$	1013	500	1024
10	$x^{10}+x^3+1$	1012	493	1024
10	$x^{10}+x^3+1$	1011	500	1024
10	$x^{10}+x^3+1$	1000	387	1024
10	$x^{10}+x^3+1$	100	50	1024
10	$x^{10}+x^3+1$	800	400	1024

Table 6: The hashed size for m=11

m	$p(x)$	n	Hashed size r	2^m
11	$x^{11}+x^2+1$	2046	4	2048
11	$x^{11}+x^2+1$	2045	1011	2048
11	$x^{11}+x^2+1$	2044	1012	2048
11	$x^{11}+x^2+1$	2043	1011	2048
11	$x^{11}+x^2+1$	2042	1012	2048
11	$x^{11}+x^2+1$	2041	1012	2048
11	$x^{11}+x^2+1$	2040	1012	2048
11	$x^{11}+x^2+1$	2039	1011	2048
11	$x^{11}+x^2+1$	2038	1012	2048
11	$x^{11}+x^2+1$	2037	1012	2048
11	$x^{11}+x^2+1$	2000	990	2048

Table 7: The hashed size for m=12

m	$p(x)$	n	Hashed size r	2^m
12	$x^{12}+x^6+x^4+x+1$	4094	4	4096
12	$x^{12}+x^6+x^4+x+1$	4093	1500	4096
12	$x^{12}+x^6+x^4+x+1$	4092	1783	4096
12	$x^{12}+x^6+x^4+x+1$	4091	64	4096
12	$x^{12}+x^6+x^4+x+1$	4090	1602	4096
12	$x^{12}+x^6+x^4+x+1$	4089	1767	4096
12	$x^{12}+x^6+x^4+x+1$	4088	1587	4096
12	$x^{12}+x^6+x^4+x+1$	4087	16	4096
12	$x^{12}+x^6+x^4+x+1$	4000	64	4096
12	$x^{12}+x^6+x^4+x+1$	3000	128	4096
12	$x^{12}+x^6+x^4+x+1$	409	194	4096

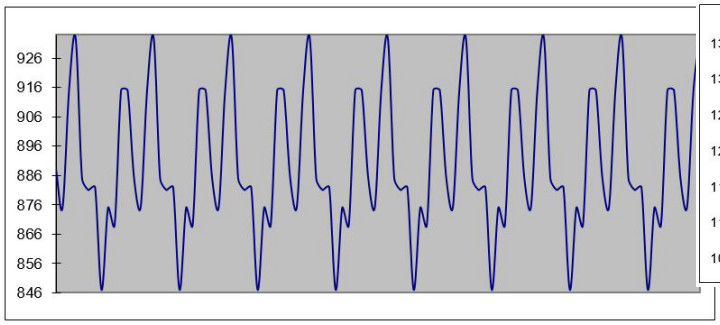


Figure 1: OHFGC (4092, 12,1783)

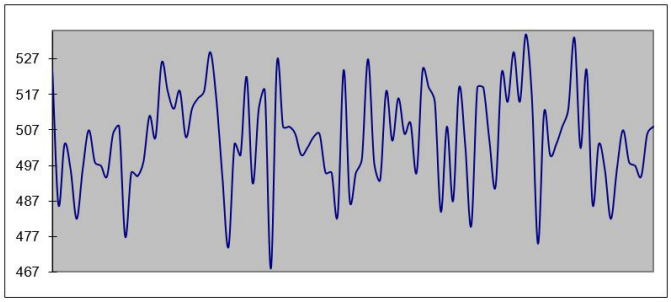


Figure 2: OHFGC(2040,11,1012)

hashed of the original document and the hashed of the amended document by a single bit within the first 100 positions in the original document.

In summary, in the four examples of the OHFGC function, each modification of the document to hash, by a single bit, causes variation of the hashed by approximately half the number of bits.

2) The OHFGC performance.

We hashed a file of size 1.01 MB by the OHFGC(n, m, r) function, Table 8 shows the execution time for the chosen parameters and which have its performance.

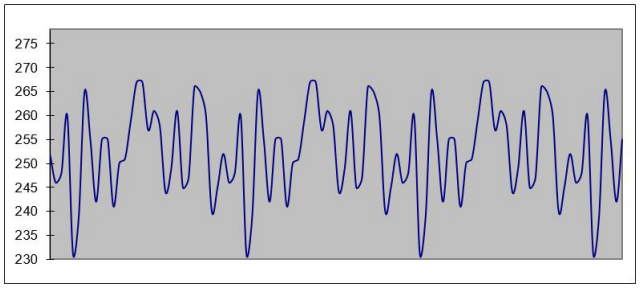


Figure 3: OHFGC (1020, 10,510)

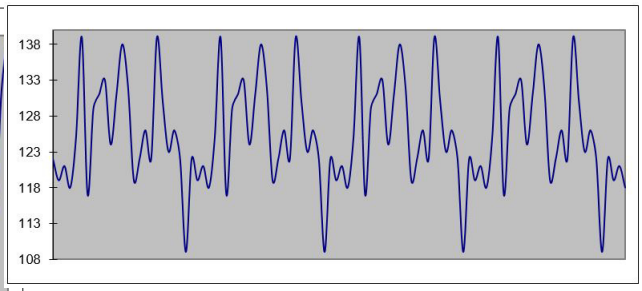


Figure 4: OHFGC (504, 9,252)

Table 8: Performance of the on core (TM) 2 duo CPU 2.00 GHZ

functions	Execution time
OHFGC (4092, 12,1783)	10,98200 s
OHFGC (2040, 11,1012)	6,70800 s
OHFGC (1020, 10,510)	3,52600 s
OHFGC (504, 9,252)	1,95000 s

5 Proposal Method for Selecting Parameters

After the behavioral study of the $OHFGC(n, m, r)$, we propose to keep a single parameter of the $OHFGC(m)$ and to link n to the document size to be hashed by the relation $n = (2m+1+document\ size) mod(2^m-2)$ and by following the hashed size r will vary from one document to another in the interval $[1,mE(\frac{2^m-2}{2m})]$.

Explication 1. *The hashed size is between 1 and* $[1,mE(\frac{2^m-2}{2m})]$ *indeed. The matrix H has at least one line. we have* $n = (2m+1+document\ size) mod(2^m-2)$ *then* $2m \prec n \prec 2^m-2$. *We have also* $r \leq mt$ *(since r is the number of lines in H after reduction) consequently* $1 \leq r \leq mE(\frac{2^m-2}{2m})$.

Remark 2. *Having the variable hashed size in a range increase the complexity of generic attacks. We take for example the following intervals (Table 9).*

Table 9: Examples of the intervals document size

m	$[1,mE(\frac{2^m-2}{2m})]$
8	[1,120]
9	[1,252]
10	[1,510]
11	[1,1023]
12	[1,2040]

6 Conclusion

In conclusion, we can announce that our OHFGC(m) function parameterized by a primitive polynomial of the degree m and of the variable size from one document to another, is an efficient and secure function. The flexibility of choosing the parameter m of the OHFGC depending on the context of the use ensures that our exclusive function can last longer as it will be used by different users in different contexts.

References

[1] A. Asimi and A. Lbekkouri, "Determination of irreducible and primitive polynomials over a binary finite field," 2009. (`file:///C:/Users/user/Downloads/asimiprim.pdf`)

[2] D. Augot, M. Finiasz, P. Gaborit, S. Manuel, and N. Sendrier, "SHA-3 proposal: FSB," Submission to NIST, pp. 81–85, 2008.

[3] D. Augot, M. Finiasz, and N. Sendrier, "A family of fast syndrome based cryptographic hash functions," in *Progress in Cryptology (Mycrypt'05)*, pp. 64–83, Springer, 2005.

[4] D. J. Bernstein, T. Lange, C. Peters, and P. Schwabe, "Really fast syndrome-based hashing," in *Progress in Cryptology (AFRICACRYPT'11)*, pp. 134–152, Springer, 2011.

[5] C. Boura, *Analyse De Fonctions De Hachage Cryptographiques*, Ph.D. Thesis, University Pierre et Marie Curie-Paris VI, 2012.

[6] I. B. Damgard, "A design principle for hash functions," in *Advances in Cryptology (CRYPTO'89)*, pp. 416–427, Springer, 1989.

[7] A. Drissi and A. Asimi, "One-way hash function based on goppa codes ohfgc," *Applied Mathematical Sciences*, vol. 7, no. 143, pp. 7097–7104, 2013.

[8] M. Finiasz, "Nouvelles constructions utilisant des codes correcteurs derreurs en cryptographie á clef publique," These de doctorat, *É*cole Polytechnique, 2004.

[9] W. R. Ghanem, M. Shokir, and M. Dessoky, "Defense Against Selfish PUEA in Cognitive Radio Networks Based on Hash Message Authentication Code," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 12–21, 2016.

[10] R. Lidl and H. Niederreiter, *Finite Fields (Encyclopedia of Mathematics and Its Applications, vol. 20)*, Reading, MA, USA: AddisonWesley, pp. 428–431, 1983.

[11] R. C. Merkle, "One way hash functions and des," in *Advances in Cryptology (CRYPTO'89)*, pp. 428–446, Springer, 1989.

[12] N. Sendrier, *Cryptosyst Emes a Cl e Publique Bas es Sur Les Codes Correcteurs D'erreurs*, Habilitation diriger les recherches, Universit Pierre et Marie Curie, Paris, France (in French), 2002.

Ahmed Drissi received his PhD degree in cryptology from the Faculty of Science, the University Ibn Zohr Agadir, Morocco in 2014. His research interests include Code theory and the Cryptology.

Ahmed Asimi received his PhD degree in Number theory from the University Mohammed V Agdal in 2001. He is reviewer at the International Journal of Network Security (IJNS). His research interest includes Number theory, Code theory, and Computer Cryptology and Security. He is a full professor at the Faculty of Science at Agadir since 2008.

Message Recovery via an Efficient Multi-Proxy Signature With Self-certified Keys

Manoj Kumar Chande[1], Cheng-Chi Lee[2,3], Chun-Ta Li[4]
(Corresponding author: Cheng-Chi Lee)

School of Studies in Mathematics, Pt. Ravishankar Shukla University[1]
Raipur, 492010, Chhattisgarh, India
E-mail: manojkumarchande@gmail.com
Department of Library and Information Science, Fu Jen Catholic University[2]
510 Jhongjheng Road, Taipei 24205, Taiwan, R.O.C.
E-mail: cclee@blue.lins.fju.edu.tw
Department of Photonics and Communication Engineering, Asia University[3]
Wufeng Shiang, Taichung, Taiwan 413, R.O.C.
Department of Information Management, Tainan University of Technology[4]
No. 529, Zhongzheng Road, Tainan City 71002, Taiwan, R.O.C.
E-mail: th0040@mail.tut.edu.tw
(Received Mar. 31, 2016; revised and accepted May 19 & June 10, 2016)

Abstract

Multi-proxy signature (MPS) scheme makes a very important branch of the proxy signature scheme family, as they are applicable in many practical situations. The MPS scheme enables the actual signer to pass on their signing authority to plural proxy signers, where each proxy/delegated signer should contribute together to create a genuine MPS to make the whole thing work. In this work, we shall present an efficient MPS scheme that apply self-certified key and the notion of message recovery. The major advantage of our scheme is that the verification of the public keys, the verification of MPS, and recovery of the message can be carried out simultaneously. This reduces the computation cost and communication load dramatically. The security analysis of the proposed scheme includes thorough discussions over the security of the secret keys, the legitimacy of the public key of the signer's, along with unforgeability of our MPS scheme (MPSS). The performance analysis of our MPSS, reflects that our scheme, has an edge regarding computational complexity, over the schemes given in Wu et al.'s and Xie et al.'s.

Keywords: Discrete logarithm problem, message recovery, multi-proxy signature, proxy signature, self-certified key

1 Introduction

What is a proxy signature scheme? By definition, this signature scheme enables the other person called proxy signer to sign in place of actual signer, with due permission [5, 10, 25]. Mambo et al. [15, 16] first brought the design of proxy signature from some authorized proxy person. Since then, enormous researches have focused on refining this specific signature itself and on making them applicable to as many real-life situations as possible [1]. Among the possibilities explored was the question of how to transfer the power of signing to plural proxy signer at a time, and in 2000, Huang and Shi [6] answered the question by offering their MPS scheme, as an extension of the fundamental proxy or delegated signature mechanism. After that, many researchers have developed and presented their own variants [2, 7, 13, 14, 19, 20, 24, 29, 30], of the MPSS (MPS scheme). Typically in a MPSS, commonly the following three entities are involved: the original/actual signer, two or more proxy signers, and recipient of signature. Please note that all the proxy signers have to jointly create the MPSS and this makes the major difference between a MPSS and a fundamental proxy signature scheme.

To an adversary, any form of digital transaction can be a target for attack. For example, with a forged public key, an attacker can try to forge as the original signer or a proxy signer. To prevent forgery attacks from taking effect, it is a good idea to authenticate the public key of all the entities involved before they participate in any part of the cryptographic processes. A common practice to do the job here is to use a certificate-based public key cryptosystem, where any legal user or verifier can confirm the public key authenticity and the verification of information regarding identity of the signer by checking the certificate issued to each signer by the certificate author-

ity (CA) [8, 21]. However, certificate verification processes considerably increase both the computation cost and the communication load. In real time applications, in particular, when many users are trying to sign documents at the same time, it is extremely demanding for the system to handle the verification of multiple certificates simultaneously. To solve this problem, Shamir [22] presents a new cryptosystem based on the identity (ID-based) scheme. In such a system, the signer can be recognized through his public key. This way, certificates are no longer necessary, and therefore no certificate verification processes are needed. The shortcoming of this approach, however, is that the CA has knowledge of secret key of every signer, as the signer register himself. This may give the CA, a fair chance to pretend to be a genuine user. This is possible by creating a legitimate pair of keys for that user and no one identify that actually CA generates the pair of keys. In other words, public key verification remained a problem.

Girault [3] introduced the self-certified public key concept. In Girault's design, the registered user gets to determine their own secret key, while the public key for each user is generated by CA. In comparison with the certificate-based approach, this system runs on a much lower computation cost, and the communication load is also lighter [12, 23]. The validity of a public key is checked when a user participates in signature schemes where self-certified public keys are used. If the signature or public key of the user fails in verification process than the user's access will be denied.

In 1994, Nyberg et al. [17] offered the first signature with the ability of message recovery. In Nyberg et al.'s scheme, the message is sent along with the signature and is then recovered by the verifier. Since no hashing of message is required, the consumption of storage space and communication bandwidth is low. The security of their scheme relies on the discrete logarithm problem (DLP). In this kind of schemes, only a legitimate signer can broadcast the authentic signature corresponds to the message to a signature's verifier, and the verifier can obtain the message and verify the authenticity of the signature. This way, the communication overhead can be effectively reduced.

Wu, Hsu, and Lin (WHL) [27] proposed couple of MPS scheme, and their security relies on DLP and the elliptic curve discrete logarithm problem (ECDLP) respectively. They combined the concept of message recovery and the self-certified public key. Later, in 2012, Xie [28] showed that WHL scheme [27] is vulnerable to a warrant attack by proxy signer via revision of original warrant. This attack through warrant revision can launched either by the proxy or the actual signer. To fix the problem, Xie presents a provably secure signature scheme resists a warrant attack and an adaptive chosen message attack under existential forgery.

Inspired by the brilliant earlier works, we have also developed an efficient MPS scheme, by applying self-certified public keys and our scheme provides message recovery as well. The remaining of our work is managed as follows: To begin with, the proposed scheme will be presented in detail in next section, followed by Section 3, in which the security analysis of our scheme is given. The performance analysis is given in Section 4. Finally, we conclude our work in last section.

2 The Proposed MPS Scheme

The details of our proposed MPS scheme is given in this part. Let's first define some notations and parameters in Table 1 that we are going to use throughout this paper.

The CA generates p, q, g, and β as system parameters and makes them public but keeps α secret. The CA also assists registered users to create their secret and public key pairs. The proposed MPS scheme has the following phases: (1) User Registration Phase, (2) Delegation Parameter Generation Phase, (3) Multi-Proxy Signature Generation Phase, and (4) Signature Verification and Message Recovery Phase. The details of the above phases are given below:

1) User Registration Phase.

Suppose a user U_i with identity ID_i wishes to register with CA. To serve the purpose, he/she needs to present keys namely a secret key and an openly accessible public key paired up. Self-certified keys are generated as follows:

a. Each user U_i selects a random number $a_i \in Z_q^*$ as their master key and computes

$$v_i = g^{h(a_i \| ID_i)} \bmod p \tag{1}$$

and then sends it to CA over a secure channel.

b. Upon receiving (v_i, ID_i) from U_i, the CA chooses an integer $t_i \in Z_q^*$, which varies with time and computes the U_i's public key y_i and the witness w_i as follows:

$$y_i = v_i \cdot g^{t_i} - h(ID_i) \bmod p \tag{2}$$
$$w_i = t_i + \alpha \cdot \{y_i + h(ID_i)\} \bmod q \tag{3}$$

for each U_i and sends (y_i, w_i) to them respectively.

c. Upon receiving (y_i, w_i), each U_i computes his secret key

$$x_i = w_i + h(a_i \| ID_i) \tag{4}$$

and checks the validity of y_i, through the following equation

$$\begin{aligned} g^{x_i} &= \{y_i + h(ID_i)\} \cdot \beta^{y_i + h(ID_i)} \bmod p \\ &= Y_i \bmod p. \end{aligned} \tag{5}$$

Table 1: Notations

Notation	Description
(p, q)	Large primes, with $q\|p-1$.
g	Generator with order q, over $GF(p)$.
m_w	Message warrant.
$h(\cdot)$	One-way hash function [4, 9, 11].
(α, β)	The private and public key pair for CA, with $\beta = g^{\alpha} \bmod p$.
U_o	Denote the original/actual signer.
U_i	Denote the proxy/delegated signer, where $i = 1, 2, ...N$.
G	Group of proxy signers.
(x_i, y_i)	For signer the key pair of private and public key, where $i = 1, 2, ...N$.
ID_i	Represents identity of the signer, where $i = 0, 1, 2, ...N$.

This verification can be done as follows:

$$\begin{aligned} g^{x_i} &= g^{t_i+\alpha\{y_i+h(ID_i)\}+h(a_i\|ID_i)} \bmod p \\ &= g^{t_i} \cdot g^{\alpha\{y_i+h(ID_i)} \cdot g^{h(a_i\|ID_i)} \bmod p \\ &= v_i \cdot g^{t_i} \cdot \beta^{y_i+h(ID_i)} \bmod p \\ &= \{y_i + h(ID_i)\} \cdot \beta^{y_i+h(ID_i)} \bmod p \\ &= Y_i \bmod p. \end{aligned}$$

2) Delegation Parameter Generation Phase.

Now U_o wishes to transfer his authority of signing to N proxy signers G $= \{U_1, U_2, ...U_N\}$. U_o and U_i take the following steps to do the job:

a. U_o chooses a random integer $k_i \in Z_q^*$ and calculates

$$K_i = g^{k_i} \bmod p \tag{6}$$

$$K = \prod_{i=1}^{N} K_i \bmod p \tag{7}$$

$$H = h\big(\beta^{\sum_{i=0}^{N}(y_i+h(ID_i))} \cdot \prod_{i=0}^{N}(y_i + h(ID_i))\|m_w\|K\big) \tag{8}$$

$$\sigma_i = x_o \cdot N^{-1} \cdot H + k_i \bmod q \tag{9}$$

b. U_o transmits (σ_i, m_w) to each $U_i \in$ G and broadcasts (K_i, K, H).

c. After getting (σ_i, m_w) from U_o, each $U_i \in$ G verifies its authenticity through the equation

$$g^{\sigma_i} = \big(Y_o\big)^{N^{-1}\cdot H} \cdot K_i \bmod p$$

If this equation checks out, then only U_i agrees to his proxy share.

3) Multi-Proxy Signature Generation Phase

To generate a signature for message M, as an alternative of U_o, each $U_i \in G$ carries out the following calculations:

a. Each $U_i \in G$ selects a random integer value $b_i \in Z_q^*$ and evaluates

$$c_i = g^{b_i} \bmod p, \tag{10}$$

then transmits c_i to other users in group G.

b. Each U_i computes

$$\begin{aligned} c &= \{M\|h(M)\} \cdot \prod_{j=1}^{N} c_j \bmod p \\ \rho_i &= b_i + (\sigma_i + x_i \cdot H) \cdot h(m_w\|c\|K) \bmod q \end{aligned} \tag{11}$$

and sends ρ_i to other members in G.

c. Now each $U_i \in$ G has a collection of (c_j, ρ_j) received from all the other members of G. U_i checks the validity by computing

$$c_j \cdot \Big[\big(Y_o\big)^{N^{-1}\cdot H} \cdot \big(Y_j\big)^{H} \cdot K_j\Big]^{h(m_w\|c\|K)} = g^{\rho_j} \bmod p$$

if the above equation checks out, then U_i computes

$$\rho = \sum_{j=1}^{N} \rho_j \bmod q$$

Now the multi-proxy signature (K, c, ρ, m_w, H) is completed.

4) Signature Verification and Message Recovery Phase.

The verifier confirms the authenticity of the generated signature, through the equation

$$M\|h(M) = c \cdot g^{-\rho} \cdot \Big[\prod_{i=0}^{n}\big(Y_i\big)^{H} \cdot K\Big]^{h(m_w\|c\|K)} \bmod p. \tag{12}$$

Now with this recovered message M and its hash value, the verifier can ensure the authenticity of both M and the generated signature. The verification equation involves the public key's of both the proxy and actual signers, which can be automatically verified. This way, all three tasks, namely verification of public key, verification of signature, and recovery of message, can be completed in one stroke.

3 Security Analysis

This section serves to check the security aspects of our MPS scheme. The security of our scheme can be divided into three parts: safety of private keys, legitimacy of signers' public keys, and unforgeability of signatures.

1) Safety of private keys.

   a. Safety of private key (α) of CA.

   Suppose an adversary is looking to obtain CA's secret key α, which lies under the protection of DLP [18, 26]. To get α from Equation (3), the adversary faces great difficulty because of the lack of knowledge of the time variant secret t_i, which is only known to CA. It can be seen from Equation (2) that t_i is secure under DLP.

   b. Safety of secret key (x_i) of signer i.

   The secret key x_i of signer i is generated through the conduction of Equation (4), which depends on the hash value $h(a_i \| ID_i)$. It can be clearly from Equation (1) that the hash value is secure under the protection of DLP.

   Let an adversary or some delegated signers attempt to get the secret key x_o of actual signer U_o from Equation (9). However, it is not feasible for them due to unknown value k_i from Equation (6) and this k_i is secure because of DLP.

   c. Infeasible to obtain secret keys from public keys.

   It is not possible for an adversary to derive secret key of the actual signer U_o or any delegated signer U_i through intercepted data (c_i, ρ_i) or from a genuine multi-proxy signature (K, c, ρ, m_w, H). As we can see, with the value of σ_i (see Equation (9)) substituted into Equation (11), we come to

   $$\rho_i = b_i + \{(x_o \cdot N^{-1} \cdot H + k_i) + x_i \cdot H\} \cdot h(m_w \| c \| K) \bmod q,$$

   where there are still two unknowns parameters k_i and b_i securely under the protection of DLP (see Equations (6) and (10)). Therefore, there is no way an adversary can derive any secret key x_o or x_i from public data.

2) Legitimacy of signers' public keys.

   The secret key x_i, identity ID_i, and public key y_i must satisfy the verification Equation (5). In other words, for any fake secret key x_i', fake identity ID_i', and fake public key y_i' to take effect, all three must pass the test of Equation (5). An adversary can create a fake value ID_i' and randomly chooses private key x_i' at will, but to come by a public key y_i' to make the trio work is extremely difficult due to the obstruction of DLP. Alternatively, if the adversary tries to fix the public key y_i' and identity ID_i', then again DLP will get in the way and nullify the adversary's attempt to derive an effective secret key x_i'.

   Lastly, if the adversary tries another route to come by a valid identity ID_i' with the made-up duo of fixed keys x_i', y_i', the attempt will still fail because of the unbreakable reversal of OWHF [4, 9, 11].

3) Unforgeability of signatures.

   Suppose an adversary is looking to reuse a genuine multi-proxy signature (K, c, ρ, m_w, H) to illegally sign the message M'. To do the job, the adversary has to find an effective ρ, which is difficult due to the obstruction of DLP (see Equation (12)).

   On the other hand, in case an adversary attempted to obtain message M by using (K, c, ρ, m_w, H), then the adversary would have to overcome the reversal of OWHF.

Then, in the following passages, we shall demonstrate that how our MPSS fulfil all fundamental security properties including (1) Identifiability, (2) Prevention of misuse, (3) Unforgeability, (4) Undeniability, and (5) Verifiability.

1) Identifiablity.

   The multi-proxy signature (K, c, ρ, m_w, H) contains the message warrant m_w, by which the verifier can identify the proxy signer and actual signer.

2) Prevention of misuse.

   The warrant m_w carries a lot of information with it including type of delegation, delegation duration, as well as indication of which message is assigned to the proxy signers for signing. Therefore, the proxy signers cannot mistakenly sign a message they are not authorized by the actual signer to sign.

3) Unforgeability.

   The actual signer U_o is not able generate a valid MPS, because there is no way for U_o to collect the private keys of all the delegated signers. On the other hand, any delegated signer or any other person cannot counterfeit a MPS either due to the lack of the actual signer's private key, which is protected due to intractability of DLP.

4) Undeniability.

   The components c and ρ of the proposed signature (K, c, ρ, m_w, H) are collectively completed by all the proxy signers, and therefore no $U_i \in G$ can deny his signature.

5) Verifiability.

   With the correctness of the verification confirmed, the verifier can authenticate the signature and identify, whether the signed message corresponds to the proxy warrant.

Table 2: Computational complexity comparison

Phases	WHL [27]	Xie's [28]	Our scheme
Registration	$4nT_e + 5nT_m + 5nT_h + nT_i$	$4nT_e + 5nT_m + 5nT_h + nT_i$	$4nT_e + 3nT_m + 2nT_h$
Proxy Key Generation	$5nT_e + 5nT_m + (3n+1)T_h + (n+1)T_i$	$(4n+1)T_e + (7n+3)T_m + (4n+1)T_h + (n+1)T_i$	$(4n+1)T_e + (5n+2)T_m + (2n+1)T_h + (n+1)T_i$
Multi Proxy Sign Gen	$(5n^2 - 3n)T_e + (6n^2 - 4n)T_m + 2n^2T_h$	$(4n^2 - 3n)T_e + (6n^2 - 3n)T_m + 2n^2T_h$	$(4n^2 - 3n)T_e + (5n^2 - 2n)T_m + (n^2+1)T_h$
Signature Verification	$4T_e + (2n+5)T_m + (2n+5)T_h$	$4T_e + (2n+5)T_m + (2n+4)T_h + T_i$	$4T_e + (n+4)T_m + (n+2)T_h + T_i$
Total Cost	$(5n^2+6n+4)T_e + (6n^2+8n+5)T_m + (2n^2+10n+6)T_h + (2n+1)T_i$	$(4n^2+5n+4)T_e + (6n^2+11n+8)T_m + (2n^2+11n+8)T_h + (2n+1)T_i$	$(4n^2+5n+5)T_e + (5n^2+6n+6)T_m + (n^2+5n+4)T_h + (n+2)T_i$

Table 3: Communication cost comparison

Phase	WHL [27]	Xie's [28]	Our scheme
Proxy Key Generation	$(n+1)\cdot\|p\| + 2n\cdot\|q\|$	$(n+1)\cdot\|p\| + (2n+1)\cdot\|q\|$	$(n+1)\cdot\|p\| + (2n+1)\cdot\|q\|$
Multi-proxy Sign Gen	$n\cdot(\|p\| + \|q\|)$	$n\cdot(\|p\| + \|q\|)$	$n\cdot(\|p\| + \|q\|)$
Signature Verification	$2\cdot\|p\| + 3\cdot\|q\|$	$2\cdot(\|p\| + \|q\|)$	$2\cdot(\|p\| + \|q\|)$
Total	$(2n+3)\cdot\|p\| + (3n+3)\cdot\|q\|$	$(2n+3)\cdot\|p\| + (3n+3)\cdot\|q\|$	$(2n+3)\cdot\|p\| + (3n+3)\cdot\|q\|$

4 Performance Analysis

Now we shall see comparison of the complexity of the proposed MPSS with [27] and [28]. We do not consider the complexity of addition and subtraction operations as they are negligible.

As Table 2 shows, the proposed scheme is obviously superior to the other two schemes as far as computational complexity is concerned.

As Table 3 shows, the three schemes have the same total communication cost and therefore are equally efficient in this matter.

5 Conclusion

In this paper, we present a new MPSS using self-certified public keys. The security analysis has established the security of the secret keys, the genuineness of the public key of signers, as well as the unforgeability of the proposed scheme. Furthermore, the performance analysis has proven that the new scheme has an edge over the WHL scheme and Xie's scheme with respect to the computational load.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. In addition, this research was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract no.: MOST 104-2221-E-165-004 and MOST 104-2221-E-030-002.

References

[1] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights", *Journal of Cryptology*, vol. 25, no. 1, pp. 57–115, 2012.

[2] F. Cao, and Z. Cao, "A secure identity-based multi-proxy signature scheme", *Computers & Electrical Engineering*, vol. 35, no. 1, pp. 86–95, 2009.

[3] M. Girault, "Self-certified public keys", *Advances in Cryptology-EUROCRYPT'91*, pp. 490-497, 1991.

[4] M. S. Hwang, C. C. Lee, and T. H. Sun, "Data error locations reported by public auditing in cloud storage service", *Automated Software Engineering*, vol. 21, vol. 3, pp. 373-390, 2014.

[5] M. S. Hwang, C. C. Lee, and S. F. Tzeng, "A new proxy signature scheme for a specified group of verifiers", *Information Sciences*, vol. 227, pp. 102-115, 2013.

[6] S. J. Hwang and C. H. Shi, "A simple multi-proxy signature scheme", *In: Proc.* 10^{th} *National Conf. on Information Security*, Hualien, Taiwan, ROC, pp. 134–138, 2000.

[7] J. H. Ji, D. Li, and M. Wang, "New proxy multi-signature, multi-proxy signature and multi-proxy multi-signature schemes from bilinear pairings", *Chinese Journal of Computers-Chinese Edition*, vol. 27, no. 10, pp. 1429–1435, 2004.

[8] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[9] C. C. Lee and Y. M. Lai, "Toward a secure single sign-on mechanism for distributed computer networks", *The Computer Journal*, vol. 58, no. 4, pp. 934-943, 2015.

[10] C. C. Lee, T. C. Lin, S. F. Tzeng, and M. S. Hwang, "Generalization of proxy signature based on factorization", *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 3, pp. 1039-1054, 2011.

[11] C. T. Li, C. Y. Weng, C. C. Lee, and C. C. Wang, "A hash based remote user authentication and authenticated key agreement scheme for the integrated EPR information systems", *Journal of Medical Systems*, vol. 39, no. 11, pp. 1-11, 2015.

[12] J. Li and S. Wang, "New efficient proxy blind signature scheme using verifiable self-certified public key", *International Journal of Network Security*, vol. 4, no. 2, pp. 193–200, 2007.

[13] X. Li, K. Chen, and S. Li, "Multi-proxy signature and proxy multi-signature schemes from bilinear pairings", *Parallel and Distributed Computing: Applications and Technologies*, Springer Berlin Heidelberg, pp. 591–595, 2004.

[14] C. Y. Lin, T. C. Wu, and J. J. Hwang, "Multi-proxy signature schemes for partial delegation with cheater identification", *Proceeding of IWAP 2*, 2002.

[15] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: Delegation of the power to sign messages", *IEICE Trans. Fundamentals*, vol. E79-A, no. 9, pp. 1338–1354, 1996.

[16] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation", *In: Proc. 3rd ACM Conf. on Computer and Communications Security*, pp. 48–57, 1996.

[17] K. Nyberg and R. Rueppel, "Message recovery for signature schemes based on the discrete logarithm problem", *Designs, Codes and Cryptography*, vol. 7, no. 1, pp. 61-81, 1996.

[18] R. Padmavathy and C. Bhagvati, "A new method for computing DLP based on extending smooth numbers to finite field for ephemeral key recovery", *International Journal of Network Security*, vol. 17, no. 3, pp. 251–262, 2015.

[19] C. Pan, S. Li, Q. Zhu, C. Wang, and M. Zhange, "Notes on proxy signcryption and multi-proxy signature schemes", *International Journal of Network Security*, vol. 17, no. 1, pp. 29–33, 2015.

[20] R. A. Sahu, and S. Padhye, "Provable secure identity-based multi-proxy signature scheme", *International Journal of Communication Systems*, vol. 28, no. 3, pp. 497–512, 2015.

[21] K. R. Santosh, C. Narasimham, and P. Shetty, "Cryptanalysis of multi-prime RSA with two decryption exponents," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 40–44, 2016.

[22] A. Shamir, "Identity-based cryptosystems and signature schemes", *Advances in cryptology*, pp. 47-53, 1984.

[23] Z. Shao, "Improvement of threshold signature using self-certified public keys", *International Journal of Network Security*, vol. 1, no. 1, pp. 24–31, 2005.

[24] N. Tiwari and S. Padhye, "Provable secure multi-proxy signature scheme without bilinear maps", *International Journal of Network Security*, vol. 17, no. 6, pp. 736–742, 2015.

[25] S. F. Tzeng, C. C. Lee, and M. S. Hwang, "A batch verification for multiple proxy signature", *Parallel Processing Letters*, vol. 21, no. 1, pp. 77–84, 2011.

[26] C. Wu, X. Du, and Z. Jiang, "Linear complexity of a family of pseudorandom discrete logarithm threshold sequences", *International Journal of Network Security*, vol. 18, no. 3, pp. 487–492, 2016.

[27] T. S. Wu, C. L. Hsu, and H. Y. Lin, "Self-certified multi-proxy signature schemes with message recovery", *Journal of Zhejiang University SCIENCE A*, vol. 10, no. 2, pp. 290-300, 2009.

[28] Q. Xie, "Provably Secure Self-certified Multi-proxy Signature with Message Recovery", *Journal of Networks*, vol. 7, no. 10, pp. 1616-1623, 2012.

[29] Q. Xue and Z. Cao, "Improvement of multi-proxy signature scheme", *Proceedings of The Fourth International Conference on Computer and Information Technology*, 2004.

[30] Y. Yu, Y. Sun, and B. Yang, "Multi-proxy signature without random oracles", *Chinese Journal of Electronics*, vol. 17, no. 3, pp. 475–480, 2008.

M. K. Chande received the B. S. and M. S. degrees in mathematics from Pt. Ravishankar Shukla University, Raipur (C.G.), India, in the year 1997 and 1999 respectively. He is currently serving as the capacity of an assistant professor in the Department of Applied Mathematics, Shri Shankaracharya Institute of Professional Management and Technology, Raipur (C.G.), India. He is life member of Cryptology Research Society of India (CRSI). Currently he is doing his Ph. D. degree from School of Studies in Mathematics, Pt. Ravishankar Shukla University, Raipur (C.G.), India. His research interest includes cryptography, analysis, design and applications of digital signatures.

C. C. Lee received the Ph.D. degree in Computer Science from National Chung Hsing University (NCHU), Taiwan, in 2007. He is currently an Associate Professor with the Department of Library and Information Science at Fu Jen Catholic University. Dr. Lee is currently as an editorial board member of International Journal of Network Security, Journal of Computer Science, Cryptography, and International Journal of Internet Technology and Secured Transactions. He also served as a reviewer in many SCI-index journals, other journals, other conferences. His current research interests include

data security, cryptography, network security, mobile communications and computing, wireless communications. Dr. Lee had published over 100+ articles on the above research fields in international journals. He is a member of IEEE, the Chinese Cryptology and Information Security Association (CCISA), the Library Association of The Republic of China, and the ROC Phi Tau Phi Scholastic Honor Society.

C. T. Li received the Ph.D. degree in Computer Science and Engineering from National Chung Hsing University, Taiwan, in 2008. He is currently an Assistant Professor of the Department of Information Management, Tainan University of Technology, Tainan, Taiwan. His research interests include information security, wireless sensor networks, mobile computing, and security protocols for ad hoc networks.

A Color Image Encryption Scheme Based on Arnold Scrambling and Quantum Chaotic

Hui Liu, Cong Jin
(Corresponding author: Cong Jin)
School of Computer, Central China Normal University
Wuhan, 430079, P.R. China
(Email: jincong@mail.ccnu.edu.cn)
(Received Apr. 20, 2016; revised and accepted June 10, 2016)

Abstract

In recent years, several algorithms of image encryption have been proposed independently. In this paper, an algorithm of image encryption based on general two-dimensional Arnold transform with keys and quantum chaotic map is proposed. First, the key streams are generated by the two-dimensional logistic map as initial conditions and parameters. Second, general Arnold scrambling algorithm with keys is exploited to permute the pixels of color components, R, G and B. Finally a serial of pseudo-random numbers generated by the quantum chaotic map is applied to modify the value of diffused pixels. In order to get the high randomness and the high complexity, the two-dimensional logistic map and quantum chaotic map are coupled with nearest-neighboring coupled-map lattices. Theoretical analyses and computer simulations confirm that the new algorithm has high level of security.

Keywords: Arnold scrambling algorithm, coupled-map lattices, image encryption, quantum chaotic map, two-dimensional logistic map

1 Introduction

1.1 Background

With the rapid growth of the transmission over the Internet, the security of digital image acquires a major concern. So image encryption becomes a hot area and a challenging task. In order to protect personal information, various image encryption algorithms are designed and proposed such as two-dimensional cellular automata-based method [20], Henon chaotic map [10, 21], Chen's hyper-chaotic system [12], Arnold transform [3, 4] and so on. As a classical algorithm Arnold transform has many advantages over others. But an obvious weakness is that it only can be applied to square matrix $N \times N$ and an ideal encryption scheme should not have periodicity. In this paper an excellent method is proposed to solve the problem. Chaotic systems have many good features such as sensitivity to initial conditions and parameters, mixing property, high efficiency and ergodicity. Inspired by the subtle similarity between chaotic systems and cryptosystem, various encryption algorithms based on chaotic map are proposed in the literature. Herein, quantum chaotic system is applied to generate pseudo-random sequence to encrypt color images in the proposed cryptosystem.

1.2 Related Work

Image is one of the most important information representation models and widely used in modern society. An international standard of encryption algorithm is not only suitable for a partial compression algorithm but permutation and diffusion properties. Permutation and diffusion properties are satisfied in cellular automata-based (CA) image system [20]. Ping proposed a novel CA-based multiple image encryption by using a kind of two-dimensional reversible CA, and by using a circular chaining mode of operation. The proposed method allows images to be processed in a 2-D way and makes the statistical information of each plain image in the group hidden in all cipher images.

In order to disturb the high correlation among pixels, the Arnold cat map [3, 4] is a good scrambling tool which has been used widely in various cryptographic and steganographic applications. Chen et al. [3] analyzed the period distribution of the cat map systematically. [4] reported a new image encryption algorithm based on singular value decomposition and Arnold transform. However, in all of these algorithms have two weaknesses, one is that the iteration times are very limited; the other is that the width and height of the plain-image must be identical. Here we propose perfect methods to solve these problems so that the proposed algorithm can be accepted widely.

Chaos-based cryptographic scheme has many brilliant advantages different from other algorithms such as sensitivity to initial conditions and parameters, mixing property, high efficiency non-periodicity and control parame-

ters [7, 15]. In recent years various encryption algorithms based on chaotic map are proposed [26, 27]. Wang and Guo [27] utilized a logistic map for generating a matrix to diffuse the left block of the plain image and then the diffused image was used as the right block of the cipher image. Tang [26] presented an algorithm dividing an input image into overlapping blocks, shuffling image blocks to make initial encryption, exploiting a chaotic map and Arnold transform to generate secret matrices, and achieving final encryption by conducting exclusive OR operations between corresponding elements of each block and a random secret matrix. Jawad [9] enhanced the security level of conventional Blowfish algorithm (BA) for color image encryption by modifying it with new F-function. And the dynamic S-box and XOR operator were generated from the F-function via four-dimensional hyperchaotic map. Lately, in [2] quantum chaos theory becomes a tool that can be used to improve the quality of pseudo-random number generators. The randomness and non-periodicity of quantum chaotic map are successfully verified by statistical complexity and the normalized Shannon entropy. So we apply these characteristics to encrypt the color image for achieving the high randomness and acquiring the non-periodicity that is caused by Arnold transform.

Generally, there are two main stages in the structure of chaos-based algorithm which consists of permutation and diffusion stages. The permutation stage shifts the position of pixels of the plain-image by some chaotic map. General Arnold transform with keys finishes the permutation stage and provides an enough large key space. The diffusion stage modifies the pixels values of shuffled image via chaotic sequences so that a minor change in one pixel of the plain-image causes a totally different cipher-image. Chaotic sequences generated by quantum chaotic map accomplished the diffusion stage and improved the randomness and complexity of the proposed cryptosystem. The diffusion-permutation-based algorithm should have a large key space and the long periodicity of permutation to increase the security. For this purpose, many researchers turn to find some improved chaos-based algorithms with large key spaces and good permutation and diffusion techniques.

1.3 Contribution and Organization

In order to encrypt all color images by Arnold transform algorithm, it is essential to make up the rectangular image into a square. Without loss of generality, we assume that the size of the color plain-image P is $W \times H$, where W is the width of the image, H is the height of the image. Through the method the plain-image is converted into a new image whose size is $N \times N$. Due to the color image that is composed of three color components, we convert three components into three matrices, namely R, G, B. General Arnold transform with keys means that parameters of the matrix A is a set of secret values. We add the matrix $(k\mu, k\nu)^T$ as secret values during the process that Arnold transform is iterated n times. The experiment proves that the chaos character is better when $n = 6$. So we get three different matrices $(k\mu_i, k\nu_i)^T$ $(i = 1, 2, 3)$ as keys to improve the high randomness and enlarge the key space. And then quantum chaotic map [1, 6, 24] is applied to generate three matrices X, Y, Z of size $N \times N$ to encrypt three matrices R, G and B. In this process, the initial condition of quantum chaotic map is a pseudo-random number, which is altered with the time of iteration. For the high complexity and the high randomness, in this paper chaotic maps are coupled with nearest-neighboring coupled-map (NCML), which extremely increases the security and sensitivity of the proposed algorithm.

The major contribution of the proposed algorithm include following points:

1) Provide a method (Equation (10)) to map an arbitrary value into a given interval to meet the demands of two-dimensional logistic map and quantum chaotic map;

2) Add matrices $(k\mu_i, k\nu_i)^T (i = 1, 2, 3)$ as keys into general Arnold transform to enlarge the key space and improve the randomness;

3) Key generator is an address mapping table, which is generated by two-dimensional logistic map. According to session keys we obtain initial conditions and parameters so that improve the sensitivity of the key generator.

The rest of this paper is organized in the following manners: Section 2 introduce the basic theory of the proposed cryptosystem. Section 3 the proposed cryptosystem is explained detailed. Section 4 simulation results and security analysis are proposed. Finally the conclusion is drawn in Section 5.

2 Basic Theory of the Proposed Cryptosystem

2.1 Two-dimensional Logistic Map

In this paper two-dimensional logistic map is applied whose definition is as follows: The two-dimensional logistic map is described as [14, 29]:

$$\begin{aligned} \varphi_1(x_n) &= \mu_1 x_n(1 - x_n) + \nu_1 y_n^2 \\ \varphi_1(y_n) &= \mu_2 y_n(1 - y_n) + \nu_2(x_n^2 + x_n y_n) \end{aligned} \tag{1}$$

when $2.75 < \mu_1 \leq 3.4$, $2.75 < \mu_2 \leq 3.45$, $0.15 < \nu_1 \leq 0.21$ and $0.13 < \nu_2 \leq 0.15$, the system can generate pseudo-numbers in the region (0,1]. All parameters are generated by key generator.

2.2 General Arnold Transform with Keys

We set that the location of the plain-image pixel is (x, y), the location of the cipher-image pixel is (x', y'). The

definition of general Arnold transform is given in [25]:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A \begin{bmatrix} x \\ y \end{bmatrix} (modN), A = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \quad (2)$$

where we set $N = 256$. When $a = b = 1$, Equation (2) is a classical two-dimensional Arnold map. In order to improve security of the cryptosystem, parameters a and b are used as secret keys, which are generated by key generator. Because Arnold transform is a bijection transform, the result of iterating Equation (2) k times still is a bijection transform. In other words, after the process of iteration for k times, point (x, y) become (x', y') and (x', y') is the one and only one point. Due to the result of orthogonal transformation is a limited discrete set, we can add a matrix $(k\mu,k\nu)^T$ as a set of secret keys to enlarge the key space. So we get general Arnold transform with keys whose definition is as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A^n \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} k\mu \\ k\nu \end{bmatrix} (modN), A = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \quad (3)$$

where n is iteration times of the matrix A. According to the inverse transformation of Equation (3), the corresponding decryption algorithm is shown as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = A^{-n} \begin{bmatrix} x' - k\mu \\ y' - k\nu \end{bmatrix} (modN), A^{-n} = \begin{bmatrix} ab+1 & -a \\ -b & 1 \end{bmatrix} \quad (4)$$

2.3 Quantum Chaotic Map

Dissipative quantum systems are often described in where the system is coupled to a path of harmonic oscillators to construct a quantum logistic map [1, 6, 24] with quantum corrections. In [1], authors analyze the effects of quantum corrections and state $\alpha = <\alpha>+\delta\alpha$, where $\delta\alpha$ shows a quantum fluctuation about $<\alpha>$. Furthermore, they prove that the very lowest-order quantum corrections can yield the chaotic map as follows:

$$\begin{aligned} \varphi_2(x'_n) &= r(x'_n - |x'_n|^2) - ry'_n \\ \varphi_2(y'_n) &= -y'_n \exp(-2\beta) + \exp(-\beta)r[(2 - x'_n - x'^*_n)y'_n \\ &\quad - x'_n z'^*_n - x'^*_n z'_n] \\ \varphi_2(z'_n) &= -z'_n \exp(-2\beta) + \exp(-\beta)r[2(1 - x'^*_n)z'_n \\ &\quad - 2x'_n y'_n - x'_n] \end{aligned} \quad (5)$$

where $x' = <\alpha>$, $y' = <\delta\alpha \dagger \delta\alpha>$, $z' = <\delta\alpha\ \delta\alpha>$, and β is dissipation parameter. Generally y, x'_n , y'_n and z'_n are complex numbers with $x'_n{}^*$ being the complex conjugate of x'_n and similarly for z'_n. However, if we set the initial value to be real number, then all successive value will also be real. According to [2], the range of the parameters as follows: $0 \le x'_n \le 1$, $0 \le y'_n \le 0.1$, $0 \le z'_n \le 0.2$, $x'_n{}^* = x'_n$, $z'_n{}^* = z'_n$. They conclude that the best value of the control parameter (r) and dissipation parameter (β) are $r = 3.99$, and $\beta \ge 6$. So we set $r = 3.99$, $\beta = 6$, and iterate Equation (5) with real initial parameters x'_0, y'_0,z'_0, $x'_0{}^*$ and $z'_0{}^*$.

2.4 Nearest-neighboring Coupled-map Lattices

The two-dimensional logistic map and the quantum chaotic map proposed in Sections 2.1 and 2.3 are independently coupled with NCML [5, 11] as follows:

$$z_{n+1}(j) = (1 - \varepsilon)\varphi(z_n(j + 1)) + \varepsilon\varphi(z_n(j + 1)) \quad (6)$$

where $n = 0, 1,..., L$-1 is the time index; $j = 1, 2, ... , T$ is the lattice state index; function φ represents a chaotic map such as φ_1, φ_2; $\varepsilon\in(0,1)$ is a coupling constant; L is the length of the plain-text; and T is maximum value of lattice state index. Here, T is chosen as 2 and 3 for the two-dimensional logistic map and the quantum chaotic map, while the other parameter is selected as ε= 0.001 to have good chaotic properties [5, 11]. Moreover, the periodic boundary condition, i.e., $z_n(j+T) = z_n(j)$ is imposed into this system.

Applying Equation (1) to Equation (6), the coupling of two-dimensional logistic map is defined as follows:

$$\begin{aligned} x_{n+1} &= (1 - \varepsilon)\varphi(x_n) + \varepsilon\varphi(y_n) \\ y_{n+1} &= (1 - \varepsilon)\varphi(y_n) + \varepsilon\varphi(x_n) \end{aligned} \quad (7)$$

and by applying Equation (2) to Equation (6), the coupling of quantum chaotic map is defined as follows:

$$\begin{aligned} x'_{n+1} &= (1 - \varepsilon)\varphi(x'_{n+1}) + \varepsilon\varphi(y'_{n+1}) \\ y'_{n+1} &= (1 - \varepsilon)\varphi(y'_{n+1}) + \varepsilon\varphi(z'_{n+1}) \\ z'_{n+1} &= (1 - \varepsilon)\varphi(z'_{n+1}) + \varepsilon\varphi(x'_{n+1}) \end{aligned} \quad (8)$$

Iterating Equation (7) and Equation (8), the required key streams for the proposed cryptosystem are produced.

3 Proposed Cryptosystem

In this section, we combine the generation process with the image processing, the permutation process and the diffusion process. The architecture of the overall image encryption cryptosystem using the proposed algorithm is shown in Figure 1.

3.1 Generation of the Initial Conditions and Parameters

The proposed cryptosystem utilizes a 128-bit external secret key, K, which is divided into 8-bit blocks, k_i, referred to as session keys. The 128-bit external secret key is given by:

$$K = k_1, k_2, ..., k_{16}. \quad (9)$$

In order to increase the security of the proposed algorithm, we apply the two-dimensional logistic map Equation (1) and nearest-neighboring coupled-map lattices Equation (6) so that the initial conditions and parameters of the system are extremely sensitive to the changes in even a single bit in the 128-bit secret key. The detailed process of key generator is described as follows:

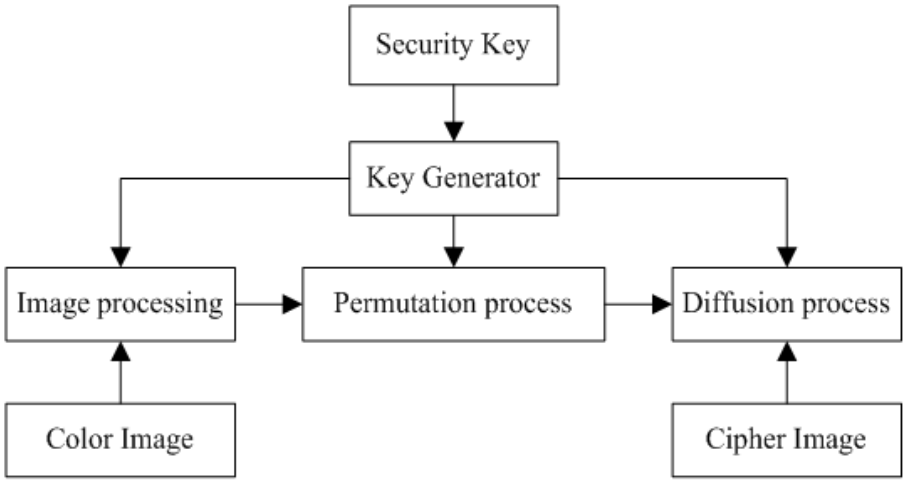


Figure 1: Overall architecture of the proposed cryptosystem

Step 1: Apply k_1, k_2, k_3, k_4 to generate μ_1, μ_2, ν_1, ν_2 respectively. We have known that when $2.75 < \mu_1 \leq 3.4$, $2.75 < \mu_2 \leq 3.45$, $0.15 < \nu_1 \leq 0.21$ and $0.13 < \nu_2 \leq 0.15$ the two-dimensional logistic map generates chaos. We set $a < t_i \leq b$, the initial conditions and parameters of system are derived as follows:

$$t_i = (\frac{k_i}{256} \times 100) mod[(b-a) \times 100] \div 100 + a \quad (10)$$

where we set $\mu_1 = t_1$, $\mu_2 = t_2$, $\nu_1 = t_3$, $\nu_2 = t_4$. So for the different k_i we can get different t_i and make sure that μ_1, μ_2, ν_1, ν_2 are in the region that the system generate chaos.

Step 2: Apply k_5, k_6, ... , k_{16} as initial condition to generate other key values. t_{max} = max([k_5, k_6, ... , k_{16}]). t_{min} = min([k_5, k_6, ... , k_{16}]). t_{ssv} = min([k_5, k_6, ... , k_{16}] - t_{min}). We set $x_0 = t_{min} \div 256$, $y_0 = t_{ssv} \div 256$, and iterate Equation (7) for ceil($t_{max} \div 2$) times with μ_1, μ_2, ν_1, ν_2, x_0, y_0 and then save their output in a new vector E whose size is $2 \times$ceil($t_{max} \div 2$). Apply the following Equation (11):

$$t_i = E_{k_i} \quad (11)$$

where $i = 5, 6, \ldots, 16$ and t_i are in the region (0, 1].

Step 3: In order to improve randomness and complexity of the encryption algorithm and broaden the key space, According to Equation (4) three sets of secret keys, a_i, b_i and $(k\mu_i, k\nu_i)^T$ are required to encrypt three component of the color image R, G, B respectively. Without loss of generality, we assume that the size of the color plain-image P is $W \times H$. Apply the transformation as the following equation to t_5, t_6, t_7:

$$\begin{aligned} a_{i-4} &= [floor(t_i \times W \times H) mod 256]/16 \\ b_{i-4} &= [floor(t_i \times W \times H) mod 256] mod 16 \end{aligned} \quad (12)$$

where a_i, b_i (i = 1, 2, 3) are the first four digits and the last four digits of eight-digit binary number respectively.

Apply the transformation as following equations to a_8, a_9, a_{10}:

$$ku_{i-7} = floor(t_i \times W \times H) mod 256 \quad (13)$$

Apply the transformation as following equations to a_{11}, a_{12}, a_{13}:

$$ku_{i-10} = floor(t_i \times W \times H) mod 256 \quad (14)$$

Step 4: Recalling as mention in Section 2.3, $y'_n \in [0,0.1]$, $z'_n \in [0,0.2]$. Applying Equation (10) analogously initial parameters x'_0, y'_0, z'_0 are derived as follows:

$$\begin{aligned} x'_0 &= t_{14} \\ y'_0 &= [(t_{15} \times 10) mod 1] \div 10 \\ z'_0 &= [(t_{15} \times 10) mod 2] \div 10 \end{aligned} \quad (15)$$

To this end, all initial conditions and parameters are generated. The proposed chaotic algorithm is greatly sensitive to secret key so that even a change in the secret key causes completely different results; as a result, the proposed algorithm with total complexity of 2^{128} can resist against any key sensitivity attack and any bruteforce attack.

3.2 Proposed Encryption Algorithm

Due to Arnold transform is not adapt to image $N \times N$, it is essential to transform image $W \times H$ into $N \times N$. we give the following equation to meet the demand:

$$N = \max([W, H]) \quad (16)$$

where set N is a bigger value between W and H. When $W = H$, $N = W = H$. In other words, if the image is

square, it remains unchanged; otherwise it will be amplified. Pixel values of increased part of the image are filled with random numbers, which are generated by the random function. It not only improves the randomness of the cryptosystem, but also if we can not get the real width and height of plain-image before decryption, we can not finish the decryption. We assume that the color plain-image P of $W \times H$ becomes P' of $N \times N$ by the transformation above. In this process we convert the matrix P with red green and blue components into three matrices R, G and B. Taking an example of the matrix R, the detailed encryption algorithm is described as follows:

Permutation process:

The process applies pseudo-random key streams generated by Equation (12), Equation (13) and Equation (14) according to Section 3.1 to permute pixels of the color image. Substituting a_1, b_1 and $(k\mu_1, k\nu_1)^T$ into Equation (3) and iterate it for n times. According to the experiment we find that when $n = 6$ the proposed cryptosystem performs better. Apply the same permutation process into G and B respectively, the plain-image becomes a cipher-image after n times iteration, namely, Matrices R, G and B all becomes R',G' and B'.

Diffusion process:

Step 1: Set L=$N \times N$ and generate the initial condition $(x'_0,\ y'_0,\ z'_0)$ according to Section 3.1 and iterate Equation (8) m+L times and discard the former m values to avoid harmful effects. Where m also can be as a secret key, we set $m = 13$ for convenience. Discarding the first m result and Sorting these L values as X= $\{x_{m+1}, x_{m+2}, \ldots, x_{m+L}\}$, Y= $\{y_{m+1}, y_{m+2}, \ldots, y_{m+L}\}$ and Z= $\{z_{m+1}, z_{m+2}, \ldots, z_{m+L}\}$.

Step 2: Transforming three matrices R',G' and B' into vectors $\overrightarrow{R'}$=$\{r_1, r_2, \ldots, r_L\}$,$\overrightarrow{G'}$ = $\{g_1, g_2, \ldots, g_L\}$, and $\overrightarrow{B'}$=$\{b_1, b_2, \ldots, b_L\}$ respectively.

Step 3: Applying the encryption transformation as the following equations:

$$\begin{aligned} C_{ri} &= ((floor(r_{m+i} \times W \times H \times k_6 \times k_7 \times k_9 \times k_{10} \\ &\quad \times k_{12} \times k_{13}) mod 256) \oplus r_i \\ C_{gi} &= ((floor(g_{m+i} \times W \times H \times k_5 \times k_7 \times k_8 \times k_{10} \\ &\quad \times k_{11} \times k_{13}) mod 256) \oplus g_i \\ C_{bi} &= ((floor(b_{m+i} \times W \times H \times k_5 \times k_6 \times k_8 \times k_9 \\ &\quad \times k_{11} \times k_{12}) mod 256) \oplus b_i \end{aligned} \quad (17)$$

where set initial values $i = 1$.Set $i = i$+1 and then iterating this step until $i \leq L$ we can get three matrices C_r, C_g and C_b.

Remark 1. M *mod* N *involves modulo operation giving a integer result between 0 and* N.

Remark 2. *ceil(a) returns the smallest integer value that is bigger than or equal to the value of a.*

Remark 3. max*(*[k_1, k_2, ... , k_n]*) returns the biggest value among all of them.*

Remark 4. min*(*[k_1, k_2, ... , k_n]*) returns the smallest value among all of them.*

Obviously the generation of the key stream depends on the 128-bit external secret key, K, and the width W, the height H of plain-image. The generation of initial conditions and parameters are derived by the two-dimensional logistic map and the nearest-neighboring coupled-map lattices. And the key stream is chosen from an array of chaotic sequence, which makes sure that cryptosystem has a high complexity, sensitivity and randomness. In the encryption process, the Arnold transform with keys is applied to permute the pixels of color components. And the quantum chaotic map is exploited to generate the key streams to modify the value of diffused pixels.

3.3 Proposed Decryption Algorithm

The decryption process is similar to the encryption one, achieved in the reverse order. In decryption process transforming matrices C_r, C_g and C_b into three vectors $\overrightarrow{C_r}$= $\{r_1, r_2, \ldots, r_L\}$,$\overrightarrow{C_g}$ = $\{g_1, g_2, \ldots, g_L\}$, and $\overrightarrow{C_b}$= $\{b_1, b_2, \ldots, b_L\}$ respectively. the detail decryption algorithm is described as follows:

Step 1: Apply the external 128-bit secret key used in the encryption process. According to Section 3.1 generate the initial conditions and parameters.

Step 2: Substituting the initial condition $(x'_0,\ y'_0,\ z'_0)$ and iterating Equation (8) m+L times, discarding the former m values to avoid harmful effects, where $m = 13$.

Step 3: Sorting these values X= $\{x_{m+1}, x_{m+2}, \ldots, x_{m+L}\}$, Y= $\{y_{m+1}, y_{m+2}, \ldots, y_{m+L}\}$ and Z= $\{z_{m+1}, z_{m+2}, \ldots, z_{m+L}\}$. Setting $i = 1$ and iterate Equation (17)) until $i = L$ we can get three vectors $\overrightarrow{C'_r}$, $\overrightarrow{C'_g}$ and $\overrightarrow{C'_b}$.

Step 4: We convert these vectors into three matrices ${R'}_r$, ${G'}_g$ and ${B'}_b$ whose size are all $N \times N$. Substituting parameters a_i, b_i and the initial condition $(k\mu_i, k\nu_i)^T$ $(i = 1, 2, 3)$, and then using the encryption algorithm Equation (4) we get R', G', B' of the image, According to the width W and the height H of the plain-image we tailor R', G', B' and get plain values of R, G and B. In this way the encryption process finished.

4 Performance and Security Analysis

We have done many measures to check the security and performance of the proposed cryptosystem. These measures consist of statistical analysis, key sensitivity analy-

Table 1: The related correlation coefficient between plain-image and cipher-image

Scan direction	Lena					
	Plain-image			Cipher-image		
	R	G	B	R	G	B
Horizontal	0.972978	0.954127	0.938846	0.001418	0.000082	-0.002191
Horizontal	0.981110	0.951084	0.934597	-0.007127	0.000587	0.000086
Vertical	0.958757	0.934720	0.915541	0.000700	0.000647	0.004526

sis, key space analysis, speed performance. Each of these measures is shown in detail in the following subsections.

4.1 Statistical Analysis

4.1.1 Histogram of Encrypted Image

An ideal cipher-image should has a uniform frequency distribution. From Figures 2, 3, 4 and 5, it is obvious that the histogram of cipher-image are independent of the type of plain-image such as binary, gray level and are nearly uniform and significantly different from the histogram of the original images. Hence it dose not provide any useful statistic data in the cipher-image to trigger any statistical attacks to the algorithm.

4.1.2 Correlation of Two Adjacent Pixels

In order to get the correlation of two adjacent pixels we have selected 3000 pairs of two adjacent pixels from plain-image and cipher-image randomly for the experiment and have calculated the correlation coefficients as follows:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$$

$$Cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$$

$$r_{xy} = \frac{Cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad (18)$$

The x and y represent gray-level values of two adjacent pixels. The distribution of two horizontally adjacent pixels of R, G and B components of plain-image and cipher-image Lena is shown in Figure 6.

Table 1 shows that the correlation between adjacent pixels of the cipher-image is much smaller than that of plain-image, so we claim that the adjacent pixels of the plain-image are uncorrelated by the proposed cryptosystem effectively from different directions.

In color images there are the high correlation between adjacent pixels of R, G and B components. The proposed cryptosystem encrypt pixels of color components so that make them affect one another. Table 2 and Table 3 show the results of the same position correlations and related adjacent position correlations between R, G and B components of plain-image and cipher-image.

Table 2: Similar position correlations between R, G and B components

Scan direction	R-G	R-B	G-B
Plain-image	0.929848	0.797885	0.949200
Cipher-image	0.000279	0.005105	0.004628

Table 3: Adjacent position correlation between R, G and B components

Scan direction	R-G	R-B	G-B
Plain-image	0.896510	0.756614	0.891265
Cipher-image	0.002288	0.006150	0.001227

4.2 Key Sensitivity Analysis

When one bit of the security key is altered, there are obviously differences between two cipher-images. The number of pixels change rate ($NPCR$) and the unified average changing intensity ($UACI$) for the two encrypted images are applied to measure the number of pixels change rate.

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%$$

$$UACI = \frac{1}{N \times N}[\sum_{i,j} \frac{|C(i,j) - C'(i,j)|}{255}] \times 100\% \quad (19)$$

where N is the height (width) of the encrypted image. We get two encrypted images C and C', whose secret keys are different in only one bit. We also define a two-dimensional array D, which has the same size as C. If $C(i,j) = C'(i,j)$, then $D(i,j) = 0$, otherwise $D(i,j) = 1$. To resist against security key attack, $NPCR$ and $UACI$ values should be large enough for an ideal cipher system. When the secret key is altered from 207 21 42 61 122 203 97 76 101 5 7 241 139 28 98 17 to 208 21 42 61 122 203 97 76 101 5 7 241 139 28 98 17 the differences is made greatly.

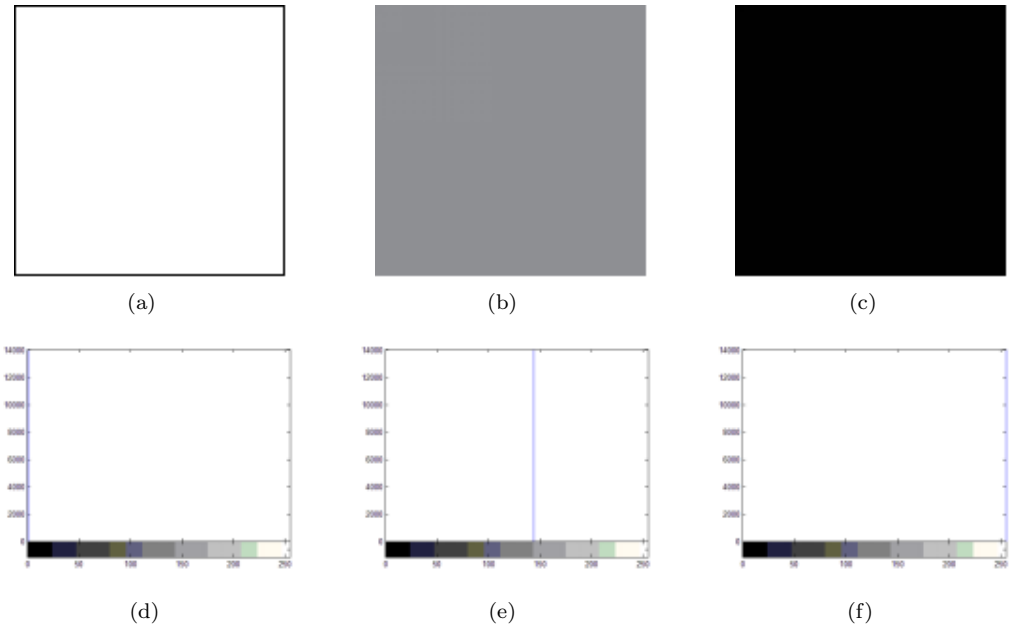


Figure 2: (a) Original white image, (b) the original monolithic gray-level image, (c) the original black image, (d) the histogram of the white image, (e) the histogram of the monolithic gray-level image, (f) the histogram of the black image

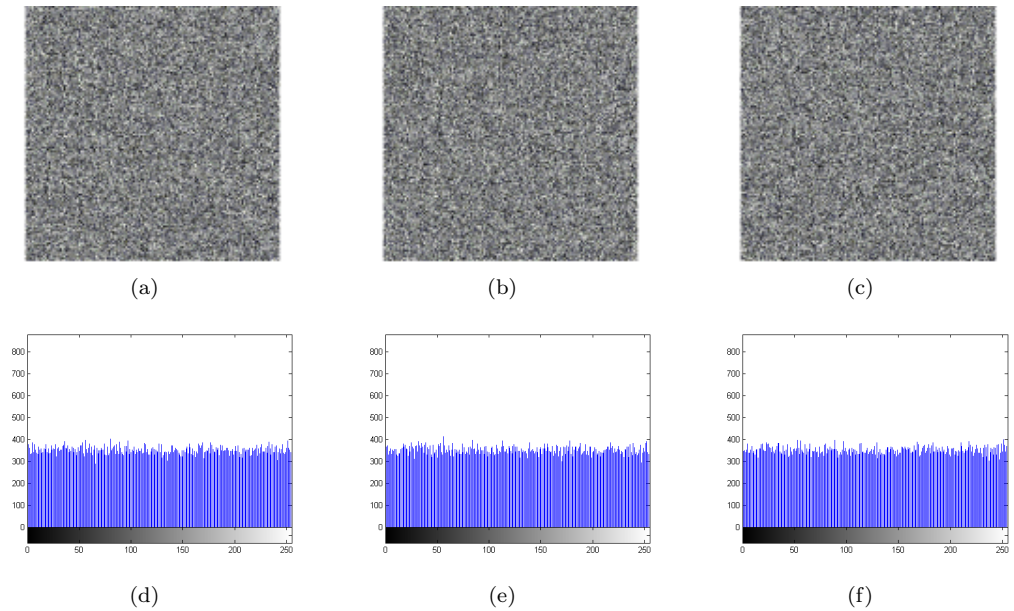
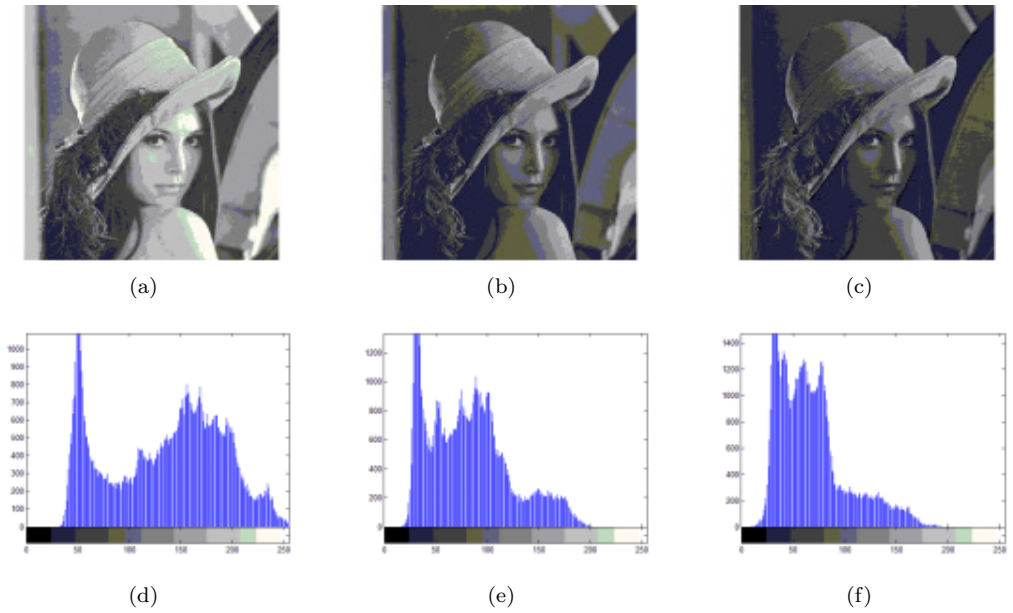


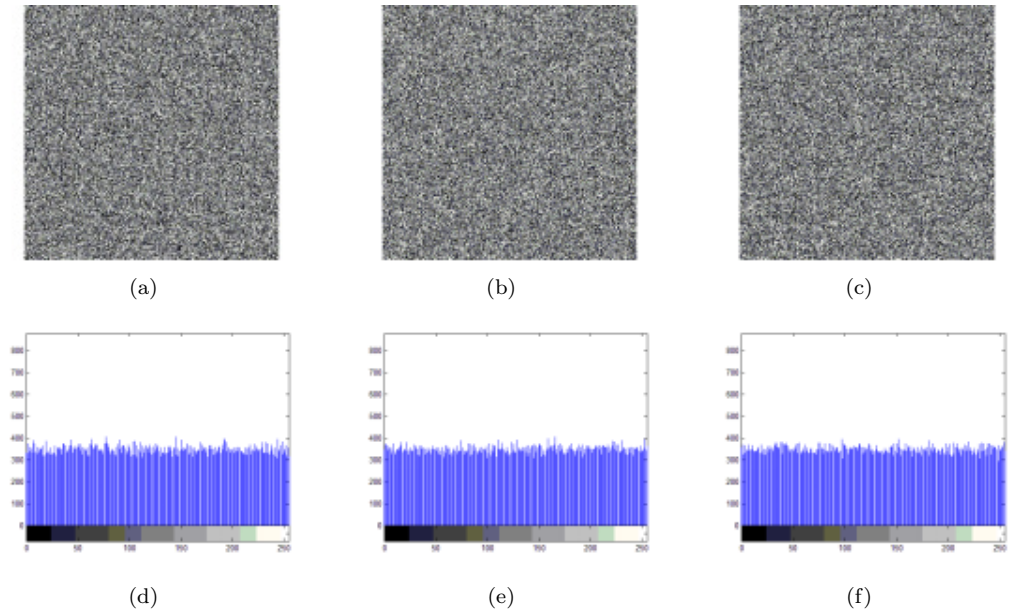
Figure 3: (a) Cipher of white image, (b) the cipher of monolithic gray-level image, (c) the cipher of the black image, (d) the histogram of the encrypted white image, (e) the histogram of the encrypted monolithic gray-level image, (f) the histogram of the encrypted black image



(a) (b) (c)

(d) (e) (f)

Figure 4: (a) Plain-image Lena-R, (b) the plain-image Lena-G, (c) the plain-image Lena-B, (d) the histogram of the plain-image Lena-R, (e) the histogram of the plain-image Lena-G, (f) the histogram of the plain-image Lena-B



(a) (b) (c)

(d) (e) (f)

Figure 5: (a) The encrypted image Lena-R, (b) the encrypted image Lena-G, (c) the encrypted image Lena-B, (d) the histogram of the encrypted image Lena-R, (e) the histogram of the encrypted image Lena-G, (f) the histogram of the encrypted image Lena-B

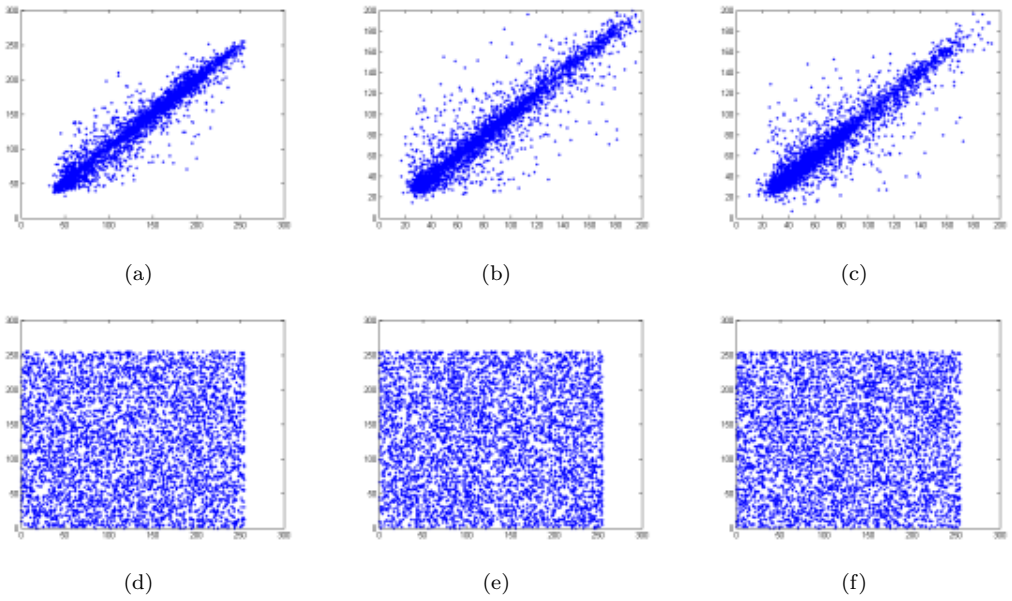


Figure 6: Distribution of two horizontally adjacent pixels in the plain-image of Lena in the (a) red, (b) green and (c) blue components. The distribution of two horizontally adjacent pixels in the cipher-image of Lena in the (d) red, (e) green and (f) blue components. (Color figure online)

Table 4 shows the average $NPCR_{R,G,B}$ and $UACI_{R,G,B}$ values and compares this proposed algorithm with other schemes in terms of the key sensitivity. The proposed algorithm is sensitive dependent on initial conditions and parameters.

Table 4: Comparison of the average $NPCR_{R,G,B}$ and $UACI_{R,G,B}$

Algorithm	Average ($NPCR$)	Average ($UACI$)
Proposed	0.996896	0.334402
[1]	0.000041	0.003320
[6]	0.996355	0.334188
[16]	0.996028	0.334289
[13]	0.000384	0.000433
[28]	0.996358	0.334428
[23]	0.996828	0.334898

4.3 Key Space Analysis

An ideal encryption scheme should have a enough large key space to defend brute-force attack. The size of the key space should be bigger than 2100 to provide a high level of security from the crytography of view [17, 22]. Due to the secret key is 128-bit long, the key space is 2^{128}. We can conclude that the proposed algorithm is large enough to resist all kinds of brute-force attacks.

4.4 Speed Performance

Apart from the security considerations, some other aspects on image cryptosystem algorithm are also important, particularly the running speed for real time Internet multimedia applications. In fact the actual execution time of a cryptosystem depends on many factors, such as CPU structure, OS, memory size, programming skill and so on. We have analyzed the speed of the proposed image encryption technique on an Intel Core I3 CPU 2.3 GHz and 3.99 GB of RAM running on Windows XP and MATLAB 7.1 programming. For accuracy each set of the timing tests was executed several times for considerable number of images and then the average obtained was reported. In Table 5 we can see the comparison results for the proposed scheme and other schemes. Table 5 shows that the proposed algorithm is very fast compared to the other schemes.

Table 5: Comparison of encryption speeds for the proposed scheme and different schemes

Algorithm	Speed (Mbit/s)
Proposed	9.89
[6]	8.11
[16]	5.15
[23]	9.12
[19]	9.39
[18]	8.16
[8]	1.45

5 Conclusions

This paper has realized the quantum image encryption and decryption and protected the information. Image information is ciphered by the proposed encryption algorithm based on general Arnold transform with keys and quantum chaotic map. By improving the Arnold transform algorithm, we not only enlarge the key space to resist against any key sensitivity and any brute-force attack, but also raise the running speed of the process of the encryption. The experiment shows that only one time general Arnold transform with keys has a good result. In order to enhance the sensitivity of the cryptosystem, the generator of the initial conditions and parameters apply the addressing map to get corresponding value. Quantum chaotic map possesses perfect chaotic character, which is used to change the pixel values of the plain-image and eliminate the periodicity generated by the algorithm of general Arnold transform with keys.

The experimental results demonstrate that the proposed method can achieve the high security level to resist various attacks and possesses the high encryption speed (speed $>$ 9.89Mbit/s). Accordingly the proposed algorithm is suitable to practical uses to protect the digital image information over the Internet.

Acknowledgments

This work was supported by the fundamental research funds for the central universities (Grant No. CCNU15GF007).

References

[1] A. Akhshani, A. Akhavan, S. C. Lim, and Z. Hassan, "An image encryption scheme based on quantum logistic map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4653–4661, 2012.

[2] A. Akhshani, A. Akhavan, A. Mobaraki, S. C. Lim, and Z. Hassan, "Pseudo random number generator based on quantum chaotic map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 1, pp. 101–111, 2014.

[3] F. Chen, K. W. Wong, X. Liao, and T. Xiang, "Period distribution of generalized discrete arnold cat map," *Theoretical Computer Science*, vol. 552, no. 4, pp. 13–25, 2014.

[4] L. Chen, D. Zhao, and F. Ge, "Mage encryption based on singular value decomposition and arnold transform in fractional domain," *Optics Communications*, vol. 219, no. 6, pp. 98–103, 2013.

[5] M. Ding and W. Yang, "Stability of synchronous chaos and on-off intermittency in coupled map lattices," *Physical Review E*, vol. 56, no. 4, pp. 4009–4016, 1997.

[6] A. A. A. El-Latif, L. Li, N. Wang, Q. Han, and X. Niu, "A new approach to chaotic image encryption based on quantum chaotic system, exploiting color spaces," *Signal Processing*, vol. 93, no. 11, pp. 2986–3000, 2013.

[7] J. Fridrich, "Symmetric ciphers based on two dimensional chaotic maps," *International Journal of Bifurcation & Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.

[8] T. G. Gao and Z. Q. Chen, "A new image encryption algorithm based on hyper-chaos," *Physics Letters A*, vol. 372, no. 4, pp. 394–400, 2008.

[9] L. M. Jawad and G. Sulong, "Chaotic map-embedded blowfish algorithm for security enhancement of colour image encryption," *Nonlinear Dynamics*, vol. 81, no. 4, pp. 1–15, 2015.

[10] J. Khan, J. Ahmad, and S. O. Hwang, "An efficient image ecryption scheme based on: Henon map, skew tent map and s-box," *International Conference on Modeling, Simulation and Applied Optimization*, vol. 10, pp. 1–6, 2015.

[11] M. Khan, T. Shah, and S. I. Batool, "Texture analysis of chaotic coupled map lattices based image encryption algorithm," *3D Research*, vol. 15, no. 3, pp. 1–5, 2015.

[12] D. L, "Color image encryption algorithm based on chua's circuit and chen's hyper-chaotic system," *Journal of Information & Computational Science*, vol. 12, pp. 1021–1028, 2015.

[13] S. Liu, J. Sun, and Z. Xu, "An improved image encryption algorithm based on chaotic system," *Journal of Computers*, vol. 4, no. 11, pp. 1091–1100, 2009.

[14] M. Machkour, A. Saaidi, and M. L. Benmaati, "A novel image encryption algorithm based on the two-dimensional logistic map and the latin square image cipher," *3D Research*, vol. 6, no. 4, pp. 1–18, 2015.

[15] R. Matthew, "One the derivation of a chaotic encryption algorithm," *Cryptologia*, vol. 8, pp. 29–42, 1989.

[16] S. Mazloom and M. A. Eftekhari, "Color image encryption based on coupled nonlinear chaotic map.," *Chaos Solitons Fractals*, vol. 42, no. 3, pp. 1745–1754, 2009.

[17] B. Norouzi, S. M. Seyedzadeh, S. Mirzakuchaki, and M. R. Mosavi, "A novel image encryption based on row-column, masking and main diffusion processes with hyper chaos," *Multimedia Tools & Applications*, vol. 74, no. 3, pp. 781–811, 2015.

[18] V. Patidar, N. K. Pareek, G. Purohit, and K. K. Sud, "Modified substitution-diffusion image cipher using chaotic standard and logistic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 15, pp. 2755–2765, 2010.

[19] V. Patidar, N. K. Pareek, and K. K. Sud, "A new substitution-diffusion based image cipher using chaotic standard and logistic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 7, pp. 3056–3075, 2009.

[20] P. Ping, Z. J. Wang, and F. Xu, "A two-dimensional cellular automata based method for multiple image," *International Conference on Computer Science & Service System*, vol. 112, pp. 101–104, 2014.

[21] N. S. Raghava, A. Kumar, and A. C. A. Deep, "Improved lsb method for image steganography using henon chaotic map," *Open Journal of Information Security & Applications*, vol. 1, no. 1, pp. 34–42, 2014.

[22] B. Schneier, *Applied Cryptography: Protocol, Algorithms, and Source Code in C*, New York: John Wiley & Sons, 2015.

[23] S. M. Seyedzadeh and S. Mirzakuchaki, "A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map," *Signal Processing*, vol. 92, no. 5, pp. 1202–1215, 2012.

[24] S. M. Seyedzadeh, B. Norouzi, M. R. Mosavi, and S. Mirzakuchaki, "A novel color image encryption algorithm based on spatial permutation and quantum chaotic map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 81, no. 1-2, pp. 1–19, 2015.

[25] X. H. Sun, *Image Encryption Algorithms and Practices with Implementations in C#*, Beijing: Science Press, 2013.

[26] Z. Tang, X. Zhang, and W. Lan, "Efficient image encryption with block shuffling and chaotic map," *Multimedia Tools & Applications*, vol. 74, no. 15, pp. 1–20, 2015.

[27] X. Wang and K. Guo, "A new image alternate encryption algorithm based on chaotic map," *Nonlinear Dynamics*, vol. 76, no. 4, pp. 1943–1950, 2014.

[28] X. Wang, L. Teng, and X. Qin, "A novel color image encryption algorithm based on chaos," *Signal Processing*, vol. 92, no. 4, pp. 1101–1108, 2012.

[29] X. Y. Wang, Y. Q. Zhang, and Y. Y. Zhao, "A novel image encryption scheme based on 2-D logistic map and dna sequence operations," *Nonlinear Dynamics*, vol. 82, no. 3, pp. 1269–1280, 2015.

Hui Liu is a M.S. candidate at the school of computer, Central China Normal University. His research interests include: information security, quantum chaotic.

Cong Jin received the M.S. degrees in applied mathematics from Harbin Institute of Technology, China. She received the Ph.D. in Institute for Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, China. From 1993 to 2003, she was a Lecturer and then become as a full professor at the Hubei University, China. From 2003 to now, she is a full professor of the school of computer, Central China Normal University, China. She has published more than 150 papers on information security, signal processing, and algorithm design and analysis. Her main research interests include computer network security, digital image processing, and software reliability prediction, etc.

Weighted Role Based Data Dependency Approach for Intrusion Detection in Database

Udai Pratap Rao[1], and Nikhil Kumar Singh[2]
(Corresponding author: Udai Pratap Rao)
Department of Computer Engineering, S. V. National Institute of Technology[1]
Surat (Gujarat) 395007, India
Department of Information Technology, UVPCE, Ganpat University[2]
Mehsana, (Gujarat), India
(Email: upr@coed.svnit.ac.in)
(Received Mar. 23, 2016; revised and accepted May 28 & June 10, 2016)

Abstract

In this paper, an approach is ascertained to detect malicious activities in RBAC (Role Based Access Control) enabled database.The proposed approach introduces weighted role based data dependency rule mining algorithm (WRBDDRM), that mines weighted role-wise data dependency rules from database log. The weights are intended to relate the sensitivity of attributes. The proposed algorithm also uses separate support and confidence for each role to generate the dependency rules and these data dependency rules are used to detect the malicious activities in the database. Transactions which disobey any of the data dependency rules are detected as malicious transactions.

Keywords: Database security, insider threats, role based access control, weighted support and confidence, sensitivity of attributes

1 Introduction

Many security steps have already been taken to prevent databases from intrusions [25]. Access control systems, intrusion detection systems, authentication systems, antivirus software and firewalls are few examples of such security measures. In order to safeguard the databases from malicious actions, intrusion detection security measures have been widely considered to detect the malicious actions in the databases. Such kind of security measures are exclusively targeted to database protection and are variant of basic IDS [3]. Intrusion detection is extensively used in different areas for detection of malicious or intrusive activity. The major areas for detection of intrusive activity are computer network [1, 14, 18, 22, 31], database [9, 34, 35], wireless sensor network [11, 23], software code [19], and electric power system [32] etc.. This paper is mainly focused on the detection of malicious activities in databases.

Currently, databases are the central component of many information systems, and therefore represents critical assets to organizations[25]. However,protecting databases from attacks presents unique bottle-neck [8, 9, 21, 39]. Well recognized and accepted security measures, such as anti-virus software, firewalls, access control methods, and file permissions, protect information systems at the network or operating system level but refrains protection against database specific attacks [6, 9, 20, 21, 25]. This happen due to the fact that actions which are malicious for DBMS may not be malicious for operating system and network. Network and operating system defense mechanisms are mainly designed to defend against external attackers [18, 20], but insider attacks are the primary threat in case of transaction-level database [25]. Recent database security research focuses on techniques to mitigate against insider attacks [6, 13, 21, 25]. The most basic level of insider attack is from authorized users, who do not posses database administrator rights [13, 21, 25]. The second category of insider threat is a group of authorized users who do not have database administrator rights, referred to as collaborators [7, 9]. A counter measure framed to protect the database from an aggregation of malicious queries may not detect those same queries if they are originated from collaborating users [9]. The last insider category is database administrators [20].

This paper synthesizes current research on database intrusion detection and proposes a database intrusion detection model that incorporates the key capabilities developed in earlier research [34, 40].

The rest of the paper is framed as follows: Section 2 reflects the related work. Section 3, contains motivation and contribution. Section 4 exhibits the related terms needed to understand the approach, transaction representation and system architecture of our proposed WRBDDRM approach. Section 5 contains learning and detection algorithms of WRBDDRM. Performance results and

analysis are emphasized in Section 6. Finally, we conclude our work in Section 7.

2 Related Work

Database IDS mainly uses one of the two approaches: misuse detection and anomaly detection. The anomaly detection is more effective than misuse detection because it detects known and unknown attacks.

Chung et al. [10] proposed the system called DEMIDS (Detection of Misuse in Database Systems) in 2000. DEMIDS defines notion of distance measure. Pair wise shortest distance and schema distance of set of attributes is measured with the help of integrity constraints. Then frequent item sets are extracted from the database log with the help of distance measure. The frequent item sets are considered as the profiles of users, which can later be used by security officer to verify existing security policies. This approach gives the basic idea of database IDS and does not provide the explored view.

In 2000, Lee et al. [24] contemplated database intrusion detection system for real-time database systems. For detecting intrusions, they exploited real-time properties of data. They keep track of update rates of data objects, which are unknown to the intruder. This approach is exclusively applicable to real time database where time of access is utmost important.

Low et al. [26] suggested a fingerprint (signature) based approach in 2002. In this, the signature of legitimate transactions are stored where new transactions executed by the user are checked against previously stored fingerprints. Signature based approaches are only possible if there are fixed number of applications which can access database. In this approach, the signature in the form of fingerprint introduces the matching overhead and parallelly effects the accuracy.

Hu et al. [13] proposed data mining approach in 2004 to identify intrusion. They used rule based classification for the system. During learning phase frequent sequential patterns are extracted and used for generating classification rules viz. read rules and write rules. These rules are used during detection phase for detecting the intrusion. They focused particularly on malicious modification of data, while malicious read operation are not caught by their proposed approach.

Vieira et al. [43] proposed DBMTD (Data Base Malicious Transaction Detector) mechanism in 2005. The approach is signature based. They have done the manual profiling of transactions, which is not possible if number of valid transactions by applications is too high. Usefulness only for small and fixed number of applications is the major drawback of the approach.

Bertino et al. [8] proposed intrusion detection scheme for RBAC-enabled database in 2005. Naive bayes classifier is used for detecting intrusion in RBAC enabled database. They have proposed three levels of triplets that can be used to transform database log after preprocessing. If role obtained by classification is same as original one which has executed the query then transaction is said to be a legitimate transaction.

Later in 2008, Kamra et al.[21] improved their approach by using quiplets instead of triplets. Drawback of their both approaches is that if database is not RBAC-administered then there is a need to maintain separate profiles for each user, which will add large memory overhead to store the profiles and execution time for classification. The approach is restricted to work at query level and not at transaction level is another major limitation of the work presented.

Srivastava et al. [40] proposed Weighed Data Dependency Rule Miner (WDDRM) algorithm in 2006 and improved the approach [16] by considering the sensitivity of attributes. This helps in extracting sequences with attributes which are sensitive but less accessed.

Mathew et al. [30] proposed data-centric approach to insider attack detection in database systems in 2010. Their experimental result shows that their technique is very effective, accurate, and is promising in complementing existing database security solutions. This approach was the first data-centric approach for detecting database intrusion.

In 2010, Rao et al. [33] improved the approach presented in [8] by extending it at transaction level. They have shown that their approach outperforms compared to query level approach [8]. In 2014, an approach for enhancing the detection rate in database Intrusion Detection System proposed [35]. This novel approach provides the flexibility in profile matching constraints. They are able to enhance the detection rate by reducing the false positive and false negative rate.

In 2015, Rao et al. [34] proposed an RBDDRM (Role based Data Dependency Rule Miner) approach for detection of database privilege abuse in RBAC (Role Based Access Control) administered database. This approach mines role wise data dependencies from database log which are considered as role profiles and are used to detect the privilege abuse by database users. The main focus is on the read, write, and conditional rules to strengthen the approach. The approach proposed in [34] is further improved by considering the sensitivity of attributes.

3 Motivation and Contribution

Database breaches have always been a threat to the privacy of individuals and organizations [17]. According to Verizon data breach investigation report of 2012, out of all attacks involving insider, 90% were malicious and was performed intentionally. It indicates that detection of malicious insider attack in the database is of great concern [42].

Insider attacks have not only grown frequently, but also found significantly more damaging to businesses than external attacks. In 2008, the Identity Theft Resource Center (ITRC) in the United States said that one in six

breaches (7.7%) was attributed to insiders, more than twice of that found in 2007 (16%) [15]. The ITRC 2008 report reached 656 breaches in which 35,691,255 records were exposed by the end of 2008, reflecting an increase of 47% over last year's total of 446. The ITRC 2013 breach report reached 614 breaches, in which 91,982,92 records were exposed by the end of 2014 [16]. In its 2008 Data Breach Investigations Report, based on more than 500 forensic investigations of security breaches, Verizon Business found that half of all internal breaches were conducted by IT administrators [5]. The 2008 CSI Computer Crime and Security Survey [37] reported continuing trends in the frequency and severity of insider abuse and financial fraud. From 2004 to 2011, respondents consistently reported insider abuse as the second most frequent type of security incident [27, 28, 29, 36, 37, 38]. The specific rate of insider abuse incidents ranged from as low as 42% of all reported incidents [29] to as high as 59% [27, 36]. Financial fraud is another form of insider attack and accounted for 8% to 12% of reported incidents between 2004 and 2011 [27, 28, 29, 36, 37, 38]. Insider attacks are not just frequent but expensive too. The two insider related categories of computer security incidents named (i) insider abuse, and (ii) financial fraud account for a major portion of computer security losses [36, 37]. As per the statistics shown by Verizon 2010 data breach investigation report [4], it is clear that some effective method is required to detect the malicious activities in the database.

Rao et al. [34] inspired from the concept of Hu and Panda [13] and proposed role based data mining approach in which rules are generated for each role separately. Conditional rules are generated along with the read and write rules. In this approach no rules will be generated if some attributes are less frequent and more sensitive to be attacked. Using sensitivity of attributes we can improve the performance of the database IDS [34]. We propose the following measures to improve the performance of [34].

1) Proposed approach extracts data access dependencies based on the weighted scheme [40] that exists between the attributes of the database using [12]. In our approach, access dependencies are extracted separately for each role based on the sensitivity of the attribute.

2) We modify the definition of read, write and conditional rules given in [34]. These modified read, write and conditional rules present strict checking of user input transactions for malicious behavior. The definition of read, write, conditional rules in [13, 34] contain only one attribute on LHS. For example:

$$\begin{aligned} w(a) &\rightarrow r(b,c)r(b,d) \\ w(f) &\rightarrow c(k,d). \end{aligned}$$

In our approach, we define the read, write, and conditional rules in which one or more attributes are allowed on LHS. For Example:

$$\begin{aligned} w(a,e,f) &\rightarrow r(b,c)r(b,d) \\ w(e,f) &\rightarrow c(b,c). \end{aligned}$$

3) We use different support and confidence for each role based on the fact that how much dependency rules are required to detect the malicious pattern precisely. Each role in the DBMS is having different access pattern of database. If some role in the system accesses the database more frequently then higher support is required and if accesses of the database are less frequent then lower value of support is needed to get the desirable frequent sequences. Now, with the help of these frequent sequences generated for each role, we use separate confidence for that role, so that generated rules can effectively detect the actual behavior of the user input transaction. If the confidence value is high, then fewer rules will be generated, hence the chances of higher false negative rate. If the confidence value is less then more rules will be generated and the chances of higher false positive rate. So, we have chosen the confidence value separately for each role in such a way that lowers the FP and FN values. To extract such data dependencies, we use database access history which is assumed to be free from attacks. These dependencies are then used to extract dependency rules. These data dependencies which are in the form of access rules (different for each role) are used to detect the database insider abuse.

4 Weighted Role Based Data Dependency Rule Miner

We have used sequential pattern mining algorithm [2] for extracting data dependencies in the database system. Sequential patterns extracted are then converted to data dependency rules. We use the rule-based classification technique for detection of malicious activity. The rules generated at the end of our approach reflect data dependencies among attributes of the database.

Our approach is concerned more about insider attack and detects external attacks as well if an intruder disobeys any data dependency rules. Our work is based on relational database; we call our proposed algorithm as *WRBDDRM (Weighted Role Based Data Dependency Rule Miner).*

4.1 Terminologies

In this section, we explain some of the formal definitions needed to understand the approach.

Read Operation: Read operation on a set of attributes is represented as $r(a_1, a_2, ..., a_n,)$. Read operation r_1, is said to be contained in read operation r_2, if r_2 have all attributes present in r_1 (means r_2 is the subset of

r_1). Read operation r_2 may have extra attributes that are not present in r_1.

Write Operation: Write operation on a set of attributes is represented as $w(a_1, a_2, \cdots, a_n)$. Write operation w_1 is said to be contained in write operation w_2, if w_2 has all attributes present in w_1. Write operation w_2 may have extra attributes that are not present in w_1.

Conditional Operation: Conditional operation on set of attributes means there is condition on those attributes in a query of transaction. For example in a query *"select name from student id=1"*, there is a condition operation on attribute *id*. There may be several attributes in one operation. Conditional operation is represented as $c(a_1, a_2, \ldots a_n)$.

Remark 1. *Attributes within one operation is an unordered set of attributes i.e. their sequence does not matter.*

Sequence: Sequence is an ordered list of one or more read, write or conditional operations. Each read, write or conditional operation in a sequence can have one or more attributes on which operation is believed to be performed simultaneously (sequence of attributes in an operation is irrelevant). Sequence is denoted as

$$< o_1\,(a_{11},\ a_{12}, \ldots\ a_{1w})\,,\ o_2\,(a_{21},\ a_{21}, \ldots\ a_{2x})\,,$$

$$o_3\,(a_{31},\ a_{32}, \ldots\ a_{3y})\,, \ldots\ o_n(a_{n1},\ a_{n2}, \ldots\ a_{nz})\ >$$

Here, o_i denotes read (r), write (w) or conditional (c) operation. a_{kl} denotes the attribute of the relation. A sequence $<\ o_{11},\ o_{12},\ o_{13}, \ldots\ o_{1n}\ >$ is said to be contained in another sequence $<\ o_{21},\ o_{22},\ o_{23}, \ldots\ o_{2m}\ >$, if there exist integers $i_1 < i_2 < i_3 < \ldots$ such that $o_{11} \subseteq o_{2i1}, o_{12} \subseteq o_{2i2}$, $o_{13} \subseteq o_{2i3}, \ldots\ o_{1n} \subseteq o_{2in}$, and also o_{1k} and o_{2ik} must be same operation (read, write or conditional) where $1 <= k <= n$.

Support: The support for a sequence is defined as the fraction of total transactions that contain the sequence. Transaction is also the sequence of operations.

Read Sequence: Read sequence is a sequence with all operations as read operations except last operation which must be write operation; all read, write operations in a read sequence can have several attributes. Read sequence of set of attributes is denoted as

$< r\,(a_{11},\ a_{12}, \ldots a_{1w})\,,\ r\,(a_{21},\ a_{22}, \ldots a_{2x})\,,$

$r\,(a_{31},\ a_{32}, \ldots a_{3y})\,, \ldots w(a_{n1},\ a_{n2} \ldots\ldots\ldots a_{nm})\ >$ which represents that transaction may need to perform all read operations in order before the transaction updates attribute $(a_{n1},\ a_{n2} \ldots\ldots\ldots a_{nm})$.

Read Sequence Set (ReadSeqSet): Read sequence set is the collection of all read sequences.

Read Rule: Read rule is a rule with exactly one write operation on LHS and ordered sequence of read operation(s) on RHS. Read rule is denoted as,

$$w(a_{n1},\ a_{n2} \ldots a_{nm})\ \rightarrow r(a_{11},\ a_{12}, \ldots\ a_{1w}),\ r(a_{21},\ a_{22},$$

$$\ldots a_{2x}), \ldots\ r(a_{(n-1)1},\ a_{(n-1)2}, \ldots\ a_{(n-1)z})\ >$$

Remark 2. *A rule r_1 is said to be contained in rule r_2, if LHS of both the rules is same and operations on RHS of r_1 is subset of operations on RHS of r_2. Also, attributes of all operations on RHS of r1 must be the subset of attributes of corresponding operations on RHS of r_2 [34].*

Write Sequence: Write sequence is a sequence with all operations as write operations; all write operations in a write sequence can have several attributes except first write operation which must have exactly one attribute. Write sequence is denoted as,

$$<w\,(a_{11},\ a_{12}.\ldots.a_{1n})\,,\ w\,(a_{21},\ a_{22}, \ldots a_{2x})\,,$$

$$w\,(a_{31},\ a_{32}, \ldots a_{3x}) \ldots\ w\ (a_{n1},\ a_{n2}, \ldots a_{nz})\ >$$

which shows that transaction may need to perform all write operations in order after the transaction updates attribute $(a_{11},\ a_{12}.\ldots.a_{1n})$.

Write Sequence Set (WriteSeqSet): Write sequence set is collection of all write sequences.

Write Rule: Write rule is a rule with exactly one write operation on LHS and ordered sequence of write operation(s) on RHS. Write rule is denoted as,

$w\,(a_{11},\ a_{12}.\ldots.a_{1n}) \rightarrow\ <w\ (a_{21},\ a_{22}, \ldots a_{2x}),$
$w\ (a_{31},\ a_{32}, \ldots a_{3x}) \ldots w\ (a_{n1},\ a_{n2}, \ldots a_{nz})\ >$

Write rule can be easily generated from write sequence.

Conditional Sequence: Conditional sequence is a sequence of exactly two operations; in which first operation must be conditional operation on one or more attributes and second operation must be read or write operation on one or more attributes. Conditional sequence is denoted as $<$ c $(a_{11},\ a_{12}, \ldots\ a_{1x})$, r/w $(a_{21},\ a_{22} \ldots\ldots a_{2n})\ >$ which represents that transaction may need to perform conditional operation on set of attributes immediately before the transaction read/write attributes $(\ a_{21},\ a_{22} \ldots\ldots a_{2n})$.

Remark 3. *A conditional sequence $<c_{11}, o_{12}>$ (o_{12} here must be read or write operation) is said to be contained in another sequence $<o_{21}, o_{22}, o_{23}, \ldots o_{2m}>$, if there exist integers i, j=i+1 such that $c_{11} \subseteq o_{2i}$ and $o_{12} \subseteq o_{2j}$, and also o_{2i} is conditional operation and o_{2j} is same operation (read, write or conditional) as o_{12}.*

Conditional Sequence Set: Conditional sequence set is collection of all conditional sequences.

Conditional Rule: Conditional rule is a rule with exactly one read/write operation on LHS and exactly one conditional operation on RHS. Conditional rule is denoted as,

$$r/w(a_{21},\ a_{22}\ldots\ldots a_{2n}) -> c(a_{11},\ a_{12},\ldots a_{2x}).$$

Conditional rule can be easily generated from conditional sequence.

Confidence: Confidence of a rule can be defined as the fraction of support of the sequence from which rule is generated to the support of the operation on LHS.

Weighted Support and Confidence [40]: Few attributes in every database that are much important to be sensed or tracked for malicious modifications as compared to the other attributes. More the sensitivity of an attribute is, more is its weight. Srivastava et al.[40] have categorized the attributes in three sets: High Sensitivity (HS), Medium Sensitivity (MS), and Low Sensitivity (LS). The sensitivity of an attribute depends on the database application. From an integrity perspective, in addition, the sensitivity for modifications of attributes are more vital than reading them. Let x be the same attribute and if x ∈ HS then W (x_w) > W (x_r), in which W represents function of weight, x_w indicates writing or altering attribute x and x_r signifies attribute x reading.

We arrange all the attributes into the aforementioned three sets on the basis of their sensitivities and allocate numerical weights to each set; once schema is given for instance, say $w_1,\ w_2,\ w_3 \in \mathbf{R}$, is the real number set and $w_3 = w_2 = w_1$ are the weights of HS, MS and LS, respectively for each category. Lets assume $d_1, d_2, d_3 \in \mathbf{R}$ are the additional weights associated with write operations, such that $d_3 = d_2 = d_1$. Let read operation accesses the attribute x and w_1 denotes the weight associated with x. If write operation also accesses the attribute x then the weight associated to x will be $w_1 + d_1$ and can be represented as,

$$\begin{aligned} W(x_r) &= w_1 \\ W(x_w) &= w_1 + d_1. \end{aligned}$$

Let us assume a sequence s with weight w_s and let N be the total number transactions. If s is present in n transactions out of N transactions, then the support of sequence s can be given as [40],

$$Support\ (s) = (n * w_s)\ /\ N.$$

Suppose R be a rule of read operation having the form $a_{jw} \rightarrow a_{1r}, a_{2r}, \cdots, a_{kr}$ produced from the read sequence $rs \in ReadSeqSet$. Let $Count(a_{jw})$ and $Count(rs)$ be the aggregate count of the attribute a_{jw} and that of rs among the total transactions. The weighted confidence of the rule R is defined as [40],

$$Confidence\ (C_R) = Count\ (rs)\ /\ Count\ (a_{jw}).$$

$Count\ (a_{jw})$ is defined as follows:

$$Count(a_{jw}) = \sum_{\forall Transaction \neq T, a_{jw} \in T and rs T} (w3 + d3) + \sum_{\forall Transaction T,\ rs \in T} max(W(rs)).$$

$Count(rs)$ is defined as,

$$Count(rs) = \sum_{\forall Transaction T, rs \in T} max(W(rs)).$$

Maximum disobeyed confidence: Maximum disobeyed confidence finds the severity of the malicious activity by using the confidence of rules [34]. It uses maximum function to get the highest confidence from confidences of disobeyed rules. The maximum confidence from disobeyed rules shows the severity of the malicious activity. Let there is a rule $\mathbf{r(a)} \rightarrow \mathbf{c(b)}$ with 70% confidence. This means that out of all transactions reading attribute 'a', 70% of transactions have conditional operation on attribute 'b' immediately before reading attribute a. While 30% of transaction that does not support this rule; do not have conditional operation on attribute b immediately before reading attribute a. So, transaction detected as malicious might be from these 30% of transactions which are not malicious. But, if transaction disobeys rule with 100% confidence then there are more chances of that transaction being malicious.

4.2 Transaction Representation

Transactions from the database log (during learning phase) and transactions from users (during the detection phase) need to be preprocessed. After the pre-processing it is represented in the format needed by our approach which is similar to the representation used by [34]. It can be better understood by following example.

Select a, b, c, d
from table_name
where $e =$ "xyz" and $f =$ "abc"

Update table_name
set $a =$ "pqr", $e =$ "xyz"
where $c =$ "abc"

Above transaction of two queries, after pre-processing is represented as,

$$< c(e,\ f),\ r(a,\ b,\ c, d),\ c(c),\ w(a,\ e) >$$

.

Attributes used by WHERE clause are considered as conditional operation on those attributes.

4.3 Proposed System Architecture

The proposed system architecture as given in Figure 1 has two phases: learning phase and detection phase. In the learning phase, the database log is used to mine role profiles where the transactions are converted to a representation as discussed in Section 4.2. After completion of pre-processing, preprocessed transactions are fed to proposed algorithm to extract the role-profiles. These profiles are then stored for later use and represents the normal behavior of the role.

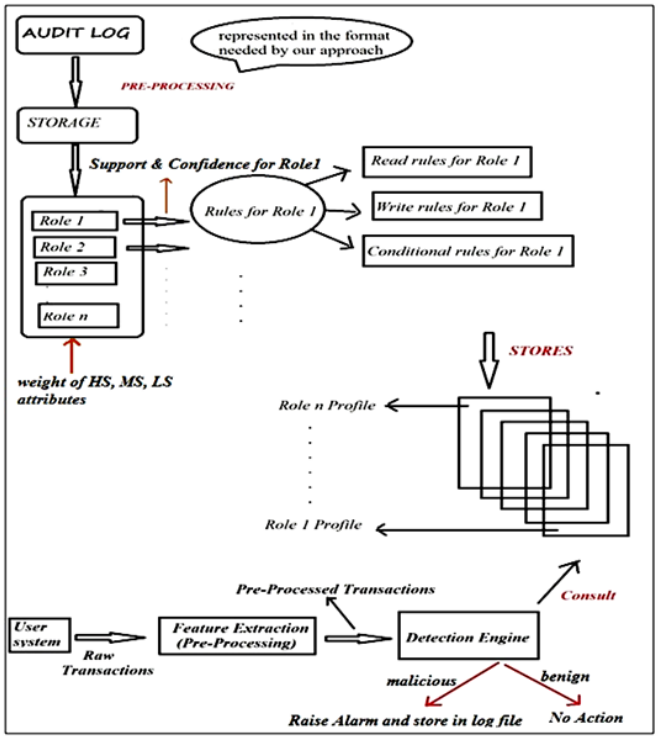


Figure 1: Proposed system architecture

In the detection phase, new user transaction is checked when it is executed. The transaction from the user is preprocessed first and converted to the same presentation used during the learning phase. The preprocessed transaction is given as input to the detection engine which consults the role profiles and checks the transaction against all the rules to the corresponding role of the users. If the transaction disobeys any rule, an alarm is raised (entry in the log of possible abuses). Otherwise, it is considered as normal transaction and no action is taken in such case.

5 The Algorithm

Our proposed approach works in two phases viz. learning phase and detection phase.

5.1 Learning Phase

Preprocessed transactions are fed to the proposed WRBDDRM learning algorithm after feature extraction from the database log. The steps of this algorithm are described in Algorithm 1. At the end of WRBDDRM learning algorithm, role wise profiles are generated and stored on the permanent storage. Terminologies for WRBDDRM learning algorithm are listed in Table 1.

Algorithm 1 WRBDDRM Learning Algorithm

Input: Set of preprocessed transactions from the database log; T
Output: Role wise rules i.e. read, write and conditional rules for each role
Initialize sensitivity group of attributes and for each role k, $1 \leq k \leq n$
Initialize $T_k = \{\ \}$
for each transaction t in T **do**
  Insert t toT_k, where k is role who executed t
end for
for all role k, where $|T_k| > 0$ **do**
  Initialize $RS_k = \{\}$, $WS_k = \{\}$, $CS_k = \{\}$, $RR_k = \{\ \}$, $WR_k = \{\ \}$, $CR_k = \{\ \}$
  Generate sequential pattern $x_i = Weighted\_AprioriAllalgorithm(T_k, min\_support_k)$
end for
for all sequential pattern x_i, where $|x_i|$>1 **do**
  if $|x_i|$=2, If the last operation in x_i is write or read operation and first operation in x_i is conditional **then**
    add x_i to CS_k
  end if
  if last operation in x_i is write and all other operations in x_i are read **then**
    add x_i to RS_k
  end if
  if first operation in x_i is write and all other operations in x_i are also write **then**
    add x_i to WS_k
  end if
end for
for every sequence s in CS_k **do**
  if $weighted\_confidence$ of conditional rule r generated from sequence s >$min\_confidence_k$ **then**
    if If no rule in CR_k (with $weighted\_confidence$ equal to$weighted_confidence$of r) contains r **then**
      Add rule r to CR_k
    end if
    Delete the rules from CR_k which has same $weighted\_confidence$ as r and are contained in r
  end if
end for
for every sequence s in RS_k **do**
  if $weighted_confidence$of read rule r generated from sequence s >$min\_confidence_k$ **then**
    if no rule in RR_k (with $weighted\_confidence$ equal to $weighted_confidence$of r) contains r **then**
      Add rule r to RR_k
    end if
  end if
  Delete the rules from RR_k which has same $weighted\_confidence$ as r and are contained in r
end for
for every sequence s in WS_k **do**
  if $weighted\_confidence$ of write rule r generated from sequence s >$min\_confidence_k$ **then**
    if no rule in WR_k (with $weighted\_confidence$ equal to $weighted\_confidence$ of r) contains r **then**
      Add rule r to WR_k
    end if
    Delete the rules from WR_k which has same $weighted\_confidence$ as r and are contained in r
  end if
  Store RR_k, WR_k, and CR_k to the permanent storage
end for

WRBDDRM learning algorithm generates role wise rules based on the sensitivity of attributes while considering separate support and confidence for each role.

Weighted AproriAll algorithm is used to gener-

Table 1: Terminologies

T	Set of all preprocessed transactions from the database log;
n	Different types of roles in the application;
T_k	Set of preprocessed transactions executed by role k;
$\|T_k\|$	Number of transactions executed by role k;
RS_k	Set of read sequences mined for role k;
WS_k	Set of write sequences mined for role k;
CS_k	Set of conditional sequences mined for role k;
RR_k	Set of read rules generated for role k;
WR_k	Set of write rules generated for role k;
CR_k	Set of conditional rules generated for role k;
X	Sequential patterns;
$min\_support_k$	Minimum support defined for role k;
$min\_confidance_k$	Minimum confidence defined for role k;
x_i	One out of many sequential patterns;
$\|x_i\|$	Length of sequential pattern x_i.

ate sequential patterns for role k with minimum supportmin_$support_k$. The steps of this algorithm are described in Algorithm 2.

Algorithm 2 Weighted AprioriAll algorithm

1: **Input:** Transactions of role k (T_k) and minimum support of role k ($min\_support_k$)
2: **Output:** Sequential pattern of role k having minimum support as $min\_support_k$
3: $\mathbf{L_1}$ = {large 1-sequences}
4: k=2
5: **for** ($\mathbf{L_{k-1}} \neq \emptyset$) **do**
6: $\mathbf{C_k}$ = **C***andidate_***G***en* ($\mathbf{L_{k-1}}$) // candidates generation algorithm given below
7: **end for**
8: **for** each sequence s in the dataset **do**
9: **for** each candidate in $\mathbf{C_k}$ **do**
10: $\mathbf{n_{c_k}}$= Increment the count that is contained in s
11: $\mathbf{Wc_k}$ = weight of each candidate by using the concept in equation 1 and 2
12: **C***andidate_***S***upport* = weighted support of candidate with weight $\mathbf{Wc_k}$ and $\mathbf{n_{c_k}}$
13: $\mathbf{L_k}$ = Candidates in $\mathbf{C_k}$ with **C***andidate_***S***upport*>**m***in_support*
14: **end for**
15: **end for**

Candidate generation algorithm is used in step 6 of Weighted AprioriAll algorithm for generating new candidates. The steps of candidate generation algorithm are described in Algorithm 3.

5.2 Detection Phase

In the detection phase, new user transaction is preprocessed and fed to detection engine. The detection engine reads the stored rules (outcome of learning phase) and

Algorithm 3 Candidate Generation algorithm

1: **Input:** Set of all large (k-1)-Sequences i.e. L_{k-1}
2: **Output:** Set of all candidate k-Sequences i.e. C_k
3: Insert into C_k
4: Select $p.litemset_1$, ..., $p.litemset_{k-1}$, $q.litemset_{k-1}$
5: From $L_{k-1}p$, $L_{k-1}q$**4.** Where $p.litemset_1 = q.litemset_1$.....

$$p.litemset_{k-2} = p.litemset_{k-2};$$

6: Delete all sequences C_k such that some (k-1) subsequences of c is not in L_{k-1}

checks new user transaction against dependency rules of related role (role of the user who has executed the transaction). If the transaction is compliant with the rules then it is normal otherwise malicious and an alarm is raised. Raising of alarm means an entry is made to log of probable attacks. The steps of detection phase algorithm are described in Algorithm 4.

5.3 Methodology of WRBDDRM Algorithm

Learning phase and detection phase can be best understood by an example. Table 2 and Table 3 show the preprocessed transactions from the database log which are executed by Role 1 and Role 2. We have considered *(a, e, s, l, r, m)* as high sensitive attributes, *(d, j, h, i)* as medium sensitive attribute, and *(b, c, f, g, h, i, m, o, p, q)* as low sensitive attributes. Weight for high sensitive, medium sensitive, and low sensitive attributes are considered as 3, 2, and 1 respectively. Rules for Role 1 is generated by considering support 40% and confidence as 75 % and rules for Role 2 are generated by considering support 45% and confidence as 70%.

Table 4 shows the strong association read, write and

Algorithm 4 WRBDDRM Detection Algorithm

Input: Preprocessed user input transactions
Output: Malicious or normal transaction
for For every role k **do**
  Initialize RR_k={}, WR_k={}, CR_k={}
  Retrieve RR_k, WR_k, CR_k from permanent storage to memory
end for
for each read operation in t **do**
  for every attribute a of the read operation **do**
    for every rule r for attribute a in CR_k **do**
      if r is disobeyed and $max\_disobeyed\_confidence < weight\_confidence(r)$ **then**
        $max\_disobeyed\_confidence = weight\_confidence(r)$
      end if
    end for
  end for
end for
for each write operation in t **do**
  for For every attribute 'a' of write operation **do**
    if If r is disobeyed && $max\_disobeyed\_confidence < weigh\_cofidence(r)$ **then**
      $max\_disobeyed\_confidence = weight\_confidence(r)$
    end if
    if $max\_disobeyed\_confidence \neq 0$ **then**
      Add the entry to log of possible attacks along with $max\_disobeyed\_confidene$
    end if
  end for
end for

Table 2: Example transactions for role 1

T1	<[e,a] r[a,b] c[a] w[d,c] r[e,c] w[d,a]>
T2	<[a,d] w[a,b,c]>
T3	<[d,a] w[e,c] w[a]>
T4	<[e,a] r[b] c[d] w[b,c] r[d,e,c] w[d,a]>
T5	<[d,e,a] r[e,b] c[b,c] w[e,c]>

Table 3: Example transactions for role 2

T1	<[g,h] r[p] w[r] c[q] r[g,i,m] r[f,h,k]>
T2	<[k] r[g,l] c[i,m] r[g,h,l] w[n]>
T3	<[h] r[f,h,k] w[f,n] r[g] c[i] w[l]>
T4	<[p] r[p] w[q] r[g] c[h,k] w[i] r[k] r[f,h]>
T5	<[g,i,m] r[g] c[i,q] w[i] r[g] c[i,k] w[l] r[g,l,>

conditional rules generated for Role 1 and Role 2. Number of strong association conditional rules are 21, number of strong association read rules are 12, and the number of strong association write rules are 4.

Now with the help of rules as shown in Table 4, new input transactions can be classified as normal or malicious. Let, new user transaction is $<c\,[d,a]\,w\,[e,c]\,w\,[a]>$. As

Table 4: Generated rule set

Rule ID	Confidence	RuleSet
Conditional RuleSet (CR_k)		
R1	100%	r[(e b)] => c[(d e a)]
R2	75%	r[(b)] => c[(d e a)]
R3	100%	w[(e)] => c[(d a)]
R4	100%	w[(e c)] => c[(d a)]
R5	100%	r[(a b)] => c[(e a)]
R6	100%	r[(e)] => c[(e a)]
R7	88%	w[(c)] => c[(a)]
R8	100%	r[(f)] => c[(h)]
R9	100%	r[(f h)] => c[(h)]
R10	100%	r[(f h k)] => c[(h)]
R11	100%	r[(f k)] => c[(h)]
R12	75%	r[(h)] => c[(h)]
R13	100%	r[(h k)] => c[(h)]
R14	100%	r[(k)] => c[(h)]
R15	100%	w[(l)] => c[(i)]
R16	100%	r[(g h l)] => c[(i m)]
R17	100%	r[(g l)] => c[(i m)]
R18	100%	r[(h l)] => c[(i m)]
R19	100%	r[(l)] => c[(i m)]
R20	100%	r[(g l)] => c[(k)]
R21	100%	r[(l)] => c[(k)]
Read RuleSet (RR_k)		
R22	100%	w[(a b)] => r[(a d)]
R23	100%	w[(a b c)] => r[(a d)]
R24	100%	w[(a c)] => r[(a d)]
R25	75%	w[(b)] => r[(a d)]
R26	100%	w[(d c)] => r[(a b)]
R27	100%	w[(d)] => r[(b)] r[(e c)]
R28	100%	w[(d a)] => r[(b)] r[(e c)]
R29	100%	w[(f n)] => r[(f h k)]
R30	74%	w[(n)] => r[(f h k)]
R31	100%	w[(i)] => r[(g)]
R32	100%	w[(l)] => r[(g)]
R33	77%	w[(n)] => r[(g l)] r[(g h l)]
R34	100%	w[(a b)] => r[(a d)]
Write RuleSet (WR_k)		
R35	100%	w[(d c)] => w[(d a)]
R36	76%	w[(b)] => w[(d a)]
R37	75%	w[(c)] => w[(a)]
R38	100%	w[(f n)] => w[(l)]

the first operation is conditional operation in the transaction so there are no rules for it. Second operation is write on attributes e and c. One conditional rule R4 is present for write operation on e and c together and one conditional rule R3 is present for the write operation on e. For the given transaction, both the rules R4 and R3 are obeyed. Moving further, third operation is the write on attribute 'a' for which rule set contains no rule. As the transaction $<c\,[d,a]\,w\,[e,c]\,w\,[a]>$ obeys all the rules, therefore, it is considered as normal transaction.

Now, consider a new transaction <c(a), w(c), w(b)>. In this first operation is conditional operation on attribute 'a', therefore no rule on conditional operation. Second operation is the write operation on attribute 'c'. R7 and R37 are rules of write operation on the attribute 'c'. New transaction follows rule R7. But rule R37 is not followed, so new transaction will be detected as malicious and max-disobeyed-confidence is equal to confidence of R37 which is equal to 75%. Similarly, on attributes 'b' last operation is the write operation . R2, R25, and R36 are write rules on attribute 'b'. New transaction does not follow the rule R2, which has confidence equal to 75% that is equal to max-disobeyed-confidence. So, max-disobeyed-confidence will remain same i.e. 75%. Now rule R25 is also not followed by new transaction which has confidence equal to 75% and is equal to max-disobeyed-confidence. The max-disobeyed-confidence will remain same i.e. 75%. Now rule R36 is also not followed by new transaction which has confidence equal to 76% and greater than max-disobeyed-confidence. The max-disobeyed-confidence will be updated from 75% to 76%.

6 Performance Results and Analysis

In this section, the comparison between our approach (WRBDDRM) and existing approach (RBDDRM) [34] is presented.

6.1 Experimental Setup

We use Java programming language for implementation and testing of our approach using Net Beans IDE 7.4. We use TPC-C (online transaction processing benchmark) [41] database schema. We have considered only two roles in the system viz. customer and administrator which are represented by Role1 and Role2 in our implementation.

The synthetic dataset is used to evaluate the performance. We manually generated 40 genuine transactions executed by different roles. Synthetic dataset is generated from 40 genuine transactions which are populated to 600 genuine transactions randomly. While randomly populating transactions, we consider the frequency with which both the roles interact to the database. In our system, Role 1 is the customer who interacts with database more frequently than the Role 2; an administrator. So while populating, frequency of execution of any transaction by Role 1 is more than the frequency of execution of any transaction by Role 2. Attacks are generated by randomly changing the some attributes in the operations of benign transaction by another operation of the same relation (same relation of schema). In this way, 100 malicious transactions are generated. Results are taken on 100 benign and 100 malicious transactions to evaluate false negatives, true negative, true positives, false positive and recall value.

6.2 Performance Results

The statistical results of the metrics: true negative, false positive, false negative, true positive and recall for RBDDRM [34] and WRBDDRM are shown in Table 5 and Table 6 respectively. We have shown here five instances of both the approaches to make the analysis clear. In RBDDRM [34] and WRBDDRM, we have varied minimum support value from 20% to 45% and minimum confidence value from 60 to 75. In proposed WRBDDRM, we vary the sensitivity of attributes i.e. HS, MS, and LS in the range of 1 to 4 to get the results. Instances for RBDDRM [34] are represented as Id# and instances for WRBDDRM is represented as WId# in Table 5 and Table 6 respectively. Role 1 and Role 2 are represented as R1 and R2 in Tables 5 and 6 respectively.

The values of True Negative, False Positive, False Negatives and True Positives are in terms of percentage. Recall Value is shown on the scale of 0 to 1.

6.3 Analysis

As there is no common fix parameters in both the approaches; RBDDRM [34] and WRBDDRM, therefore we can not compare them directly. In RBDDRM [34] support and confidence for all the roles are same while in our proposed approach, for each role support and confidence are different. WRBDDRM also uses different sensitivity parameters for the attributes. We have taken different instances of both the approaches on the basis of their respective parameter values for comparison.

Both RBDDRM [34] and WRBDDRM are compared by taking instances of respective approaches, i.e. Id# and WId# on X-axis and performance evaluation metrics on the Y-axis. The same is presented in Figures 2, 3, 4, 5, and 6 respectively.

From the graphs, we can see that Id1 is an instance of RBDDRM [34] in which support and confidence are 30 and 60 respectively. WId1 is an instance of the WRBDDRM in which support, confidence for Role 1 and Role 2 are 30, 60 and 35, 65 respectively. Weights for HS, MS, and LS attributes are 2, 1.5, and 1 respectively. Here, we see that support and confidence of Role 1 is same in both the approaches and support and confidence for Role 2 in our approach is more than RBDDRM [34]. Due to this, less number of rules are generated compared to RBDDRM [34]. Weights are associated with the attributes

Table 5: Results of RBDDRM algorithm [34]

Instances of RBDDRM	Minimum Support (%) R1&R2	Minimum Confidence(%) R1&R2	TN(%)	FP(%)	FN(%)	TP(%)	Recall
Id1	30	60	45	55	47	53	.53
Id2	35	60	55	45	47	53	.53
Id10	25	75	73	27	53	47	.47
Id11	25	60	55	45	38	62	.62
Id13	45	70	82	18	67	33	.33

Table 6: Results of proposed WRBDDRM algorithm

Instances of WRBDDRM	Minimum Support (%)		Minimum Confidence(%)		Weight of Attributes			TN (%)	FP (%)	FN (%)	TP (%)	Recall
	R1	R2	R1	R2	HS	MS	LS					
WId1	30	35	60	65	2	1.5	1	55	45	41	59	.59
WId2	30	35	60	65	2.5	2	1	43	57	38	62	.62
WId10	25	20	60	55	2	1.5	1	34	66	7	93	.93
WId11	25	20	55	55	2	1.5	1	38	62	10	90	.90
WId13	45	45	70	70	2	1.5	1	55	45	44	56	.56

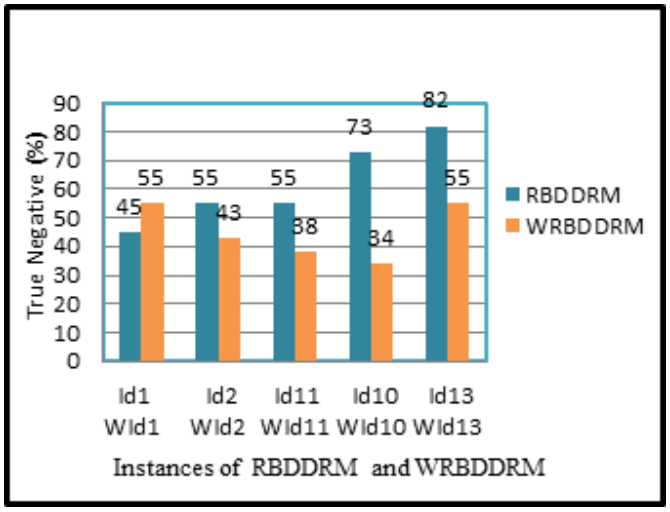


Figure 2: TN Vs various instances of both the approaches

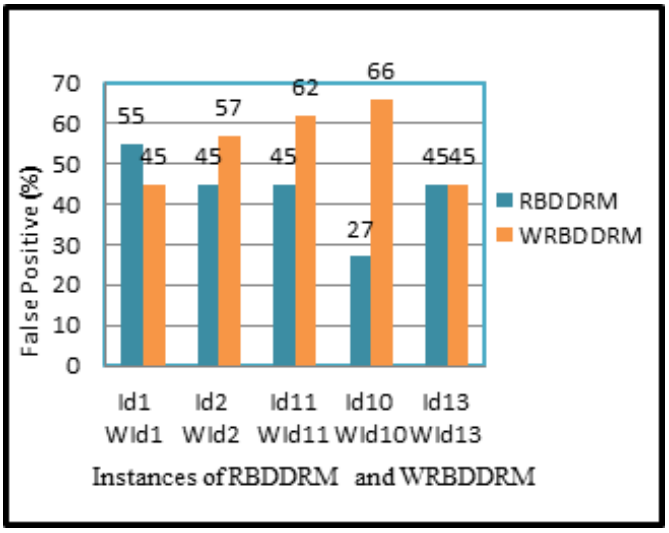


Figure 3: FP Vs various instances of both the approaches

on the basic of their sensitivity and relatively more rules will be generated that too results in increased recall value. Similarly, we can compare for other instances of both the approaches for any performance evaluation metrics used.

Figure 2 and Figure 3, shows the comparison of RBDDRM approach [34] and WRBDDRM approach for true negative and false positive respectively. Figure 4 and Figure 5 show the comparison of false negatives and true positives of both the approaches.

It is also observed from the graph as shown in Figure 3 that false positives of our approach are more than the existing RBDDRM [34] approach; but there is a significant improvement in case of false negatives as shown in Figure 4. False negatives in WRBDDRM are lesser and true positive rate is higher than RBDDRM [34]. Due to the lower false negative rate and higher false positive rate, attack detection capability of WRBDDRM is reasonably high. Comparisons for attack detection capabilities i.e. recall value of our approach and RBDDRM [34] is shown in Figure 6. Improvement in case of false positives for our approach is not adequate because of generating more conditional rules due to the consideration of the attributes' sensitivity. This can be reduced by considering the separate support and confidence for conditional rules.

From the graph as shown in Figure 6, we can say that, for every similar type of instance in both the approaches, recall value of WRBDDRM is higher than the RBDDRM approach [34].

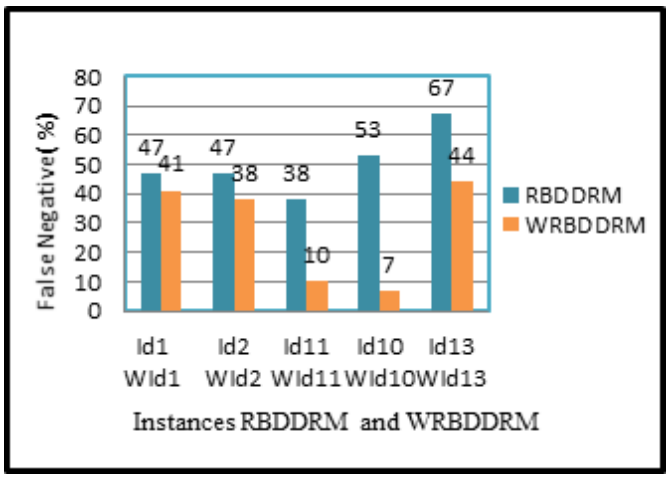


Figure 4: FN Vs various instances of both the approaches

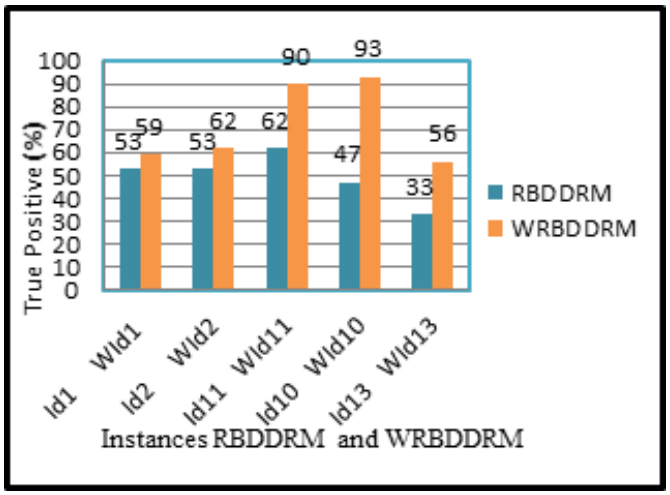


Figure 5: TP Vs various instances of both the approaches

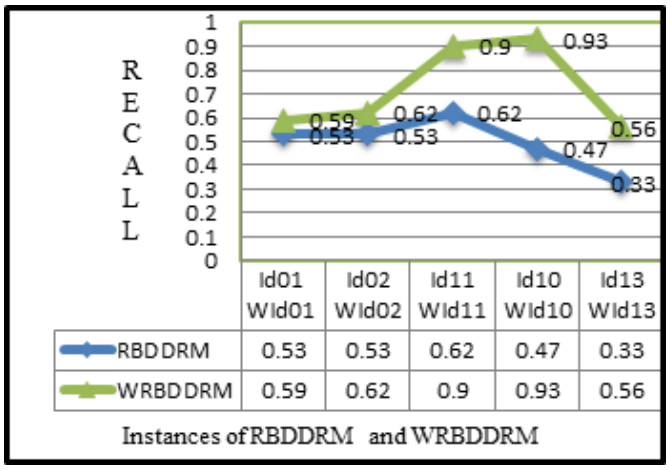


Figure 6: Recall Vs various instances of both the approaches

7 Conclusion

The proposed approach incorporated weighted data dependency rules to strengthen the security of database IDS. By analyzing the experimental results, it is clearly observed that our approach WRBDDRM outperforms in terms of attack detection capability i.e. recall value compared to RBDDRM algorithm.

Acknowledgments

This research work is supported by Institute Research Grant (Ref. No.: Dean (R&C)/1503/2013-14, dated: 17-02-2014) of S. V. National Institute of Technology Surat (Gujarat) 395007-India.

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] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh IEEE International Conference on Data Engineering*, pp. 3–14, 1995.

[3] R. Bace and P. Mell, "NIST special publication on intrusion detection systems," *Technical Report, DTIC Document*, 2001.

[4] W. Baker, M. Goudie, A. Hutton, C. D. Hylender, C. Novak, D. Ostertag, C. Porter, and M Rosen, "2010 data breach investigations report," *Verizon RISK Team*, 2010.

[5] T. Barbaro and M. Andzeller, *2008 Data Breach Investigations Report*, 2008. (`http://www.verizonenterprise.com/resources/security/databreachreport.pdf`)

[6] E. Bertino, S. Jajodia, and P. Samarati, "Database security: research and practice," *Information Systems*, vol. 20, no. 7, pp. 537–556, 1995.

[7] E. Bertino, T. Leggieri, and E. Terzi, "Securing dbms: characterizing and detecting query floods," in *Proceedings of the 2004 Information Security Conference*, pp. 195–206, Springer, 2004.

[8] E. Bertino, E. Terzi, A. Kamra, and A. Vakali, "Intrusion detection in RBAC-administered databases," in *Proceedings of 21st IEEE Annual Computer Security Applications Conference*, pp. 1–10, 2005.

[9] A. S. Chikhale, and S. S. Dhande, "Protection of data base security via collaborative inference detection," *International Journal of Advanced Research*, vol. 3, no. 2, pp. 665–670, 2015.

[10] C. Y. Chung, M. Gertz, and K. Levitt, "Demids: A misuse detection system for database systems," in *Integrity and Internal Control in Information Systems*, pp. 159–178. Springer, 2000.

[11] L. Coppolino, S. D. Antonio, A. Garofalo, and L. Romano, "Applying data mining techniques to intrusion detection in wireless sensor networks," in *Proceedings of IEEE 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 247–254, 2013.

[12] S. Hashemi, Y. Yang, D. Zabihzadeh, and M. Kangavari, "Detecting intrusion transactions in databases using data item dependencies and anomaly analysis," *Expert Systems*, vol. 25, no. 5, pp. 460–473, 2008.

[13] Y. Hu and B. Panda, "A data mining approach for database intrusion detection," in *Proceedings of the 2004 ACM Symposium on Applied Computing*, pp. 711–716, 2004.

[14] N. Hubballi and V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: A survey," *Computer Communications*, vol. 49, pp. 1–17, 2014.

[15] Identity Theft Resourse Center, *2008 Data Breach Insider Theft Category Summary,* 2008. (`http://www.idtheftcenter.org`)

[16] Identity Theft Resourse Center, *2013 Data Breach Insider Theft Category Summary,* 2013. (`http://www.idtheftcenter.org/images/breach/Insider_Theft_Summary_2013.pdf`)

[17] Info Security, *22 Million User IDS May Have Been Stolen From Yahoo Japan,* 2013. (`http://www.infosecurity-magazine.com/view/32498/22-million-user-ids-may-have-been-stolen-//from-yahoo-japanutm_medium=twitterutm_source=twitterfeed`)

[18] J. Jabez and B. Muthukumar, "Intrusion detection system (ids): Anomaly detection using outlier detection approach," *Procedia Computer Science*, vol. 48, pp. 338–346, 2015.

[19] H. Jelodar, J. Aramideh, "Common techniques and tools for the analysis of open source software in order to detect code clones: A study," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 64–69, 2014.

[20] A. Kamra and E. Bertino, "Design and implementation of an intrusion response system for relational databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 875–888, 2011.

[21] A. Kamra, E. Terzi, and E. Bertino, "Detecting anomalous access patterns in relational databases," *The VLDB Journal*, vol. 17, no. 5, pp. 1063–1077, 2008.

[22] D. Kumar and N. Kumar, "An approach for collaborative decision in distributed intrusion detection system," *International Journal of Computer Applications*, vol. 133, no. 13, pp. 8–14, 2016.

[23] M. Kumar, K. Dutta, I. Chopra, "Impact of wormhole attack on data aggregation in hieracrchical WSN," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 70–77, 2014.

[24] V. Lee, J. Stankovic, and S. H. Son, "Intrusion detection in real-time database systems via time signatures," in *Proceedings of 6th IEEE Real-Time Technology and Applications Symposium*, pp. 124–133, 2000.

[25] P. Liu, "Architectures for intrusion tolerant database systems," in *Proceedings of IEEE 18th Annual Computer Security Applications Conference*, pp. 311–320, 2002.

[26] W. L. Low, J. Lee, and P. Teoh, "Didafit: Detecting intrusions in databases through fingerprinting transactions," in *Proceedings of 4th International Conference on Enterprise Information Systems* , pp. 121–128, 2002.

[27] W. Lucyshyn, L. A. Gordon, M. P. Loeb, and R. Richardson, "2004 CSI/FBI computer crime and security survey," *Computer Security Institute*, 2004.

[28] W. Lucyshyn, L. A. Gordon, M. P. Loeb, and R. Richardson, "2005 CSI/FBI computer crime and security survey," *Computer Security Institute*, 2005.

[29] W. Lucyshyn, L. A. Gordon, M. P. Loeb, and R. Richardson, "2006 CSI/FBI computer crime and security survey," *Computer Security Institute*, 2006.

[30] S. Mathew, M. Petropoulos, H. Q. Ngo, and S. Upadhyaya, "A data-centric approach to insider attack detection in database systems," in *Proceedings 13th International Symposium on Recent Advances in Intrusion Detection*, pp. 382–401, Springer, 2010.

[31] G. V. Nadiammai and M. Hemalatha, "Effective approach toward intrusion detection system using data mining techniques," *Egyptian Informatics Journal*, vol. 15, no. 1, pp. 37–50, 2014.

[32] S. Pan, T. H. Morris, and U. Adhikari, "A specification-based intrusion detection framework for cyber-physical environment in electric power system.," *International Journal of Network Security*, vol. 17, no. 2, pp. 174–188, 2015.

[33] U. P. Rao, G. J. Sahani, and D. R. Patel, "Machine learning proposed approach for detecting database intrusions in rbac enabled databases," in *Proceedings of IEEE International Conference on Computing Communication and Networking Technologies (ICCCNT'10)*, pp. 1–4, 2010.

[34] U. P. Rao and N. K. Singh, "Detection of privilege abuse in RBAC administered database," in *Intelligent Systems in Science and Information*, pp. 57–76, Springer, 2015.

[35] U. P. Rao, N. K. Singh, A. R. Amin, and K. Sahu, "Enhancing detection rate in database intrusion detection system," in *Proceedings of IEEE Science and Information Conference (SAI'14)*, pp. 556–563, 2014.

[36] R. Richardson, "The 12th annual computer crime and security survey," *Computer Security Institute*, pp. 1–30, 2007.

[37] R. Richardson, "The 13th Csi computer crime and security survey," *Computer Security Institute*, pp. 1–30, 2008.

[38] R. Richardson, "15th annual 2010/2011 computer crime and security survey," *Computer Security Institute*, pp. 1–44, 2011.

[39] A. C. Squicciarini, I. Paloscia, and E. Bertino, "Protecting databases from query flood attacks," in *Proceedings of IEEE 24th International Conference on Data Engineering*, pp. 1358–1360, 2008.

[40] A. Srivastava, S. Sural, and A. K. Majumdar, "Weighted intra-transactional rule mining for database intrusion detection," in *Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp. 611–620, Berlin, Heidelberg, 2006.

[41] TPC, *TPC Benchmark C, Standard Specification, Ver. 5.1*, July 11, 2016. (`http://www.tpc.org/tpcc`)

[42] Verizon, *Data Breach Investigations Report,* 2012. (`http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2012-ebk_en_xg.pdf`)

[43] M. Vieira and H. Madeira, "Detection of malicious transactions in dbms," in *Proceedings of IEEE 11th Pacific Rim International Symposium on Dependable Computing*, pp. 1–8, 2005.

Udai Pratap Rao is currently an Assistant Professor at the Department of Computer Engineering, S.V. National Institute of Technology Surat, Gujarat, India. He obtained his Ph.D. degree in Computer Engineering in 2014. His research interests include data mining, database security, information security & privacy, and big data analytics.

Nikhil Kumar Singh obtained his master's degree in Computer Engineering from S. V. National Institute of Technology Surat, Gujarat, India. He has completed his B.Tech in Computer Science and Engineering from Institute of Engineering and Rural Technology, Allahabd, India. Nikhil Kumar Singh is currently an assistant professor at U. V. Patel College of Engineering, Ganpat University, India. He has research interests in computer networks, data security, web mining and network security.

A Novel and Concise Multi-receiver Protocol Based on Chaotic Maps with Privacy Protection

Yang Sun[1], Hongfeng Zhu[2], and Xueshuai Feng[3]
(Corresponding author: Hongfeng Zhu)

Software College, Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang 110034, P.R. China
(Email:17247613@qq.com[1]; zhuhongfeng1978@163.com[2]; 1275064307@qq.com[3])
(Received Sept. 8, 2015; revised and accepted Feb. 10 & Mar. 9, 2016)

Abstract

Multi-receiver encryption is an essential cryptography paradigm, which can transmit one message securely among the users by the to form over an insecure network. In this paper, we propose a novel Multi-Receiver scheme using Chaotic Maps, named MRCM, aiming to require one ciphertext with non-interactive process for achieve authentication and the message transmission secretly. Our scheme eliminates the private key generators (PKG) in one domain or multi-domain, in other words, our scheme will be highly decentralized and aim to capture distributed. Our goals are to minimize the hazards of single-point of security, single-point of efficiency and single-point of failure about the PKG. Next, our scheme is based on chaotic maps, which is a high efficient cryptosystem and is firstly used to construct multi-receiver public key encryption. Furthermore, unlike bilinear pairs cryptosystem that need many redundant algorithms to get anonymity, while our scheme can acquire privacy protection easily. Moreover, a novel idea of our MRCM scheme is to adopt chaotic maps for mutual authentication and privacy protection, not to encrypt/decrypt messages transferred between the sender and the receivers, which can make our proposed scheme much more efficient. Finally, we give the formal security proof about our scheme in the standard model and efficiency comparison with recently related works.

Keywords: Ban logic, chaotic maps, multi-receiver, privacy protection

1 Introduction

Multi-receiver encryption is an essential cryptography paradigm, which enables flexible, on-demand, and low computing to transmit one message securely over an insecure network, especially for wire/wireless communications. In 2000, Bellare et al. [1] first proposed the scheme of the multi-receiver in public key encryption. Since then, the growing number of researchers started pay attention to this field, a significant proportion of the protocols have been proposed in various areas, aiming at improving properties and narrowing calculation expense. Generally, in a multi-receiver public key encryption scheme, all users share the common public key encryption system to implement messages sending and receiving. Let us suppose that there are $n+1$ users in the system, including receivers indexed by $1, ..., n$, indicating each receiver have a pair (pk_i, sk_i) as their public and private key for $i = 1, ..., n$ respectively.

If a sender wants to send a message $M_i (i = 1, \cdots, n)$ to n receivers, a sender has to employ all receivers public key to encrypt message, afterwards sends the ciphertexts $(E_i, \cdots, E_n)$ to the common channel. According to the ciphertexts, every receiver picks out respective message and decrypts it by its private key sk_i to catch information. It is worth noting that in this encryption system, the sender and receiver are not invariable, it means each user can become a sender at this moment may also turn to a receiver next time. But we always in a definite model of 1-to-n (one sender-to-n receivers) and single-message $(M_1 = ... = M_i ... = M_n)$ encryption communications. This setting of public key encryption is called as 1-to-n multi-receiver public key encryption system in the following documents [7, 14, 21]. Such as the signcryption mechanism proposed by Sun and Li [22] in 2010, its protocol requires only one or none pairing computation to signcrypt a message for multiple receivers instead of computing bilinear paring repeatedly.

It is generally known that the network platform is insecure for us to communicate, so many researchers put emphasis on keep anonymity [4, 18, 25, 30]. Meanwhile in the field of multiple receivers, researchers also pursue identity privacy protection. In 2013, Wang [24] proposed an anonymous multi-receiver remote data retrieval model for pay-TV in public clouds, which can withstand malicious corporation and consumer. In the same year, Pang et al.

present a novel multi-recipient signcryption scheme [16] with complete anonymity that can achieve both the signer and the receiver anonymity. Motivated by the notion of multi-receiver [1] and identity-based which was presented by Shamir [20], Baek et al. [1] proposed a new multi-receiver identity-based encryption (MR-IBE) scheme in 2005.

In this protocol, a sender encrypt a message to receivers with each identifier information instead of the public key, then each receiver decrypt this message by his private key, which connected with their ID. And different with the protocol of [5], this scheme only needs one or none pairing computation, it is greatly shorten the calculation time. There is no denying the fact that this new model opens a new road for the network security management. Based on this protocol, Fan et al. [8] proposed an anonymous multi-receiver identity-based encryption scheme, it illustrated that the identity of any receiver can be concealed to anyone else. However, in the following years, the researchers conducted a series of improvement [13, 26, 32] to solve this anonymity problem. In the year of 2011, Qin et al. [17] introduced a threshold signcryption scheme, which can solve the problem of single-point failure among a number of participants.

Unlike the previous encryption system for multi-receiver, in this paper, we construct a new efficient scheme based on chaotic maps named MRCM. As a basic algorithm, chaotic maps [9, 12, 23, 28] not only meet the operation efficiency, but also possess strong functionality. Therefore, we utilize traditional public key encryption method which based on chaotic maps to realize information transmission. Besides, as far as we know, it is the very first time that the researchers introduce a chaotic maps-based encryption scheme in the multi-receiver setting.

Due to in the IBE model [1], where the private key is allocated by a trusted private key generator (PKG), the unique private key generator is under great deal of work pressure. If the PKG system collapsed, all of the legal receivers will unable obtain their own private key, which will seriously affect the communication between the sender and receivers. For the purpose of overcome this potential problem, our scheme uses the conventional public/private key pairing (pk_i, sk_i) to achieve message encrypt/decrypt. With this method, the single-point is dispersed into multi-point so that can eliminate the insecurity cased by PKG, and improve the efficiency indirectly. At the same time, different from the scheme which depends on bilinear pairing to obtain anonymity in [24] and [16], endowed with anonymity by nature is our biggest advantage.

The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Next, a new chaotic maps-based multi-receiver scheme is described in Section 3. In Section 4, we give the security of our proposed protocol. The efficiency analysis of our proposed protocol is given in Section 5. This paper is finally concluded in Section 6.

2 Preliminaries

2.1 Pseudo-random Function Ensembles

If a function ensemble $F = \{F_n\}_{n\in N}$ is pseudo-random [15], then for every probabilistic polynomial oracle A and all large enough n, we have that:

$$Adv^F(A) = |Pr[A^{F_n}(1^n) = 1] - Pr[A^{G_n}(1^n) = 1]| < \varepsilon(n),$$

where $G = \{G\}_{n\in N}$ is a uniformly distributed function ensemble, $\varepsilon(n)$ is a negligible function, $Adv^F = max_A\{Adv^F(A)\}$ denotes all oracle A, and $Adv^F(A)$ represents the accessible maximum.

2.2 Definition and Hard Problems of Chebyshev Chaotic Maps

Let n be an integer and let x be a variable with the interval $[-1, 1]$. The Chebyshev polynomial [27] $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ is defined as $T_n(x) = cos(ncos^{-1}(x))$. Chebyshev polynomial map $T_n : R \rightarrow R$ of degree n is defined using the following recurrent relation:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x),$$

where $n \geq 2$, $T_0(x) = 1$, and $T_1(x) = x$.

The first few Chebyshev polynomials are:

$$T_2(x) = 2x^2-1, T_3(x) = 4x^3-3x, T_4(x) = 8x^4-8x^2+1, \cdots$$

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that

$$T_r(T_s(x)) = T_{rs}(x).$$

An immediate consequence of this property is that Chebyshev polynomials commute under composition

$$T_r(T_s(x)) = T_s(T_r(x)).$$

In order to enhance the security, Zhang [33] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$. The enhanced Chebyshev polynomials are used in the proposed protocol:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \bmod N,$$

where $n \geq 2$, $x \in (-\infty, +\infty)$, and N is a large prime number. Obviously,

$$T_{rs}(x) = T_r(T_s(x)) = T_s(T_r(x)).$$

Definition 1. *(Semi-group property) Semi-group property of Chebyshev polynomials:* $T_{rs}(x) = T_r(T_s(x)) = cos(rcos^{-1}(scos^{-1}(x))) = cos(rscos^{-1}(x)) = T_s(T_r(x)) = T_{sr}(x)$, *where* r *and* s *are positive integer and* $x \in [-1, 1]$.

Table 1: Notations

Symbol	Definition
ID_i	The identity of users
$U_i(0 \leq i \leq n)$	The users involved in CRRM scheme
a, b	Nonces
$(x, T_{K_i}(x))$	Public key of $user_i$ based on Chebyshev chaotic maps
K_i	Secret key of $user_i$ based on Chebyshev chaotic maps
F	Pseudo-random function
$\|\|$	Concatenation operation

Definition 2. *(Chaotic Maps-Based Discrete Logarithm (CDL) problem) Given x and y, it is intractable to find the integer s, such that $T_s(x) = y$. The probability that a polynomial time-bounded algorithm A can solve the CDL problem is defined as $Adv_A^{CDL}(p) = Pr[A(x, y) = r : r \in Z_p^*, y = T_r(x) \bmod p]$.*

Definition 3. *(CDL assumption) For any probabilistic polynomial time-bounded algorithm A, $Adv_A^{CDL}(p)$ is negligible, that is, $Adv_A^{CDL}(p) \leq \varepsilon$, for some negligible function ε.*

Definition 4. *(Chaotic Maps-Based Diffie-Hellman (CDH) problem) Given $x, T_r(x)$ and $T_s(x)$, it is intractable to find $T_{rs}(x)$. The probability that a polynomial time-bounded algorithm A can solve the CDH problem is defined as $Adv_A^{CDH}(p) = Pr[A(x, T_r(x) \bmod p, T_s(x) \bmod p) = T_{rs}(x) \bmod p : r, s \in Z_p^*]$.*

Definition 5. *(CDH assumption) For any probabilistic polynomial time-bounded algorithm A, $Adv_A^{CDH}(p)$ is negligible, that is, $Adv_A^{CDH}(p) \leq \varepsilon$, for some negligible function ε.*

2.3 Definition and Properties of Chebyshev Chaotic Maps [6, 10]

Definition 6. *$f : J \to J$ is said to be topologically transitive if for any pair of open sets $U, V \subset J$, there exists $k > 0$ such that $f^k(U) \cap V \neq \phi$.*

Definition 7. *$f : J \to J$ has sensitive dependence on initial conditions if there exists $\delta > 0$ such that for any $x \in J$ and any neighborhood N of x, there exist $y \in N$ and $n \geq 0$ such that $|f^n(x) - f^n(y)| > \delta$.*

Definition 8. *Let V be a set, then $f : V \to V$ is said to be chaotic on V if*

1) *f has sensitive dependence on initial conditions.*
2) *f is topologically transitive.*
3) *Periodic points are dense in V.*

Definition 9. *Let $f : A \to A, f : B \to B$ be two maps, if there exists a continuous surjection $h : A \to B$ such that $h \cdot g = g \cdot h$, we say that these two maps f and g are topologically semi-conjugate.*

Theorem 1. *A non-zero polynomial is the n^{th} Chebyshev polynomial or its constant times iff the nonzero polynomial is the root of the differential equation*

$$(1 - x^2)y'' - xy' + n^2 y = 0 (n \in Z_+).$$

Lemma 1. *If $f : A \to A, g : B \to B$ are topologically semi-conjugate,*

1) *When p is the periodic point of f, then $h(p)$ is the periodic point of g;*
2) *When the periodic point of f is dense in A, the periodic point of g is dense in B, where h is the topologically semi-conjugate between f and g.*

Lemma 2. *Assume $f : A \to B$ is a map, $A_0, A_1 \subset A$, then $f(A_0 \cap A_1) \subset f(A_0) \cap f(A_1)$.*

Lemma 3. *When $f : A \to A$ is topologically transitive, $g : B \to B$ is topologically semi-conjugate f via h, then g is topologically transitive.*

Lemma 4. *Let $R : S' \to S'$ be a map of the circle into itself, then $R(\theta) = n\theta (n \in Z, n \geq 2)$ is chaotic, where θ is the radian value.*

The concrete proof of chaotic properties can be found in the literature [10] and the enhanced properties of Chebyshev polynomials that defined on interval $(-\infty, +\infty)$ still have the semi-group property (see [33]).

3 The Proposed MRCM Scheme

3.1 Notations

The concrete notations used hereafter are shown in Table 1.

3.2 MRCM Scheme

Figure 1 illustrates the MRCM scheme.

Setup. Simply speaking, for all the users $U_i(0 \leq i \leq n)$, their public keys are $(x, T_{k_i}(x))(0 \leq i \leq n)$ and the corresponding secret keys are $k_i(0 \leq i \leq n)$. And without loss of generality, we assume the user U_0 is

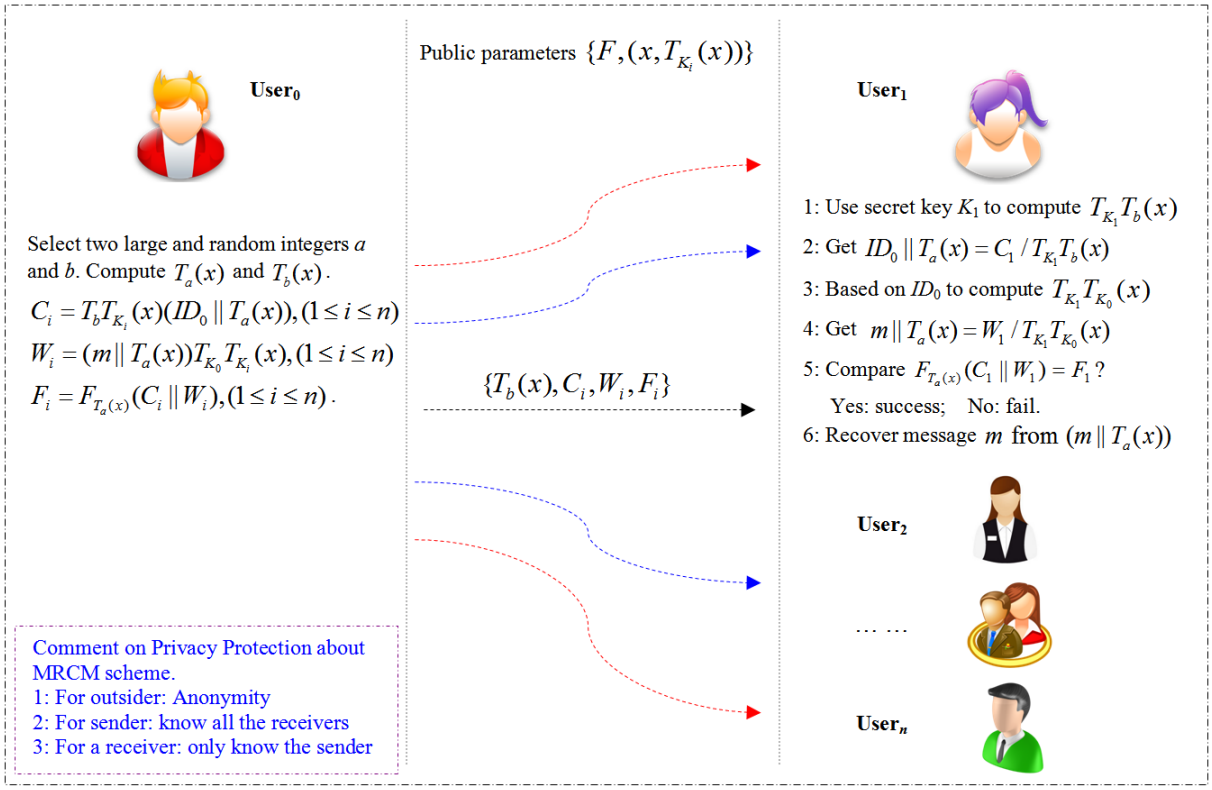


Figure 1: Chaotic maps-based multi-receiver with privacy protection scheme

the sender, and the users $U_i(1 \le i \le n)$ are the receivers. Due to space limitation in this paper, we are not able to discuss the details about how to distribute the public-private key pairs of the users.

Encrypt. When a user U_0 wants to send the same message m to the users $U_i(1 \le i \le n)$, she chooses two large and random integers a and b. Next, U_0 computes $T_a(x)$, $T_b(x)$, $C_i = T_b T_{K_i}(x)(ID_0||T_a(x)), (1 \le i \le n)$, $W_i = (m||T_a(x))T_{K_0}T_{K_i}(x), (1 \le i \le n)$ and $F_i = F_{T_a(x)}(C_i||W_i), (1 \le i \le n)$. Finally, U_0 sends $\{T_b(x), C_i, W_i, F_i\}$ to the users $U_i(1 \le i \le n)$.

Decrypt.

1) Upon receiving $\{T_b(x), C_i, W_i, F_i\}$ from the sender, firstly, any user can recover the identity of the sender by using secret key K_i to compute $T_{K_i}T_b(x)$ and get $ID_0||T_a(x) = C_1/T_{K_1}T_b(x)$.
2) Based the sender's identity ID_0, U_i can get the public key $T_0(x)$ and compute $T_{K_i}T_{K_0}(x)$ for getting $m||T_a(x) = W_1/T_{K_1}T_{K_0}(x)$. This step is also authenticating the sender, if the sender is the "sender", the last step any user can recover the right message, if not, the recovered message will not be the plaintext.
3) U_i authenticates the message integrity $F_{T_a(x)}(C_1||W_1) = F_1$?. If yes, the cipher-text is valid. Otherwise, the cipher-text is invalid or has been damaged during transmission.
4) Finally, based on their secret key K_i, any user in the group can recover the message $m = \frac{W_i}{T_{K_i}T_{K_0}(x)} - T_a(x) = \frac{W_i}{T_{K_i}T_{K_0}(x)} - (\frac{C_i}{T_{K_i}T_b(x)} - ID_0)$.

3.3 Consistency

Let $\{T_b(x), C_i, W_i, F_i\}$ be a valid ciphertext, for any user U_i, we have

$$\begin{aligned} & \frac{W_i}{T_{K_i}T_{K_0}(x)} - (\frac{C_i}{T_{K_i}T_b(x)} - ID_0) \\ = & \frac{W_i}{T_{K_i}T_{K_0}(x)} - (ID_0||T_a(x) - ID_0) \\ = & \frac{W_i}{T_{K_i}T_{K_0}(x)} - T_a(x) \\ = & m||T_a(x) - T_a(x) \\ = & m. \end{aligned}$$

4 Security Consideration

4.1 Security Analysis for Security Requirements and the Comparisons

There are many security requirements about protocol type. Because our proposed scheme is multi-receiver type with one message without exchanging process, there are

Table 2: Definition and the reasons why we do not discuss

Attack Type	Security Requirements	Definition	Reasons why we do not discuss
Automatic validation attacks	Guessing attacks (On-line or off-line)	In an off-line guessing attack, an attacker guesses a password or long-term secret key and verifies his/her guess, but he/she does not need to participate in any communication during the guessing phase. In an undetectable on-line guessing attack, an attacker searches to verify a guessed password or long-term secret key in an on-line transaction and a failed guess cannot be detected and logged by the server.	No password involved
	Losing smart device and guessing attacks	An adversary gets the user's smart device and then carries out the guessing attacks.	No password involved
	Human Guessing Attacks	In human guessing attacks, humans are used to enter passwords in the trial and error process.	No password involved
No freshness verify attacks	Perfect forward secrecy	An authenticated key establishment protocol provides perfect forward secrecy if the compromise of both of the node's secret keys cannot results in the compromise of previously established session keys.	No session key produced
	Known session key security	Each execution of the protocol should result in a unique secret session key. The compromise of one session key should not compromise the keys established in other sessions.	No session key produced

many security requirements no need to discuss (see Table 2).

Next, from the Table 3, we can see that the proposed scheme can provide known secure session key agreement, impersonation attack and so on.
Some other security attributes:

1) The security of one ciphertext with some authentications.

Theorem 2. *Our proposed scheme is one ciphertext security under the CMBDLP and CMBDHP assumptions.*

Proof. Our proposed scheme is based on PKC (Public Key Cryptosystem), so there are two key points should be taken into account: each message must mix with a large random nonce and any public key cannot be used to encrypt secret message directly. Therefore, we construct $W_i = (m||T_a(x))T_{K_0}T_{K_i}(x)$, $(1 \leq i \leq n)$ to covered the secret message m. The encrypted message W_i is generated from a which is different in each session and is only known by the sender U_0. Any receiver can decrypt W_i using his/her own secret key, but the decrypted process is completely different: The middle process value $T_{K_0}T_{K_i}(x)$ only can be computed by the corresponding receiver which is secure under the CMBDLP and CMBDHP assumptions, and furthermore getting the $m = m||T_a(x) - T_a(x)$. Additionally, since the values a of the random elements is very large, attackers cannot directly guess the values a of the random elements to generate $T_a(x)$. Therefore, the proposed scheme provides one ciphertext security. □

2) The security of privacy protection.

Theorem 3. *Our proposed scheme is privacy protection partly under the CMBDLP and CMBDHP assumptions.*

Proof. We divide the participants into three characters: the sender, the receivers and the outsiders (including attacker, any curious nodes and so on). The sender's identity is anonymity for outsiders because ID_0 is covered by $C_i = T_bT_{K_i}(x)(ID_0||T_a(x)), (1 \leq i \leq n)$, and then only the legal receivers can use his/her secret key to recover the ID_0. Due to PKC-based about our scheme, the ID_0 must be emerged to the legal receivers, or they cannot know the public key of the sender. The sender must know the receiver's identity because our scheme is adopted PKC and chaotic maps. All the receivers cannot know the others receivers because they only recover the corresponding C_i using their own secret key.

Table 3: Definition and simplified proof

Attack Type	Security Requirements	Definition	Simplified Proof	Hard Problems
Missing encrypted identity attacks	Man-in-the-middle attack(MIMA)	The MIMA attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.	All the information includes the ID and some nonces: a, b and the another form $T_a(x), T_b(x)$.	Chaotic maps problems
	Impersonation attack	An adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications protocol.	All the information includes the $ID, (pk_i, sk_i)$ and some nonces: a, b and the another form $T_a(x, T_b(x))$.	Chaotic maps problems
No freshness verify attacks	Replay attack	A replay attack is a form of network attack in which a valid data transmission is repeated or delayed maliciously or fraudulently.	Every important message includes the nonces: a, b and the another form $T_a(x), T_b(x)$.	Chaotic maps problems
Design defect attacks	Stolen-verifier attacks	An adversary gets the verifier table from servers by a hacking way, and then the adversary can launch any other attack which called stolen-verifier attacks.	There are no any verification tables in any node.	Chaotic maps problems

We construct $C_i = T_b T_{K_i}(x)(ID_0||T_a(x)), (1 \leq i \leq n)$ to covered the sender's identity. The encrypted message C_i is generated from b which is different in each session and is only known by the sender U_0. Any receiver can decrypt C_i using $T_b(x)$ and his/her own secret key, but the decrypted process is completely different: the middle process value $T_{K_i}T_b(x)$ only can be computed by the corresponding receiver which is secure under the CMBDLP and CMBDHP assumptions, and furthermore getting the $ID_0||T_a(x) = C_i/T_{K_i}T_b(x)$, you can get ID_0 and $T_a(x)$ in the same time. Additionally, since the values b of the random elements is very large, attackers cannot directly guess the values a of the random elements to generate $T_b(x)$. Therefore, the proposed scheme provides privacy protection.

The privacy protection of our MRCM scheme belongs to the ID hiding(a user may use a resource or service without disclosing the user's identity during the protocol interaction), anyway, we must emphasize three points:

a. Any outsider cannot get any ID information (sender or receivers) about our proposed scheme.

b. Only the sender knows the ID information of all receivers.

c. Any receiver cannot get any other receiver's ID information. We sum up the privacy protection of our scheme in the Table 4.

□

4.2 Security Proof Based on the BAN Logic

For convenience, we first give the description of some notations (Table 5) used in the BAN logic analysis and define some main logical postulates (Table 6) of BAN logic [3].

Remark 1. *$(X)_K$ means that the formula X is hash function with the key K. But in our scheme, we redefine $(X)_K$: the formula X is pseudo-random function with the key K to adopt the standard model.*

According to analytic procedures of BAN logic and the requirement of multi-receiver scheme, our MRCM scheme should satisfy the following goals in Table 7.

First of all, we transform the process of our protocol to the following idealized form.

$$(U_0 \to U_i)C : U_i \lhd T_b(x), T_b T_{K_i}(x)(ID_0||T_a(x)),$$

$$(m||T_a(x))T_{K_0}T_{K_i}(x), (C_i||W_i)_{T_a(x)};$$

According to the description of our protocol, we could make the following assumptions about the initial state,

Table 4: Privacy protection comparisons

Security attributes		[19] 2009	[31] 2010	[16] 2013	Ours
Missing encrypted identity attacks	For outsiders	No	Yes	Yes	Yes
	For receivers	No	No	No	No
Receiver anonymity	For outsiders	No	No	Yes	Yes
	For other receivers	No	No	No	Yes
	For the sender	No	No	No	No

Table 5: Notations of the BAN logic

Symbol	Definition
$P\mid\equiv X$	The principal P believes a statement X, or P is entitled to believe X.
$\#(X)$	The formula X is fresh.
$P\mid\Rightarrow X$	The principal P has jurisdiction over the statement X.
$P\lhd X$	The principal P sees the statement X.
$P\mid\sim X$	The principal P once said the statement X.
(X,Y)	The formula X or Y is one part of the formula (X,Y).
$\langle X\rangle_Y$	The formula X combined with the formula Y.
X_K	The formula X is encrypted under the key K.
$(X)_K$	The formula X is hash function with the key K. If there is no K, and that means is no key input.
$P\overset{K}{\longleftrightarrow}Q$	The principals P and Q use the shared key K to communicate. The key K will never be discovered by any principal except P and Q.
$\overset{K}{\longrightarrow}P$	The public key of P, and the secret key is described by K^{-1}.

Table 6: Logical postulates of the BAN logic

Symbol	Definition
$\frac{P\mid\equiv P\overset{K}{\longleftrightarrow}Q,P\{X\}_K}{P\mid\equiv Q\mid\sim X}$	The message-meaning rule (R_1)
$\frac{P\mid\equiv\#(X)}{P\mid\equiv\#(X,Y)}$	The freshness-conjunction rule (R_2)
$\frac{P\mid\equiv\#(X),P\mid\equiv Q\mid\sim X}{P\mid\equiv Q\mid\equiv X}$	The nonce-verification rule (R_3)
$\frac{P\mid\equiv Q\mid\Rightarrow X,P\mid\equiv Q\mid\equiv X}{P\mid\equiv X}$	The jurisdiction rule (R_4)
$\frac{P\mid\equiv Q\mid\equiv(X,Y)}{P\mid\equiv Q\mid\equiv X}$	The belief rules (R_5)
Remark 3: Molecule can deduce denominator for above formulas.	

Table 7: Goals of the proposed scheme

Goals
Goal 1. $U_0\mid\equiv(U_0\overset{m}{\longleftrightarrow}U_i)$; Goal 2. $U_0\mid\equiv U_i\mid\equiv(U_0\overset{m}{\longleftrightarrow}U_i)$;
Goal 3. $U_i\mid\equiv(U_i\overset{m}{\longleftrightarrow}U_0)$; Goal 4. $U_i\mid\equiv U_0\mid\equiv(U_i\overset{m}{\longleftrightarrow}U_0)$;
Where U_0 means the sender, $U_i(1\leq i\leq n)$ means the $n-receiver$, and m means the messages.

Table 8: Assumptions about the initial state of our protocol

Initial states	
$P_1 : U_0\| \equiv \xrightarrow{T_{K_i}(x)} U_i$	$P_2 : U_i\| \equiv \xrightarrow{T_{K_0}(x)} U_0$
$P_3 : U_0\| \equiv \#(a)$	$P_4 : U_0\| \equiv \#(b)$
$P_5 : U_0\| \equiv U_0 \xleftrightarrow{T_{K_0}T_{K_i}(x)} U_i$	$P_6 : U_i\| \equiv U_i \xleftrightarrow{T_{K_i}T_{K_0}(x)} U_0$

which will be used in the analysis of our protocol in Table 8.

Based on the above assumptions, the idealized form of our protocol is analyzed as follows. The main steps of the proof are described as follows. According to the ciphertext C and P_2, P_6 and attributes of chaotic maps, and relating with R_1, we could get:

$$S_1 : U_i\| \equiv U_0\| \sim C_i.$$

Based on the initial assumptions P_3, P_4 and relating with R_2, we could get:

$$S_2 : U_i\| \equiv \sharp C_i.$$

Combine $S_1, S_2, P_3, P_4, P_5, P_6, R_3$ and attributes of chaotic maps, we could get:

$$S_3 : U_i\| \equiv \sharp ID_0, T_a(x), T_b(x).$$

Based on R_5, we take apart S_3 and get:

$$S_4 : U_i\| \equiv \sharp T_b(x), S_5 : U_i\| \equiv \sharp T_a(x).$$

Combine S_3, S_4 and attributes of chaotic maps, we can get the fresh and privacy protection about sender's identity. Combine S_5 and attributes of chaotic maps, we can get the message m for all the $U_i(1 \le i \le n)$.

Combine 1. *Because the 1-to-n parties (U_0 and $U_i(1 \le i \le n)$) communicate each other just now, they confirm the other is on-line. Moreover, since the $U_i(1 \le i \le n)$ can get ID_0 and $T_a(x)$ from the $T_bT_{K_i}(x)(ID_0||T_a(x))$ with his own secret key, and based on S_5, R_4 with chaotic maps problems, we could get:*

Goal 1. $U_0\| \equiv (U_0 \xleftrightarrow{m} U_i)$;

Goal 2. $U_0\| \equiv U_i\| \equiv (U_0 \xleftrightarrow{m} U_i)$;

Goal 3. $U_i\| \equiv (U_i \xleftrightarrow{m} U_0)$;

Goal 4. $U_i\| \equiv U_0\| \equiv (U_i \xleftrightarrow{m} U_0)$;

According to (Goal 1 $\sim$ Goal 4), we know that both sender U_0 and receivers $U_i(1 \le i \le n)$ believe that the $U_i(1 \le i \le n)$ can authenticate U_0 and recover the message based on the fresh nonces a, b and the $(pk_i, sk_i)(0 \le i \le n)$.

5 Efficiency Analysis

5.1 The Comparisons among Different Algorithms

Compared to RSA[1], ECC[2] and Bilinear map[3], Chebyshev polynomial computation problem offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings. Chaotic maps encryption algorithm: As a special form of motion, Chaos means that in a certain nonlinear system can appear similar to the behavior of random phenomena without needing any random factors. Chaotic system has the characteristics of certainty, boundness, sensibility to initial parameters and unpredictability, etc. Chaotic maps encryption algorithm utilizes the unique semi-group mature of Chebyshev chaotic maps, based on two difficult problems-the chaotic maps discrete logarithm problem and the chaotic maps Diffie-Hellman problem, puts forward a kind of encryption algorithm. Compared with ECC encryption algorithm, Chaotic maps encryption algorithm avoids scalar multiplication and modular exponentiation computation, effectively improves the efficiency. However, Wang [27] proposed several methods to solve the Chebyshev polynomial computation problem. To be more precise, on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, the computational time of a one-way hashing operation, a symmetric encryption/decryption operation, an elliptic curve point multiplication operation and Chebyshev polynomial operation is 0.0005s, 0.0087s, 0.063075s and 0.02102s separately [11]. Moreover, the computational cost of XOR operation could be ignored when compared with other

[1] **RSA encryption algorithm:** RSA encryption algorithm is a kind of algorithm based on big integer factorization, its public keys and secret keys are the function of two large prime numbers (Which large prime numbers are more than 100 digits of decimal.). RSA encryption algorithm, as the first algorithm which can be used to encryption and digital signature, is easily to understand and operate.

[2] **ECC encryption algorithm:** ECC encryption algorithm is a kind of public-key cryptosystem algorithm, its mathematical theory is that using the rational points on the Elliptic curve constitutes Abel additive group, and utilizes the computational difficulty of discrete logarithm.

[3] **Bilinear map:** In mathematics a pairing function is a process to uniquely encode two natural numbers into a single natural number. In mathematics, a bilinear map is a function combining elements of two vector spaces to yield an element of a third vector space. It is called bilinear because it is linear in each of its arguments.

operations. According to the results in [2], one pairing operation requires at least 10 times more multiplications in the underlying finite field than a point scalar multiplication in ECC does in the same finite field.

Through the above mentioned analysis, we can reached the conclusion approximately as follows:

$$T_p \approx 10T_m, T_m \approx 3T_c, T_c \approx 2.42T_s, T_s \approx 17.4T_h,$$

we sum up these formulas into one so that it can reflect the relationship among the time of algorithms intuitively.

$$T_p \approx 10T_m \approx 30T_c \approx 72.6T_s \approx 1263.24T_h,$$

where T_p: Time for bilinear pair operation; T_m: Time for a point scalar multiplication operation; T_c: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial; T_s: Time for symmetric encryption algorithm; T_h: Time for Hash operation. Table 9 given the comparison for RSA, ECC and Chaotic maps.

About these algorithms, our proposed multi-receiver scheme only used the chaotic cipher as the main algorithm which is more efficient bilinear pair operation and a point scalar multiplication operation ECC-based (see the Table 10). As for Hash operation and pseudo-random function, it can be ignored compared with the other three algorithms.

5.2 The Efficient Usage about Chaotic Maps

Most of chaotic maps-based protocols for achieving key agreement or encrypted messages usually adopt $ChaoticMaps-BasedDiffie-Hellman(CDH)problem$ to get the same session key to encrypting/decrypting messages transferred between user and server [9, 28, 29]. But our proposed scheme only uses $CDHproblem$ to get temporary key for attaching messages to it, which can make our scheme more efficient, and the users's privacy information is protected. In other words, we change the usage of chaotic maps from the form $E_{T_aT_b(x)}(messages)$ to another form $T_aT_b(x) \cdot (messages)$, obviously, the latter is much more efficient than the former.

5.3 The Comparisons among Our MRCM Scheme and the Related Literatures

In this section, we make a comparison between the MRCM and other multi-receiver scheme to judge its function and competence. From Table 10, we can conclude that our scheme is more efficient than the others.

6 Conclusion

In this paper, we propose MRCM, a novel scheme towards building a PKC-based scheme for a sender sending only one encrypted message with some authentication information to multi-receiver, and at the same time, achieving privacy protection. The core idea we have followed is that the most existing multi-receiver schemes are bilinear pairing-based, for improving the efficiency, should be exploited to securely change another efficient cryptosystem, such as, chaotic maps in this paper. Since the hash function is not used, and chaotic maps is adopted to a new encrypted algorithm without using symmetrical encryption, the proposed solution offers significant advantages (the standard model and high-efficiency) with respect to a traditional multi-receiver protocols. Compared with the related works, our MRCM scheme is not the trade off between security and efficiency, but is comprehensively improved scheme.

Acknowledgments

This work is supported by the Liaoning Provincial Natural Science Foundation of China (Grant No. 201602680).

References

[1] J. Baek, R. Safavinaini and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," *Lecture Notes in Computer Science*, vol. 3386, pp. 380–397, Springer, 2005.

[2] P. S. L. M. Barreto, B. Lynn and M. Scott, "On the selection of pairing-friendly groups," in *Selected Areas in Cryptography*, LNCS 3006, pp. 17–25, Springer-Verlag, 2004.

[3] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, pp. 18–36, 1990.

[4] H. L. Chan, X. Deng, H. Zhu, "Design and security analysis of anonymous group identification protocols," *Lecture Notes in Computer Science*, vol. 2274, pp. 188–198, Springer, 2002.

[5] B. Dan, M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal on Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[6] R. L. Devaney, J. P. Eckmann, "An introduction to chaotic dynamical systems," *Mathematical Gazette*, vol. 19, no. 2, pp. 204–205, 1990.

[7] E. Ekrem, S. Ulukus, "Multi-receiver wiretap channel with public and confidential messages," *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 2165–2177, 2013.

[8] C. Fan, L. Y. Huang and P. H. Ho, "Anonymous multi receiver identity-based encryption," *IEEE Transactions on Computers*, vol. 59, no. 9, pp. 1239–1249, 2010.

[9] C. Guo, C. C. Chang, "Chaotic maps-based password-authenticated key agreement using smart cards," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, pp. 1433–1440, 2013.

Table 9: Comparison for RSA, ECC and Chaotic maps

	RSA encryption algorithm	ECC encryption algorithm	Chaotic maps encryption algorithm
Items	Differences		
Mathematical basis	Large prime number	Elliptic curve	Chebyshev polynomial
Difficult problem assumptions	large prime factorization problem	Discrete logarithm calculation problem on the elliptic curve	Chaotic maps discrete logarithm problem, Chaotic maps Diffie-Hellman problem
Operation Cost	√ √	√ √ √	√ √ √
Operation Speed	√	√ √	√ √ √
Security Level	√	√ √ √	√ √ √
Normal √	Good √ √	Excellent √ √ √	

Table 10: Comparisons between our proposed scheme and the related literatures

Phase	Some property	[19] 2009	[31] 2010	[16] 2013	Ours
Encrypt	Number of parameters	$n+9$	13	10	2
	Computation Complexity	$(n+1)T_a + (n+3)T_m + T_e + 2T_h$	$T_p + (m+n+1)T_m + (2m+n+3)T_e + 2T_h$	$T_p + 2T_a + 6T_m + T_e + 2T_h$	$nT_f + 2nT_c + 2nT_{mo}$
	Ciphertext length	$3\|G_1\| + \|M\| + n\|ID\|$	$(m+n+2)\|G_1\| + \|M\| + m\|ID\|$	$(n+4)\|G_1\| + \|M\|$	$(2n+1)\|G_2\| + n\|F\|$
Decrypt	Ciphertext validity or integrity	$3T_p + 2T_a + (3n+3)T_m + 2T_e + (n+1)T_h$	$(m+5)T_p + T_a + (m+\|M\|+2)T_m + 2T_h$	$2T_p + T_a + T_m + T_h$	T_f
	Authorized or not	$3T_p + 2T_a + (3n+3)T_m + 2T_e + (n+1)T_h$	$(m+5)T_p + T_a + (m+\|M\|+2)T_m + 2T_h$	$2T_p + T_a + T_m + T_h$	No need
	Decryption	$3T_p + 2T_a + (3n+3)T_m + 2T_e + (n+1)T_h$	$(m+5)T_p + T_a + (m+\|M\|+2)T_m + 2T_h$	$2T_p + nT_a + (n-1)T_m + 2T_h$	$2T_c + 2T_{mo}$
Model		Random Oracle	Random Oracle	Random Oracle	Standard Model

T_p: Time for bilinear pair operation
T_a: Time for addition operation
T_m: Time for a point scalar multiplication operation
T_{mo}: Time for integer multiplication operation in the field
T_e: Time for exponentiation operation
T_h: Time for Hash operation
T_s: Time for symmetric encryption algorithm
T_c: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial using the algorithm in literature [33].
T_f: Time for pseudo-random function

$|G_1|$: The length of the elements in G_1; $|G_2|$: The length of the elements in G_2; $|ID|$: the length of ID;
Let G_1 be an additive group and G_2 be a multiplicative group with the same prime order q;
$|M|$: The length of the plaintext M; $|F|$: the length of the output of pseudo-random function.
m: The number of signers/sender (m=1 in schemes [16, 19, 31] and our scheme); n: the number of receivers.

Random Oracle: A random oracle is a random mathematical function, that is, a function mapping each possible query to a (fixed) random response from its output domain, for example, regarding hash function as a real random mathematical function in the practical application.
Standard Model: The standard model is the model of computation in which the adversary is only limited by the amount of time and computational power available, without using a random mathematical function.

[10] J. C. Jiang, Y. H. Peng, "Chaos of the Chebyshev polynomials," *Natural Science Journal of Xiangtan University*, vol. 19, no. 3, pp. 37–39, 1996.

[11] L. Kocarev, S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*, pp. 53–54, 2011.

[12] C. C. Lee, D. C. Lou, C. T. Li, C. W. Hsu, "An extended chaotic-maps-based protocol with key agreement for multi server environments," *Nonlinear Dynamics*, vol. 76, no. 1, pp. 853–866, 2014.

[13] H. Li, L. Pang, "Cryptanalysis of Wang et al.'s improved anonymous multi-receiver identity-based encryption scheme," *IET Information Security*, vol. 8, no. 1, pp. 8–11, 2014.

[14] J. Minonzio, M. Talmant and P. Laugier, "Multi-emitters and multi-receivers probe for long cortical bone assessment," *Journal of the Acoustical Society of America*, vol. 127, no. 3, pp. 2032–2035, 2010.

[15] PR Newswire, "Ticketmaster Launches New, Innovative CAPTCHA Solutions, Making The Fan Experience Better," *PR Newswire*, US, 2013. (`http://www.prnewswire.com/news-releases/ticketmaster-launches-new-innovative-captcha-solutions-making-the-fan-experience-better-189000181.html`)

[16] L. Pang, H. Li, L. Gao, Y. Wang, "Completely anonymous multi-recipient signcryption scheme with public verification," *PLoS ONE* vol. 8, no. 5, 2013.

[17] H. Qin, Y. Dai and Z. Wang, "Identity-based multi-receiver threshold signcryption scheme," *Security and Communication Networks*, vol. 4, no. 11, pp. 1331–1337, 2011.

[18] K. R. Santosh, C. Narasimham, and P. Shetty, "Cryptanalysis of multi-prime RSA with two decryption exponents," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 40–44, 2016.

[19] S. S. D. Selvi, S. S. Vivek, R. Srinivasan, C. P. Rangan, "An efficient identity-based signcryption scheme for multiple receivers," in *Proceedings of the 4th International Workshop on Security*, pp. 71–88, 2009.

[20] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," *Lecture Notes in Computer Science*, vol. 196, pp. 47–53, Springer, 1985.

[21] Z. Shen, H. Yu, Y. Hu and C. Shen, "Joint symbol detection for multi-receiver without signal synchronization and array alignment," *IEEE Communications Letters*, vol. 18, no. 10, pp. 1755–1758, 2014.

[22] Y. X. Sun, H. Li, "Efficient signcryption between TPKC and IDPKC and its multi-receiver construction," *Science in China. Series F: Information Sciences*, vol. 53, no. 3, pp. 557–566, 2010.

[23] Z. Tan, "A chaotic maps-based authenticated key agreement protocol with strong anonymity," *Nonlinear Dynamics*, vol. 72, pp. 311–320, 2013.

[24] H. Wang, "Anonymous multi-receiver remote data retrieval for pay-TV in public clouds," *IET Information Security*, vol. 9, no. 2, pp. 108–118, 2015.

[25] H. Wang, H. Zhang, J. Li and X. U. Chen, "A(3,3) visual cryptography scheme for authentication," *Journal of Shenyang Normal University(Natural Science Edition)*, vol. 31, no. 3, pp. 397–400, 2013.

[26] H. Wang, Y. Zhang, H. Xiong and B. Qin, "Cryptanalysis and improvements of an anonymous multi-receiver identity-based encryption scheme," *IET Information Security*, vol. 6, no. 1, pp. 20–27, 2012.

[27] X. Wang, J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, pp. 4052–4057, 2010.

[28] Q. Xie, J. Zhao, X. Yu, "Chaotic maps-based three-party password-authenticated key agreement scheme," *Nonlinear Dynamics*, vol. 74, pp. 1021–1027, 2013.

[29] J. H. Yang, T. J. Cao, "Provably secure three-party password authenticated key exchange protocol in the standard model," *Journal of Systems and Software*, vol. 85, pp. 340–350, 2012.

[30] E. J. Yoon, K. Y. Yoo, J. W. Hong, S. Y. Yoon, D. I. Park and M. J. Choi, "An efficient and secure anonymous authentication scheme for mobile satellite communication systems," *Eurasip Journal on Wireless Communications and Networking*, vol. 10, pp. 1687–1695, 2011.

[31] B. Zhang, Q. Xu, "An ID-based anonymous signcryption scheme for multiple receivers secure in the standard model," in *Proceedings of the AST/UCMA/ISA/CAN Conferences*, pp. 15–27, 2010.

[32] J. Zhang, J. Mao, "An improved anonymous multi-receiver identity-based encryption scheme," *International Journal of Communication Systems*, vol. 28, pp. 645–658, 2015.

[33] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos Solitons Fractals*, vol. 37, no. 3, pp. 669–674, 2008.

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 international conference papers on the above research fields.

Hongfeng Zhu obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hongfeng Zhu is a full associate 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. Dr. Zhu had published more than 50 international journal papers (SCI or EI journals) on the above research

fields.

Xueshuai Feng graduated with a Bachelor of Engineering from Shenyang Normal University in 2015. In his college, after completing the learning task, he interests in exploring his professional knowledge. During graduate, under the guidance of his master instructor, he researches IoT security theory and technology.

Characterization and Comparison of DDoS Attack Tools and Traffic Generators - A Review

Sunny Behal, Krishan Kumar
(Corresponding author: Sunny Behal)
Department of Computer Science & Engineering, Punjab Technical University
SBS State Technical Campus, Ferozepur, Punjab, India
(Email: sunnybehal@sbsstc.ac.in)
(Received Jan. 13, 2016; revised and accepted Mar. 19 & Apr. 17, 2016)

Abstract

Distributed Denial of Service (DDoS) attack imposes a severe threat to the extensively used Internet based services like e-commerce, e-banking, transportation, medicine, education etc. Hackers compromises the vulnerable systems for launching DDoS attacks in order to degrade or sometimes completely disrupt such services. In recent years, DDoS attacks have been increased in frequency, sophistication and strength. Though a no. of solutions have been proposed in literature to combat against DDoS attacks but still defending from a DDoS attack is a challenging issue. Hackers are also continuously upgrading their skills to launch diversified attacks and are developing new sophisticated attack tools and traffic generators to circumvent these countermeasures. The purpose of this paper is to characterize and compare the popular DDoS attack tools and traffic generators used by the attackers in recent times. The technical details provided would surely help the researchers to handpick the appropriate DDoS attack tool and traffic generator for designing their real experiments so that their proposed DDoS defense methods could be validated in a better way.

Keywords: Attack tools, DDoS, network security, traffic generators

1 Introduction

A DDoS attack is a malicious attempt from multiple systems to make computer or network resources unavailable to its intended users, usually by interrupting or suspending services connected to the Internet [13].

DDoS attacks are launched through the well organized, distributed and remotely controlled network so that compromised computers (called zombies or bots) can be used for sending large volume of continuous and simultaneous attack requests to the target system(s). DDoS attacks mainly cause unusual behavior in the form of unavailability, inability to access the particular website or a service and slow down the performance of the network. As a result, the target systems responds slowly or are completely crashed. The DDoS attacks that are launched by causing the disruption in the legitimate user connectivity (exhaustion of bandwidth, reducing router processing capacity and network resource usage) are termed as Network layer attacks whereas the attacks that are launched by disruption in the legitimate user services (exhaustion of the server resources like CPU, memory, disk/database bandwidth, sockets, input/output bandwidth) are termed as Application layer attacks [1, 6, 7, 12, 15, 29]. In recent years, DDoS attacks have been increased in strength, frequency and sophistication. The attackers are continuously upgrading their skills and modifying their modus operandi and are using latest technologies to launch diversified DDoS attacks. Although, many solutions have been proposed by the researchers to detect, prevent or mitigate DDoS attacks, but still attackers are persistently developing new methods and means to circumvent these countermeasures.

There are number of tools available that can generate the similar looking legitimate traffic as well as attack traffic and can easily circumvent the existing DDoS defense solutions. For example, D-ITG [5, 8, 9, 16, 18] is such a powerful traffic generator that can be used to generate legitimate as well as attack traffic. It has been observed that all of the DDoS attacks are launched now-a-days by using botnets [45].

The key contributions of this paper are:

1) To Identify various attack tools used for launching DDoS attacks.
2) To study and investigate the characteristics of attack tools.
3) To compare and characterize the attack tools on identified attributes.
4) To propose attack tools taxonomy.
5) To identify, compare and characterize various legitimate and background traffic generators.

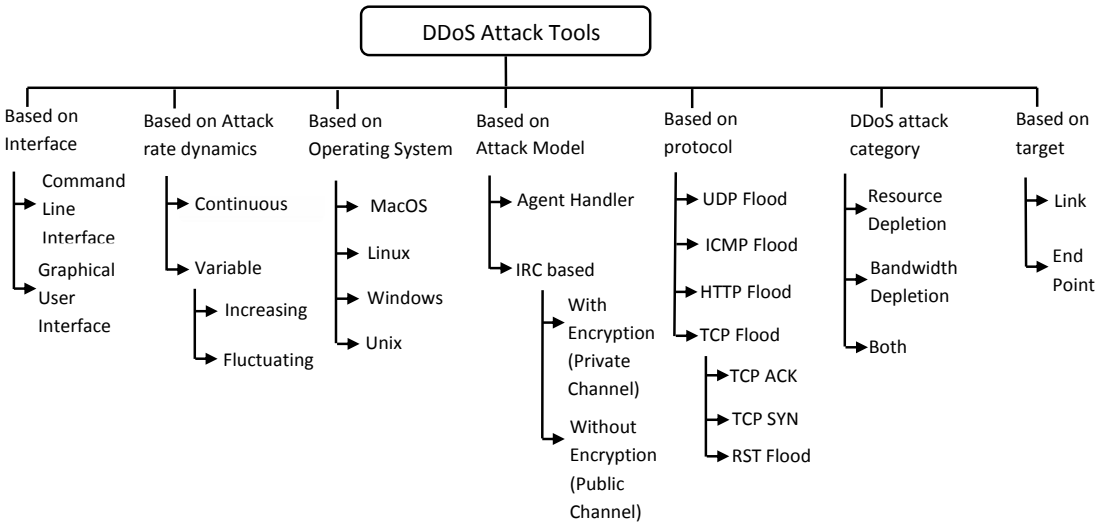


Figure 1: Taxonomy of DDoS attack tools

6) To provide research directions for the design of real-time experiments for validating DDoS defense.

This paper is a unique attempt that highlights the key technical features of the DDoS attack tools and traffic generators used by the attackers to launch DDoS attacks like their architecture model, the type of protocols supported or the type of traffic generated etc. The detailed technical information about the attack tools and traffic generators is provided with reference to the real experimentation purposes so that the internal working of these tools could be unleashed. This information would help the researchers to choose the appropriate DDoS attack tools or traffic generators for their real time experimentations so that better solution to the ever growing DDoS problem could be developed.

In the literature, there are number of existing surveys on the botnets and DDoS attack tools [20, 28, 31, 32, 42, 44] but none of them is complete in itself. Kumar [4], Mirkovic [33] and Specht [41] presented the taxonomies of attack tools but did not categorized the attack tools and traffic generators. Hoque [20] and Srivastva [42] provided the taxonomy of DDoS attacks and key features of few popular DDoS attack tools but lack the technical details. kaur et al. [28] presented the some of the typical DDoS attack tools used by the attackers but did not give any information about traffic generators and their usage. In spite of these extensive surveys, a comprehensive solution to DDoS attacks have not been formulated till date. What is lacking in the literature is the detailed comparison based on the key technical features of the DDoS attack tools and traffic generators so that a better solution could be developed. This survey is first of its kind as per our knowledge that provides the detailed technical details, do the characterization and comparison of popular DDoS attack and traffic generators. There are number of DDoS attack tools and traffic generators available but no one has focused to sum up all the features under single title till date. This paper also provide the detailed comparison of these attack tools and traffic generators along with their technical details.

This paper is organized as follows: Section 2 presents the taxonomy of DDoS attack tools and their description, Section 3 emphasizes on the comparison of the DDoS attack tools based upon their key features, Section 4 provides the comparison of traffic generators and Section 5 concludes the paper by highlighting the need for realtime experimentations.

2 Taxonomy of DDoS Attack Tools and Their Comparison

In this section, the taxonomy of the DDoS attack tools (as shown in Figure-1) is provided based on the identified attributes like type of interface they used, their attack rate dynamics, target operating system, attack model, protocols used, DDoS attack category and target area [4, 21, 27, 33, 41].

Type of Interface used: The interface used by the DDoS attack tools can be either command line interface or graphical user interface. Goldeneye, trinoo, shaft etc. use command line interface whereas hoic, udp flooder, xoic etc use graphical user interface.

Attack rate dynamics: Depending upon the attack rate dynamics, attack tools can either generate the continuous attack traffic (no variations in sending attack request) rate and variable attack rate (tool can vary the attack rate to avoid the detection which can be the increasing rate and fluctuating rate).

Operating System Supported: A number of DDoS attack tools are designed to support the various operating systems like unix, linux, solaris or windows.

Attack model: DDoS attack tools can make use of either Agent-Handler model or and IRC model. Agent-Handler is based on master-slave relation whereas the IRC system use public channels for launching attacks.

Protocol: The type of protocol specifies the kind of traffic generated by the attack tools for generating flood attacks, communication between the agent-handler, handler-client and client-agent. Flood attacks mainly use UDP, ICMP (ICMP ECHO request and ICMP ECHO reply), HTTP, TCP (TCP-SYN, TCP-ACK and RST-flood) protocols.

DDoS attack category: The consequence of a DDoS attack is the unavailability of the resources or bandwidth of the victim. Hence, the attackers use those attack tools that can exhaust target system or network's resources and bandwidth. There are number of DDoS attack tools available that can deplete both the resources and the bandwidth in no time.

Target area: DDoS attacks can either congest the link or end point. So, DDoS attack tools are typically designed for the congestion at the link level(congestion at the victim network) or at the end point level(congestion at the victim server).

All the popular attack tools are compared on the basis of identified key features as shown in Table:1. The key features includes the impact of attack which cause depletion either at bandwidth or resource level, scope of the attack tool, the type of attack launched, support of operating systems, implementation language etc. Further, it has also been observed that all DDoS attack tools follows the same generalized attack tool architecture as given by Lee [41].

Stacheldraht. Stacheldraht is the C-based DDoS tool that can create the ICMP flood, SYN flood, UDP flood and Smurf attack towards the target. It has the capability to congest the link and spoof the IP address. It can run on the Linux and the Solaris 2.1. It has command line based interface and its DDoS attack architecture model is agent based [20].

TFN. TFN (tribe flood network) can generate a number of different kinds of attacks. It is also called the "Son Of Trinoo". It is the command line based which executes on the windows, linux etc. It is written in the C language and has the attack architecture similar to the handler-agent model. It generates DDoS attack that has the capability to deplete both resource and bandwidth [38] of the target.

Trinity. Trinity is the command line based attack tool that can launch UDP, fragment, SYN, RST, random, flags and null flood requests that leads to the endpoint resource exhaustion and link congestion. This tool uses the encrypted format and requires the Linux platform. The architecture model of the Trinity is the IRC-based [20].

Bubonic. A C-based attack tool which can use Linux, Unix and Windows as the underlying platforms for its execution. It is a DoS attempt to exploit or victimize the windows2000 machine by randomly sending a huge volume of the TCP packets with the random settings to increase the load on the machines which leads the machines to a crash. Random settings involve the setting of random IP addresses and random port number [20].

Jolt. A command line based DoS attack tool sends a large number of ICMP packets in order to target the victim machine running on the windows 95 or NT so that the victim machine fails to reassemble them for use. It implementation language is C. However, this kind of attack do not cause any drastic damage to the victim system, and the machine is still in the state, to be recovered [20].

Mstream. A C-based and command line interface DDoS attack tool has ability to forge the source addresses. It creates the TCP ACK flood and TCP RST flood requests to the target server. It can generate botnets and also spoof the ip addresses of the attackers while performing DDoS attacks. Both of these requests can exhaust the network resources and consumes bandwidth of the victim server [38].

Shaft. Shaft is the command line interface DDoS attack tool that can exhaust the bandwidth and resources of the victim server. It provides statistics for TCP, UDP and ICMP flooding attacks and helps the attackers to identify the victim machine status (either completely down or alive) or to decide on the termination of zombies in addition to the attack. Its architecture model is Agent-Handler based [38].

Targa. Targa is the C-based attack tool which can deplete the bandwidth and resources. It is the DoS attack tool which is the collection of the 16 different programs of DoS. These attacks can be launched individually as well as in the group also. It has the ability to spoof the ip addresses and requires the linux platforms [20].

Trinoo. Trinoo is the DDoS attack tool, that uses a master host and several broadcast hosts. Master host instructs the various broadcast hosts to launch the attack. An Application layer attack tool that has the capability to deplete the resources and leverages the bandwidth of the victim network. It is command line based and its architecture model is the Agent-Handler based [38].

Blast20. Blast20 is the DOS attack tool is called as the TCP service stress tool is able to identify the potential weaknesses in the network servers instantly.

Table 1: Comparison of DDoS Attack Tools

Year	Name	Target Impact	Scope	Type of Attack Traffic	Operating System supported	Number of Zombies	whether makes botnets? (yes/no)	Encryption (yes/no)	Ip Spoofing (yes/no)	Implementation Language	Interface Type	Attack Model
1999	Stacheldraht [20]	Bandwidth, Resource	DoS, DDoS	icmp, udp	linux, solaris	Multiple	yes	yes	yes	C	CLI	Agent based
1999	TFN Tribe flood network) [38]	Bandwidth, Resource	DDoS	tcp, udp, icmp	windows, linux,solaris	Multiple	yes	no	yes	C	CLI	Agent based
1999	Trinity cite-Hoq2014	Bandwidth, Resource	DoS, DDoS	tcp, udp	linux	Multiple	yes	no	no	-	CLI	IRC based
2000	Bubonic [20]	Bandwidth, Resource	DoS	tcp	windows, linux, unix	single	no	no	no	C	CLI	-
2000	Jolt [20]	Resource	DoS	icmp	window95, windowsNT	Single	no	no	yes	C	CLI	-
2000	Mstream [38]	Bandwidth	DoS, DDoS	tcp, udp, icmp	linux, windows	Multiple	yes	no	yes	C	CLI	Agent based
2000	Shaft [38]	Bandwidth, Resource	DoS, DDoS	udp,icmp, tcp	linux, unix	Multiple	yes	no	yes	-	CLI	Agent based
2000	Targa [20]	Bandwidth, Resource	DoS	tcp, udp, icmp	linux	Single	yes	no	yes	C	CLI	-
2000	Trinoo [38]	Bandwidth	DDoS	udp, tcp,http	linux, solaris	Multiple	yes	yes	no	C	CLI	Agent based
2001	Blast20 [20]	Resource	DoS	tcp	windows, linux, unix	Single	no	-	-	-	CLI	-
2001	Crazy Pinger [20]	Bandwidth, Resource	DoS	icmp	windows, linux, unix	Single	no	no	yes	-	GUI	-
2001	Kaiten [20]	Bandwidth, Resource	DDoS	tcp, udp	windows	Multiple	Yes	no	no	-	CLI	IRC based
2001	Knight [38]	Bandwidth, Resource	DDoS	tcp, udp	windows	Mutiple	yes	no	-	C	CLI	IRC based
2003	Nemsey [38]	Bandwidth	DoS	tcp	windows	Single	no	no	no	-	GUI	-
2005	FSMax [20]	Resource	DoS	-	windows	Single	no	no	no	-	CLI	-
2005	Hping [20]	Resource	DoS	icmp, udp, tcp	linux, windows	Single	no	no	yes	TCL	CLI	-
2007	Black Energy [20]	Bandwidth, Resource	DDoS	tcp, udp, icmp,http	linux	Multiple	yes	no	-	-	CLI	IRC based
2007	Hgod [20]	Bandwidth, Resource	DDoS	tcp, udp, icmp	windows	Multiple	-	no	yes	-	CLI	IRC based
2007	Panther [20]	Bandwidth	DoS	icmp, udp	-	Single	no	-	-	-	-	-
2007	RefRef [20]	Resource	DDoS	-	windows	Multiple	-	no	no	perl	CLI	IRC based
2008	LOIC [40]	Resource	DoS, DDoS	tcp, udp, icmp,http	linux,mac os,windows,android	Multiple	yes	no	no	C-Sharp	GUI	IRC based
2008	UDP Flooder [40]	Bandwidth	DoS	udp	windows	-	-	no	yes	-	GUI	IRC based
2009	DDOSIM [40]	Resource	DDoS	tcp,smtp,http,udp	linux	Multiple	yes	no	no	C++	CLI	-
2009	Slowloris [40]	Bandwidth, Resource	DoS	http	windows, linux	Single	no	no	no	Perl	GUI and CLI	-
2009	TOR's hammer [38]	Bandwidth, Resource	DoS, DDoS	http	unix, linux, macos	Multiple	yes	no	no	Python	CLI	Agent based
2010	Davoset [38]	Resource	DoS, DDoS	http	linux	Multiple	yes	no	no	Perl	CLI	-
2010	Owasp [37] Http Dos Post	Resource	DoS	http	windows	Single	no	no	no	Python	GUI	-
2010	Pyloris [40]	Resource	DoS, DDoS	tcp,imap, udp,smtp,http, ftp, telnet	linux, windows, macos	Multiple	yes	no	yes	Python	CLI	IRC based
2010	XOIC [40]	Resource	DoS, DDoS	udp, tcp, icmp	windows	Multiple	yes	no	no	C-Sharp	GUI	IRC based
2011	Aldi Botnet [2]	Resource	DDoS	http, tcp	windows	Multiple	yes	no	no	-	GUI	Web based
2011	R-U DEAD -YET [38]	Resource	DoS, DDoS	http	linux	Single	no	no	yes	Python	CLI	-
2011	SSL DoS [25]	Resource	DoS	tcp	windows, unix	Single	no	-	-	-	-	-
2012	Golden-Eye [19]	Resource	DoS	http	Linux, Windows, MAC	Single	no	no	no	Python	CLI	-
2012	HOIC [40]	Resource	DDoS	http	windows	Multiple	yes	no	no	Basic	GUI	-
2012	HULK [38]	Resource	DoS, DDoS	http	linux, windows	Single	no	no	no	Python	CLI	-
-	Silent-Ddoser [3]	Bandwidth, Resource	DDoS	udp, tcp, http	windows	Multiple	yes	yes	no	VB.net	GUI	IRC based
-	SEER [14]	Resource	DDoS	icmp, tcp, udp	windows	Multiple	yes	no	yes	java	GUI	-

It is command line based tool which has the ability to exhaust the resources of the victim server. The parameters required to launch attack are target IP address, start size and end size of the packet [20].

Crazy Pinger. Crazy Pinger is the DoS attack tool which can launch attack by sending a large volume of ICMP packets to the victim machine or to the large remote network. Crazy Pinger is the GUI based attack tool that can spoof the ip addresses and can exhaust the resource and bandwidth. This kind of tool is easy to use and is effective over the multiple platforms [20].

Kaiten. Kaiten is the DDoS attack tool which can launch multiple attacks, viz., UDP flood, TCP flood, SYN flood and PUSH+SYN flood. It uses random source IP addresses for generating botnets. The Kaiten is the command line based tool with IRC as the DDoS attack architecture model. It has the ability to deplete the resource and the bandwidth of the victim server [20].

Knight. Knight is an IRC-based tool can launch multiple DDoS attacks to create SYN flood, UDP flood and urgent pointer flood on windows machines. An IRC based tool that can destroy the resources and the bandwidth of the victim system. It is the command line interface attack tool whose implementation language is the C- language. It can make botnet also [38].

Nemsey. Nemsey is the DoS attack tool whose presence specifies the computer is insecure and infected with the malicious software. It is a GUI based attack tool that can deplete the bandwidth of the victim server. It does not generate the multiple sources and spoof the ip addresses. It attempts to launch an attack with a specified number of packets of specified sizes [38].

FSMax. FSMax is the DoS attack tool which can be used to test the stress of the network and to test the server for buffer overflows which may be exploited during attack, text file is accepted as the input which is executed through a sequence of tests based on the input. FSMax has the ability to exhaust the resources of the victim server [20].

Hping. Hping can handle the random packet size and the fragmentations. Hping performs the firewall rule testing, port scanning and protocol based network performance testing. Its implementation language is TCL and has command line interface [20].

Black Energy. Black Energy is the simple and powerful IRC based architecture model attack tool and a well-known cybercrime toolkit. This tool continues to be widely used to deny services for commercial websites and targets the critical energy infrastructure. It is command line based that can deplete the resources and bandwidth of the victim server [20].

Hgod. Hgod tool is the windows XP based tool which can spoof the source IPs and specifies protocols and the port numbers during the attack. It is used for launching TCP SYN flooding attack. The architecture model is IRC based with command line interface and has the capability to exhaust the resource and bandwidth of the victim server [20].

Panther. A UDP based DoS attack tool that can flood the specified IP at a particular port number. It takes IP address as the input parameter to launch the attack. This tool is the windows based. Panther has the ability to deplete the bandwidth of the victim server and can generate the traffic of UDP and TCP types. However, it is not so powerful attack tool [20].

RefRef. RefRef is the DDoS attack tool which is used to exploit existing SQL injection vulnerabilities. It sends the SQL malformed queries which are carrying payloads that force the servers to exploit their own resources. Its implementation language is PERL and has the command line interface. It is the attack tool that has an architecture model of IRC based model. This tool works with the perl compiler in order to launch DDoS attacks [20].

LOIC. LOIC is an open source network testing tool developed by Praetox Technologies. It was used by 4chan during Project Chanology to attack web servers. It is a GUI based DDoS attack tool which can deplete the resources of the victim server like CPU, memory etc[40].

UDP Flooder. UDP flooder is the port scanner and has the user friendly graphical user interface that can target the random ports and random packet size. It is the IRC-based attack tool which can also spoof the ip addresses of the source. It can deplete the bandwidth of the victim server in no time [40].

DDoSim. A DDoS attack tool that uses the random IP addresses to stimulate several zombies with full TCP connection. DDoSim can generate the HTTP-GET flood attack to target random IP addresses and random ports. A command line interface whose implementation language is the C^{++} and has the ability to deplete the resources of the victim server [40].

Slowloris. Slowloris attack tool creates the flood of TCP SYN requests to the target victim. During the Iranian presidential election in 2009, Slowloris was used as a prominent tool to leverage DoS attacks against sites run by the Iranian government. It has both graphical user and the command line interfaces and is implemented in the perl language [40].

Tor's Hammer. A python based slow post DoS testing tool that runs through TOR network. Tor's Hammer

uses random source IP address making difficult to trace back the source machine of the attacker. This tool that can deplete the bandwidth and resources of the victim server. It has the command line interface and its architectural model is the Agent based model [38].

Davoset. Davoset is a command line tool for conducting DDoS attacks on the sites via Abuse of functionality and XML external entities vulnerabilities at sites for attack on other sites (including DoS and DDoS attacks). Davoset is the PERL based attack tool and has the ability to make the multiple zombies generates the botnets for launching the DDoS attack [38].

Owasp DoS http post. Owasp DoS is the open web application software project for testing performance, availability and capacity planning of web application. Owasp, a graphical user interface is the Slow HTTP POST attack requests are sent to the victim and maintain SSL half connection with the victim. it has ability to deplete the resources of the victim server [37].

Pyloris. Pyloris is the script based tool and is used for testing a service level vulnerability to a particular class of Denial of Service attacks. It uses the inbuilt methods of Slowloris operating system and is used to test the server's readiness to withstand Botnet based DDoS attacks. It is written in Python and has the IRC based attack model [40].

XOIC. Xoic is a GUI based tool that can perform the DDoS attack on any server with specified IP address, a user-selected port and a user-selected protocol. It seems to be more powerful than the Loic. Its implementation language is the C-sharp and has IRC based DDoS attacks that can be performed with TCP, HTTP, UDP, ICMP packet messages [40].

Aldi Botnet. Aldi botnet is a newer inexpensive DDoS bot that is growing in the wild and is designed to deplete the resources of the victim server. Arbor company on September 30,2011 revealed that there are at least 50 distinct aldi botnets that have been seen in the wild with 44 unique command and control points [2].

R-U-dead-Yet. Rudy is the python based slow attack tool to crash the web server. It has two modes, one is the interactive menu mode and another is the unattended configuration based execution mode. A python based tool that can launch attack in order to deplete the resources of the victim server and execute over the Linux platform [38].

SSL DoS. SSl DoS is the windows based tool that can cause denial of service attack without creating the botnets. This tool can be executed on the both windows and Linux, is more effective and powerful. It can launch the network layer flood attacks. It has the ability to exhaust the resources of the victim server [25].

Golden-Eye. Golden-Eye is the multi-threaded python based attack tool that can launch the http flood attack. Attack vector exploitation can be done by HTTP keep alive + no cache messages. It does not encrypt the attack packets and doesn't not support IP spoofing. This tool can execute on the Windows, Linux, MAC operating systems and can deplete the resources of the server [19].

HOIC. High speed, multi-threaded attack tool and has the capability to flood upto 256 websites at once. HTTP GET flood and POST requests are sent to the target server. Anonymous was the first group to utilize it and launch attack against the website of the US department of justice. It has the ability to resource of the victim server [40].

Hulk. HTTP unbearable load king has ability to take down the server in a minute as it directly affects the server's load. It generates TCP SYN flood and multi-threaded HTTP GET flood requests. It can hide the actual user agent. It has the ability to send the different patterns of attack requests that can obfuscate the referrer for each request [38].

Silent ddoser. Silent ddoser can create UDP, SYN and HTTP flood requests to the target victim. It has ability to update the bots on the botnet at ongoing attack. It utilizes triple-DES, RC4 encryption and has IPV6 capabilities with password stealing function. It is a windows tool which has graphical user interface that have the IRC based model [3].

SEER. SEER generates the attack traffic by using the Flooder tool, developed by SPARTA and the Cleo tool developed by UCLA. SEER can generate the same traffic with many variations, can show the movement of traffic from one client to another means the graphical vision of the complete traffic with topology made in the deter testbed. The main disadvantage of this attack tool is, it is only used with the deter testbed. It has graphical user interface and implementation language is java [14].

A wide variety of DDoS attack tools are available on the internet. Most of them are very powerful and destructive; they can easily crash down the target network and web applications in terms of bandwidth and resource depletion in no time. Out of these attack tools Black Energy, Loic, Hoic, r-u-dead-yet and Hulk can generate legitimate looking HTTP traffic. Although Tfn, Trinoo, Stacheldraht, Shaft, Mstream and Trinity have the capabilities to launch powerful DDoS attacks but they are obsolete now a days and are not powerful enough as compared to other attack tools in the list. The parameters required to launch an attack vary with the type of attack

tool used. The year-wise comparison shows the drastic change in the technology wise features of attack tools over the past years.

3 Traffic Generators and Their Comparison

Traffic generators are the tools which can generate the legitimate traffic as well as attack traffic. This section provides the detailed comparison of traffic generators based on their key features as summarized in the Table 2.

Bit-Twist: A highly Scriptable tool which selects the specific range of the packets and then save them in another trace file. It can send multiple trace files at a time and sends the packets at the specific speed. This is Windows and Linux based tool with the feature of the command line computers. It generates the transport layer and the application layer [40].

Byte-Blower: IP testing tool that helps to quickly access the performance and the stability of the IP networks and the network equipment. This gives the real time view. Its implementation language is TCL (tool command language) and is based on the Linux, windows, MacOS [10].

Curl Loader: An Open source and flexible tool for generating and testing of load. Loader uses real HTTP,FTP and TLS/SSL protocol stacks and can simulates tens of thousands and hundreds of users with their own IP addresses. Command line based traffic generator that can run over only linux systems. A tool which can generate the traffic of the application layer and network layer [40].

D-ITG: Distributed internet traffic generator produces traffic that accurately replicates appropriate real time stochastic process by making use of both IDT(inter-departure time) and PS (packet size) features. It is capable of generating traffic at the network, transport and the application layer. It has command line interface which can run on the windows, linux and BSD. The user can generate legitimate traffic, attack traffic and flash traffic [5].

Geist: An internet traffic generator for server architecture evaluation that remains limited to the HTTP GET requests . It can generate the dynamic GET parameters and can also handle the cookies. A command line based traffic generator can generate the traffic of application layer and its implementation language is C. It runs on the windows platform [26].

Harpoon: Harpoon can generate the traffic from traces or from the high- level specification . Harpoon traffic can runs over the HTTP and application behavior that may be different from the real time traffic. A unix based generator which can generate the traffic of application layer, transport layer and the datalink layer. It is used for the testing of network switching hardware [19].

HTTP-Perf: A generator used to measure the web server performance. Its major characteristics includes the robustness (ability to generate and sustains the server overload), support for HTTP/1.1and SSL protocols and extensibility. A command line based traffic generator that can generate application layer traffic with fix number of HTTP GET requests and can be used to check the performance of the web [40].

Iperf: A multi-threaded generator in which client-server can have multiple connections. It is used for active measurements of maximum achievable bandwidth on IP networks. A java based traffic generator which is having graphical user interface as the platform. It can generate network layer and application layer traffic for measuring maximum achievable bandwidth [24].

KUTE: Kernel based traffic engine has KUTE-REC and KUTE-SND. Kute-Rec can count the received packets, inter-arrival time, measures high packet rates and Kute-Snd can generate high packet rate for software solution. A kernel based traffic engine which can generate transport layer traffic and its implementation language is the C [30].

LAN-forge-fire: It is a java based traffic generator with graphical user interface that can generate the application layer traffic against the web-server, VOIP, gateways, firewalls, and load balancers. It requires full TCP connection and provides the support for the hping and nmap [11].

M-GEN: An open source generator that can generates the real time traffic patterns. It can be used in network simulation environment like NS-2 and Opnet. MGEN supports the TCP messaging and the IPv6 networking. A command line based traffic whose implementation languages are TCL and NS-2. It can generate the application layer as well as transport layer traffic [34].

Netperf: An open source generator that can be used to measure the performance of many different types of networks . Its testing is for unidirectional throughput and end-to-end latency. A C-based command line tool which can be executed on the command line based user interface and can generate the transport layer and the network layer traffic [35].

Ostinato: An open source, cross-platform network traffic generator that can craft and send packets of several streams with different protocols at different rates. A python based traffic generator which is having graphical user interface. It can generate the traf-

Table 2: Comparison of traffic generators

Name of Generator	Implementation Language	Type of traffic generated	OS Supported	GUI/CLI	Embedded in Testbed	Input Parameters	Operating Layer	Key Features
SEER [14]	Java	TCP, UDP, HTTP, ICMP	Windows, Linux, Unix	GUI	yes	server IP, client IP, thinking time	Network, Transport, application layer	Legitimate traffic generation, DDoS traffic generation, Visualization
D-ITG [5]	C^{++}	HTTP, TCP/IP	Linux,Windows, Free BSD, OSX(Leopard)	CLI	yes	Inter Departure time, packet size Random and Variable	Transport and application layer	IPv4 traffic generation, IPv6 traffic generation
HTTPerf [40]	-	HTTP, SSL	Linux(Debian), Unix	CLI	no	No. of headers, no. of clients, timeouts, maximum no. of connections	Application Layer	Measures web servers performance, Generates fix no. of HTTP GET requests and keep track on responses by measuring response rate
Pylot [39]	Python	HTTP, HTTPS	Windows XP, Vista, Ubuntu, Cygwin, MACOS	GUI	no	No. of agents, request intervals, rampup time, test duration	Application layer	Multithreaded load generator, Real time stats, Cross platform, Custom timers, Results reports with graphs
Packmime [23]	NS	HTTP	Linux	-	no	response size, request size, flow arrive, server, client, request rate	Transport and Application layers	Simulate different RTT, Bottleneck links, Loss rates
Tmix [17]	NS-2	TCP, IP	Linux	-	Geni Testbed	Load data files,start time	Transport layer	Generate realistic traffic
Ostinato [36]	Python	TCP, ICMP, UDP	Windows,Free BSD, Linux, MACOS	GUI	no	No. of packets, stream rates, no. of streams	Network layer, Transport layer	Cross-platform network traffic generator, Can open,edit,replay,save the pcapfiles
Surge [22]	HTTP	-	-	-	no	-	Application layer	Http traffic generator
Webstone [46]	C	HTTP	WindowsNT, Solaris, UNIX	CLI	Deter testbed	no. of minimum clients, iterations and time per run	Application Layer	Distributed multipurpose benchmark, Measure the performance of the web server's hardware and software products
Geist [26]	C	HTTP	Windows	CLI	no	server, client, protocols	Application Layer	Limited to HTTP GET requests, Does not follow the HTTP redirects
RUDE [40]	C	UDP	Linux	GUI	no	servers, clients, protocols	Transport Layer	Real time UDP data emitter, Generates traffic to the network which can be received and logged on the other site of the network with CRUDE(collector for RUDE)
KUTE [30]	C	UDP	2.6 linux kernel	-	yes	count received packets, inter arrival time, test hardware driver, performance of receiving stack, router/switches	Transport Layer	Kernel based traffic engine
LAN Forge Fire [11]	Java	HTTP, HTTPS,FTP, TELNET, SFTP, TFTP	Linux,Windows, Solaris	GUI	no	Clients, source, packet information	Application Layer	Need full TCP connection, supports many impairments latency, jitter,bandwidth, packet loss, packet reordering
TCP replay [43]	C/C^{++}	TCP, IP	Unix, win32	CLI	Deter testbed	Server, Client, Port No., IP address range	Network Layer	Ability to injects previously captured traffic in libpcap format
MGEN [34]	NS-2, TCL	TCP, UDP, IP	Unix, MACOSX, Win32	CLI	no	Host address, Receive port list, Time to live, type of service, socket buffer size	Network and Transport Layer	TCP Messaging, IPv6 networking, Generates real time traffic patterns, Supports additional statistical patterns, Transport buffering message count, Payload Enhancement
Harpoon [19]	-	HTTP, UDP, TCP, IP	Unix	-	Deter testbed	No. of nodes, file sizes, thinking time, client session, server session, server port	Datalink layer, Transport,Application layer	Generate representative background traffic, Testing of network switching hardware
Bit-Twist [40]	-	TCP, UDP, IP,ARP	Windows, Linux, MACOS X	CLI	no	packet size, request rate	Network and Transport Layer	Powerful libpcap based ethernet packet generator, Complement of TCPDUMP,Capable of sending multiple trace files at a time
Curl Loader [40]	C	HTTP, HTTPS,FTP,FTPS	Linux	CLI	no	Interface, Client, IP address range	Transport and Application Layer	Simulates hundreds of thousands of HTTP/HTTPS and FTP/FTPS clients with its own ip address, FTP Passive and active.
Trafgen [19]	C	HTTP	Linux	CLI	no	Input configuration file,Outgoing traffic devices, no. of frames	Application layer	Fuzzy testing, Fast network traffic generator for debugging and performance evaluation
Netperf [35]	C	TCP, UDP, SCTP, IP	BSD, Unix, Windows, others	CLI	no	minimum interval in real seconds, buffer alignment	Network and Transport Layer	Testing for unidirectional Throughput, Testing for unidirectional end-to-end Latency
Iperf [24]	Java	TCP, UDP, SCTP	Windows, Linux, MACOS, free BSD, openBSD	GUI	no	Specific time, Buffer, target layer bandwidth, protocols	Network and Transport layer	Active measurement of maximum achievable bandwidth on IP network
Byte-Blower [40]	TCL	IP, TCP	Windows, Linux, MACOS	GUI	no	protocols, set latency parameters	Data link,Network, transport layer	IP testing, Scalability of IP network and network impact
Seagull [40]	C^{++}	TCP, UDP, IP, HTTP	Linux, Win32	CLI	no	protocols, user interface Scheduling/core	Network, transport layer and Application layer	Multiprotocol traffic generator, Multithreaded for performance and reliability, Dynamically adjustable scenario rate

fic of network layer and transport layer which is the cross-platform traffic network [36].

Packmime: A HTTP-traffic generator that was developed by the researchers in the internet traffic research group at Bell Labs. Its implementation in NS-2 is done at the UNC-chapelHill. It generates the application layer and transport layer generator. It can simulate different RTT, bottleneck links and loss rates and is publicly available [23].

Pylot: A free open source generator that generates HTTP load tests for the purpose of benchmarking and analysis. It can generate the concurrent load, verifies server responses and produces reports with metrics. A python based graphical user interface tool that is publicly available and generates the application layer traffic. It is a multithreaded load generator [39].

RUDE: Real time UDP data emitter is a flexible programs that can generates traffic and this traffic can be received and logged on the other site of the network with CRUDE(collector of RUDE). A C-based graphical user interface tool that can be used a real time UDP date emitter. It is publicly available and generates the traffic for transport layer [40].

Seagul: An open source tool which allows the additional support of a brand new protocols in less than two hours with no programming knowledge. A powerful traffic generator for functional, load, endurance, stress and performance tests for almost any kind of the protocol. A command line based traffic generator have the C^{++} as its implementation language [40].

SEER: Security experimentation environment(SEER) that developed by SPARTA Inc. It provides the user interface to the experimenters for writing scripts and performs experiments in the Deter environment. A java based traffic generator having the graphical user interface which can generate the legitimate traffic generators [14].

Surge: Scalable URL reference generator that performs reference matching, server file size distribution, request size distribution, relative file popularity, embedded file references. It is the HTTP-based traffic generator which can be used to generate the application layer traffic. It can perform the distribution of the various file [22].

TCP-Replay: It provides the means repeatable and reliable environment for testing a variety of network devices such as switches, routers and firewall networks. It is best suitable for intrusion detection and intrusion prevention systems. A command line based tool which is embedded into the deter testbed and can be used to generate the network layer traffic [43].

Tmix: A traffic generator that is embedded in the GENI platform that is capable of generating the realistic traffic. This generator requires the full TCP and one-way TCP. It can generate the transport layer traffic which is considered as the realistic traffic. It is publicly available over the internet and can execute only on the Linux platform [17].

Trafgen: A multi-threaded network traffic generator with the potential of fuzzy testing that means a packet configuration can be built with random numbers on all or certain packet offsets. It is the C-based traffic generator which is executed by using the command line interface of the linux operating system. It can be used to test the fuzzy system [19].

Webstone: A distributed multi-process benchmark that is embedded in the Deter testbed. It can be used to test the performance of the HTTP in contrast to server's platform. It can measure the average and maximum connect time and the response time. It is the application layer traffic generator which is the CLI based and its implementation language is the C language [46].

A wide variety of traffic generators are available out of which some are licensed and others are free to download. These can be differentiated by using various features like traffic generated type, implementation language, operating system used, layer wise differentiation etc, as summarized in Table 2. The attackers are very smart now a days as they use such traffic generators to generate legitimate looking traffic so as to circumvent the existing defense methods. The need of the hour is to know more and more about the technical details and trends of such attack tools and traffic generators used by the attackers. This information would surely be helpful for the researchers for designing their realtime experiments for the validation of their proposed defense methods.

4 Conclusion

DDoS attack is a severe threat that makes the Internet based web services unavailable to the legitimate users and cause huge financial losses to the communication, banking, medicine and research applications. A number of surveys and taxonomies of DDoS attack tools and traffic generators have been proposed till date but all of them lacks in one dimension or the other. The existing taxonomies have failed to provide the technical details of such tools and their usage. We have done the extensive survey of the popular DDoS attack tools and traffic generators used by the attackers to launch diversity of attacks. In this paper, we have extensively surveyed the popular DDoS attack tools and traffic generators based on the identified key features. Such an extensive survey will surely help the experimenters to hand pick an appropriate attack tool or the traffic generator for designing their real experiments.

Acknowledgments

This Research work has been supported by the All India Council for Technical Education (AICTE), New Delhi, India under Grant Research Promotion Scheme Grant No. 8023/RID/RPS-93/2011-12.

References

[1] M. Aamir and M. A. Zaidi, "DDoS attack and defense: Review of some traditional and current techniques," *CoRR abs/1401.6317*, 2014. (http://docplayer.net/678758-Ddos-attack-and-defense-review-of-some-traditional-and-current-techniques.html)

[2] Aldibotnet, *DDoS Attack Tools*, 2012. (https://asert.arbornetworks.com/ddos-tools)

[3] Arbor Networks, *Silent Ddoser,* 2015. (https://www.arbornetworks.com/blog/asert/tag/silent-ddoser/)

[4] R. Arun and S. Selvakumar, "Distributed denial-of-service (DDoS) threat in collaborative environment - A survey on ddos attack tools and traceback mechanisms," in *Proceedings of IEEE International Conference on Advance Computing*, pp. 1275–1280, 2009.

[5] A. Avallone, A. Pescape and G. Ventre, "Distributed Internet Traffic Generator (D-ITG): Analysis and experimentation over heterogeneous networks," in *International Conference on Network Protocols*, Atlanta, Georgia, 2003.

[6] S. Behal, A. S. Brar, and K. Kumar, "Signature-based botnet detection and prevention," in *Proceedings of International Symposium on Computer Engineering and Technology*, pp. 127–132, 2010.

[7] M. H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya, and J. K. Kalita, "Detecting distributed denial of service attacks: methods, tools and future directions," *The Computer Journal*, vol. 57. no. 4, pp. 537–556, 2013.

[8] A. Botta, A. Dainotti, and A. Pescape, "A tool for the generation of realistic network workload for emerging networking scenarios," *Elsevier Journal of Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.

[9] V. Bukac, *Traffic Characteristics of Common DoS Tools*, Technical Report FIMU-RS-2014-02, 2014.

[10] ByteBlower, *ByteBlower*, 2014. (https://www.excentis.com/blog/)

[11] Candela Technologies, *Lanforge Fire: Network Testing and Emulation Solutions*, 2015. (http://www.candelatech.com)

[12] G. Carl, G. Kesidis, R. Brooks, and S. Rai, "Denial-of-service attack-detection techniques," *IEEE Internet Computing*, vol. 10, no. 1, pp. 82–89, 2006.

[13] CERT, *Computer Emergency Report Team (CERT)*, 2015. (http://www.cert.org/)

[14] DeterLab, *SEER: The Security Experimentation EnviRonment*, 2012. (http://seer.deterlab.net/trac)

[15] D. Dittrich, *The DoS Project's Trinoo, Distributed Denial of Service Attack Tool*, 1999. (http://staff.washington.edu/dittrich/misc/trinoo.analysis.txt)

[16] C. Douligeris and A. Mitrokotsa, "Ddos attacks and defense mechanisms: classification and state-of-the-art," *Elsevier Journal of Computer Networks*, vol. 44, no. 5, pp. 643–666, 2004.

[17] Geni, *Tmix*, 2011. (http://groups.geni.net/geni/wiki/genitmix)

[18] S. Ghansela, "Network security: Attacks, tools and techniques," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 6, pp. 419–421, 2013.

[19] Github, *DDoS Attack Tools*, 2013. (https://github.com/)

[20] N. Hoque, M. H. Bhuyan, R. C. Baishya, D. K. Bhattacharyya, and J. K. Kalita, "Network attacks: Taxonomy, tools and systems," *International Journal of Network and Computer Applications*, vol. 40, pp. 307–324, 2014.

[21] A. Hussain, S. Schwab, S. Fahmy, J. Mirkovic, R. Thomas, "DDoS Experiment methodology," in *Proceedings of DETER Community Workshop on Cyber Security Experimentation*, pp. 8–14, 2006.

[22] ICIR, *Traffic Generators for Internet Traffic*, 2010. (http://www.icir.org/models/trafficgenerators.html)

[23] Internet Traffic Research Group, *Packmime*, 2005. (http://www.isi.edu/nsnam/ns/doc/node555.html)

[24] iPerf, *iPerf - The Network Bandwidth Measurement Tool*, 2015. (https://iperf.fr/iperf-download.php)

[25] Kali Linux Tutorials, *THC-SSL-DoS - A Denial of Service Tool Against Secure Web-servers and for Testing SSL-Renegotiation*, 2011. (https://www.thc.org/thc-ssl-dos)

[26] K. Kant, V. Tewari, and R. Iyer, "Geist: A web traffic generation tool," in *Proceedings of International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pp. 227–232, Springer, 2002.

[27] A. Kapravelos, C. Grier, N. Chachra, C. Kruegel, G. Vigna, and V. Paxson, "Hulk: Eliciting malicious behavior in browser extensions," in *Proceedings of 23rd USENIX Security Symposium*, pp. 641–654, 2014.

[28] H. Kaur, S. Behal, and K. Kumar, "Characterization and comparison of distributed denial of service attack tools," in *Proceedings of IEEE International Conference on Green Computing and Internet of Things (ICGCIoT'15)*, pp. 1139–1145, 2015.

[29] N. Kaur and S. Behal, "P2P-BDS: Peer-2-Peer botnet detection system", *IOSR Journal of Computer Engineering*, vol. 16, no. 5, pp. 28–33, 2014.

[30] Kute, *KUTE – Kernel-based Traffic Engine*, 2007. (http://caia.swin.edu.au/genius/tools/kute/)
[31] C. Liu, C. Peng, and I. Lin, "A survey of botnet architecture and batnet detection techniques," *International Journal Network Security*, vol. 16, no. 2, pp. 81–89, 2014.
[32] M. Mahmoud, M. Nir, and A. Matrawy, "A survey on botnet architectures, detection and defences.," *International Journal Network Security*, vol. 17, no. 3, pp. 264–281, 2015.
[33] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *International Journal ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.
[34] U.S. Naval Research Lab, *Networks & Communications Systems Downloads*, 2015. (http://downloads.pf.itd.nrl.navy.mil/mgen)
[35] Netperf, *Netperf Homepage*, 2015. (http://www.netperf.org/netperf)
[36] Ostinato, *Network Traffic Generator and Analyzer*, 2010. (http://ostinato.org/)
[37] OWASP, *Open Web Application Security Project*, 2014. (https://www.owasp.org/)
[38] Packet Storm, *DDoS Attack Tools*, 2015. (http://packetstormsecurity.org)
[39] Pylot, *Pylot - Web Performance Tool*, 2007. (http://www.pylot.org/)
[40] Sourceforge, *DDoS Attack Tools*, 2012. (http://sourceforge.net/projects)
[41] S. M. Specht and R. Lee, *Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures*, pp. 543–550, ISCA PDCS, 2004.
[42] A. Srivastava, B. B. Gupta, A. Tyagi, A. Sharma, and A. Mishra, "A recent survey on ddos attacks and defense mechanisms," in *Advances in Parallel Distributed Computing*, pp. 570–580, Springer, 2011.
[43] Tcpreplay, *Tcpreplay*, 2014. (http://tcpreplay.synfin.net)
[44] M. Uma and G. Padmavathi, "A survey on various cyber attacks and their classification.," *International Journal Network Security*, vol. 15, no. 5, pp. 390–396, 2013.
[45] F. Wang, H. Wang, X. Wang, and J. Su, "A new multistage approach to detect subtle ddos attacks," *Joournal of Mathematical and Computer Modelling*, vol. 55, no. 1, pp. 198–213, 2012.
[46] WebStone, *The Benchmark for Web Servers*, 2002. (http://www.mindcraft.com/webstone/)

Biography

Sunny Behal has done Bachelor of Technology in Computer Science and Engineering from SBS State Technical Campus, Ferozepur, Punjab, India in 2002. He finished his Masters in Computer Science and Engineering from Guru Nanak Dev Engineering College, Ludhiana, Punjab, India in 2010. His Research interests includes Botnets, DDoS attacks, Information and Network Security. Currently, He is full time Ph.D. Research Scholar at SBS State Technical Campus, Ferozepur and have published more than 40 Research papers in different International Journals and Conferences of repute.

Krishan Kumar has done Bachelor of Technology in Computer Science and Engineering from National Institute of Technology, Hamirpur in 1995. He finished his Masters in Software Systems from BITS Pilani in 2001. He finished his Ph. D. from Department of Electronics and Computer Engineering at Indian Institute of Technology, Roorkee in 2008. Currently, he is working as Associate Professor at SBS State Technical Campus, Ferozepur, Punjab, India. His general research Interests are in the areas of Information Security and Computer Networks. He has published around 200 + research papers in different International Journals and Conferences of Repute including more than 500 citations.

Multi-objective Optimization for Computer Security and Privacy

Seyed Mahmood Hashemi, Jingsha He and Alireza Ebrahimi Basabi
(Corresponding author: Seyed Mahmood Hashemi)
School of Software Engineering, Beijing University of Technology (BJUT)
Beijing Engineering Research Center for IoT Software and Systems
100 Ping Le Yuan, Chaoyang District, Beijing 100124, China
(Email: Hashei2138@yahoo.com)
(Received Feb. 20, 2016; revised and accepted Apr. 12 & May 7, 2016)

Abstract

There is need for a scheme to ensure the security and privacy of the administrator and users. Providing security is different to providing privacy, because their goals differ. Security is based on organization goals, but privacy is based on user goals. Providing security and privacy must be according to organization constraints. The most important constraints in any organization are economic issues. A useful scheme must consider all these requirements. In this paper, we present a scheme that provides security and privacy and considers various constraints. We model the problem as a multi-objective optimization problem.

Keywords: Multi-objective distributed constraint optimization problem, multi-objective optimization, PGP, single-objective distributed constraint problem

1 Introduction

Obviously, security is an important issue for any system. High levels of security for Internet communications usually restrict access to data to accredited individuals or organizations. Paradoxically, these security measures, themselves, invade personal privacy and organizational needs for confidentiality.

The main concepts for security are Confidentiality, Integrity and Availability (CIA). Privacy means protection of personal information. Of course, the official meaning of security and privacy are similar, but if we note the origins of each, we can understand the difference between them. Security is a high priority of an organization. In other words, governing bodies want to keep their information in a secure state, but privacy is an expectation of users. Protection of personal information (such as personal pages or mail, etc.) is attractive for users.

On the other hand, the major advantage of networking is user productivity. Thus any network system must provide access to assets for users. User productivity and costs of the implementation of a system are two points that play a major role in the success of a system in the real world. The objective of this paper is to provide a scheme for system developers with optimum levels of security, privacy, user productivity and cost.

In this paper, we present a Multi-objective Distributed Constraint Optimization Problem (MO-DCOP), which is an extension of the Single-Objective Distributed Constraint Problem (DCOP). In MO-DCOP, different aspects of a distributed system are optimized simultaneously. The presented model is decentralized, so there is not any agent needed to maintain information.

The rest of this paper is organized as follows: Section 2 is assigned to related works. In Section 3, we describe four concepts that are used in this paper: polling system, business intelligence, fuzzy systems and multi-objective optimization. Section 4 defines our problem. In Section 5, we present our proposed algorithm. Experimental results are presented in Section 6 and finally Section 7 presents the conclusion.

2 Related Works

The evolution of the current industrial context and the increase of competition pressure, has led companies to adopt new concepts of management [2]. The implementation of the most important part of the plan phase, consisting of the definition of an appropriate global management plan QSE (Quality, Security and Environment) has been proposed [3]. This implementation is based on the multi-objective influence diagrams (MIDs) [30]. The proposed approach has three phases: Plan phase, Do phase and Check & Art phase. The first phase gathers all quality, security and environmental objectives issued from the requirements, and then analyzes them. In this phase we can define a global management QSE plan. The second phase has the input of the global management plan QSE

and the corresponding global monitoring plan generated from the plan phase and will also implement the selected treatments. In the third phase, finalization of the process of integration occurs through measuring the effectiveness of different decisions. Neubauer et al. provide a structured and repeatable process that includes: defining evaluation criteria according to corporate requirements, strategy, assessing and/or refining the existing IT security infrastructure, identifying stakeholder preferences (risks, boundaries), determining the solution space of all efficient (Pareto optimal) safeguard portfolios, and interactively selecting the individually "best" safeguard portfolio [32]. This paper tries to combine different benefits and costs into one formula. This presents a problem because the authors do not present a multi-objective optimization problem. Kumar et al. focus on PGP (pretty good privacy) [26], which was shown by Zimmerman in 1991 to provide security with available cryptographic algorithms [37]. Algorithms are chosen according to the user requirements of time, cost and required security level. Kumar et al. answer the question: How do you choose appropriate algorithms, from the available pool, to suit the user requirements of time, cost and security? They assign a security level to an algorithm according to its performance [39] investigate security models, which consider risk assessment approaches to be applied for threat modelling, network hardening and risk analysis. Overall, security models can be classified based on the methodologies used to optimally invest into computer security. We have specified the following:

- Risk assessment models;
- Cost-benefit models;
- Game models;
- Multi-objective decision support models.

Cost-benefit analysis looks into intangible costs/returns and addresses the perspective of time. The simplicity of the frameworks can give suitable investment solutions for low risk investments. However, these methods do not consider uncertainty and give misleading indications for long-term investments. In [40], the risk assessment involves a calculation of risk in relation to financial returns, rather than the defined risk of possible losses related to degradation of information security. They demonstrate a novel approach of selecting security countermeasures with respect to both investment cost and the risk of possible degradation of CIA. Their security countermeasure is represented as a binary value. Also, they thought "security solutions can be classified based on the function they provide". The main challenge Information System (IS) managers face is to strike an appropriate balance between risk exposure and the opportunity to mitigate risk through investments in security. Thus, the authors of [23] propose a decision analytical approach, but the paper does not present a formula for multi-objective optimization. Service provisioning (SP) is defined as the set of interrelated decisions in order to select a service (by a server) to attend to a request (by a client). In [34], the results of the author's case study provides evidence in support of the notion that the use of imitation (recall) in DPSP's (dynamic provider of service provision) cipher selection process reduces its overheads dramatically. In paper [33], the authors introduce a novel presentation for cyber security problems using the formalization of a Multi-objective Distributed Constraint Optimization Problem (MO-DCOP). An MO-DCOP is the extension of a mono-objective Distributed Constraint Optimization Problem (DCOP) which is a fundamental problem that can formalize various applications related to multi-agent cooperation. They develop a novel algorithm called Branch and Bound search algorithm (BnB) for solving a cyber security problem. This algorithm utilizes the well-known and widely used branch and bound technique and depth-first search strategy and finds all trade-off solutions. The purpose of any risk analysis is providing decision makers with the best possible information about the probability of loss [5]. Behnia et al. compare several different approaches for risk analysis and declare the weakness and strength for each of them.

3 Preliminaries

Our scheme has different parts and we need to have specific science to solve the problem in each part. In this part, we describe things which are needed for the proposed algorithm.

3.1 Polling System

There is need for a system which can satisfy all stakeholders' opinion. We say stakeholder for anyone who has any role in developing a team. The polling system is an appropriate method for keeping votes [20].

The polling system consists of a source for the service and a number of queues for clients with a policy for assigning service to the client. For example, in Figure 1, $\lambda 1$, $\lambda_2, \cdots, \lambda_N$ are clients and $S_1, S_2, \cdots, S_N$ are assignment policies.

The polling system can be a system with time sharing and N terminals. In that system, the central computer votes to terminals based on their requirements for data. Data transfers from terminals to the central computer through a voting scheme. Default:

1) Processes enter into the queues with a Poisson distribution;
2) Clients are served during the time as a random variable;
3) After servicing to a queue, the server assigns to other queues with a switch-over time [27].

Common polling systems are [24, 43]:

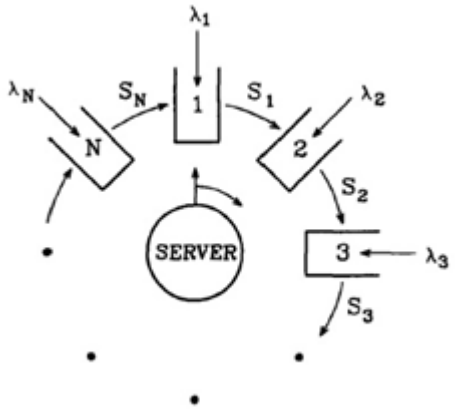


Figure 1: Polling system

Exhausted: Server is assigned to a queue for all clients in that queue.

Gated: Server is assigned to a queue with a specific time range.

Limited-1: Predefined clients which can give server.

Polling system has various applications.

Token Ring Networks: In a cyclic net, terminals need acceptance of the central computer [6].

Robotic Systems: A robotic system consists of a central robot and various inputs. For modelling this system, we can use the polling system where the robot is the server and the inputs are clients. Clients are set in queues based on their types.

Various Non-generic Computers and Communication Systems: In these systems, one processor serves to a particular type of task. A common way is to accumulate tasks into different types. In the model, tasks are clients and the processor is the server [28, 42].

Transportation (Automated Guide Vehicle): In these models, many vehicles must be carried in a narrow way. The polling system consists of automated vehicles with default paths. In this model, transportation transforms clients from various queues to specific destinations [17].

Stochastic Economic Lot Scheduling Problem (SELSP): This application is about producing by using a machine with limited capacity where the requirements produce stochastic [12, 13].

Health Care: An emergency in a hospital can be modelled with a polling system. Tasks are set in queues with an unlimited buffer.

Random Polling: The best examples for these models are distributed control systems. There is no central control, so deciding about the next terminal is done with polling.

There are some notes in the polling systems: stability, priority, structure for polling, definition of limitations and waiting time.

3.2 Business Intelligence

In the above part, we talked about the Polling System which allows us to utilize all stakeholders' opinions to improve the security of system, but on the other hand, there are users that want privacy. The balance between security and privacy is a challenging discussion. First of all, there is a need for a system which determines what level of privacy is needed [21].

Business Intelligence (BI) is a set of disciplines that include extracted data, combinations of data, the analysis and knowledge discovered which enables the system to comprehend the input/output environment [7, 9]. The aim of BI is to prepare a document for verification by the system, a prototype for deployment and obtaining a strategic and applicable knowledge base from a scientific view [44].

BI has five layers as shown in Figure 2. Also, BI helps developers to [16]:

Fast data processing: BI can access, select and modify any time. The speed is guaranteed.

Intelligent correlation analysis: BI uses mathematical models and declares scientific rules.

Multi-dimensional analysis: BI gets a combinational analysis in the format of products, brands and K, then constructs a multi-dimensional data structure.

BI uses various intelligent tools such as: tools to gather implicational data and extract business knowledge [8], and competing intelligent tools which try to get data from competing environments [9]. Constructing a BI system includes the following steps [14]:

1) Planning and direction;
2) Gathering released information;
3) Gathering resources from users;
4) Analysis;
5) Report and inform.

There are two taxonomy types in BI: one-dimensional data and two-dimensional data [29]. A list of results from hyperlinks belongs to one-dimensional data. Tree data and networking data belong to two-dimensional data. Two-dimensional data allows people to search based on human abilities.

Special text is displayed in the BI system within three phases [36]. Firstly, according to users' attractive, necessary characteristics of text. This phase can be named

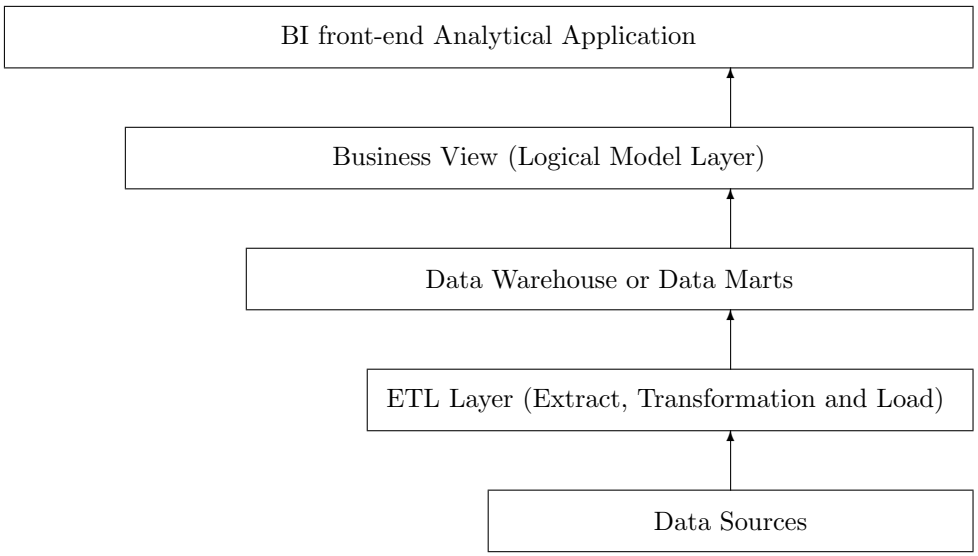


Figure 2: Layers of BI

'Analysis'. In this phase, some search techniques for the analysis of text in the network are used. These techniques have the responsibility to discover resources and patterns from the network [11]. In the second phase, which is named 'Algorithm', an applicable and flexible structure with clustering is constructed. Algorithms can be divided into two categories [22]: 1) hierarchical and 2) partitional. The final phase is 'visualization' where data is displayed to users. Visualization means display of coding data in a special format as understanding with human eyes.

3.3 Fuzzy Systems

Fuzzy systems are knowledge-based systems or rule-based systems [41]. A fuzzy system consists of a number of rules. Each rule relates input(s) to output(s). Input(s) and output(s) in fuzzy systems are recognized in fuzzy sets. Let a system with uncertainty have the input-output relation $y = f_s(x)$, where $y \in R$, and $y \in R^{nX}$. A fuzzy system represents the knowledge related to inputs and outputs by nC fuzzy rules $R_1, \cdots, R_C$ which are expressed in the form

$$R_i: \text{ If } (x_{k,1} isr A_{i,1}) \text{ and } (x_k, nX) \text{ then } (y^*_{k,i} isr B_i), \quad (1)$$

where $y_k = f_s(x_k)$ is an observation vector (x_k, y_k) of the system; $x_{k,j}$ is the j^{th} variable of x_k; $A_{i,j}$ is the membership function of the fuzzy set for the j^{th} variable in the i^{th} rule, which determines a fuzzy number for the j^{th} variable of input space; $y^*_{k,i}$ is the estimate of $y_k = f_s(x_k)$ by R_i; The operator "and" denotes the t-norm operation between two membership values; and "isr" denotes the belonging of an object into a fuzzy set. An important contribution of fuzzy systems theory is that it provides a systematic procedure for transforming a knowledge base into a non-linear mapping.

The objective of a non-linear mapping is producing output(s) with input(s). Mapping is done when there is a relation. Producing a relation (formula) from rules is the role of the Inference Engine. Researchers have proposed many inference engines and each of them has their own features (strengths/weaknesses).

A fuzzy system has two advantages. First, we can combine different votes or opinions with the formula of the inference engine. Second, the use of multiple fuzzy sets allows the proposed algorithm to be strong against changes.

3.4 Multi-objective Optimization

MOO is necessary when multiple cost functions are considered in the same problem. The aim of MOO is tuning the decision variables to satisfy all objective functions F_i to an optimum value. This class of problems is modelled by Equation (2).

$$\begin{aligned} &\text{Optimize } [F_1(X), \cdots, F_k(X)] \\ &\text{Subject to } g_i(X) \leq 0, h_j(X) = 0; \\ &\quad i = 1, \cdots, m; j = 1, \cdots, p \end{aligned} \quad (2)$$

where k is the number of objective functions; X is the decision vector; m is the number of inequality constraints and p is the number of equality constraints.

This goal causes a difference between these algorithms and their ancestor Single-Objective Optimization, which is based on the concept of best, while the multi-objective optimization uses the concept of dominance. Dominance is defined in [10]:

$$\begin{aligned} &\overrightarrow{U} = (u_1, \cdots, u_k) \prec \overrightarrow{V} = (v_1, \cdots, v_k) \\ &iff \ \ \forall i\{1, \cdots, k\} u_i \leq v_i, \exists j\{1, \cdots, k\} u_j < v_j. \end{aligned} \quad (3)$$

In words, a vector $\overrightarrow{U}$ of variables dominates another vector of variables $\overrightarrow{V}$ if and only if $\overrightarrow{U}$ can reach an op-

timal value for some criteria without causing a simultaneous non-optimal value for at least one criterion. If two vectors cannot dominate each other, they are called non-dominated vectors.

4 Problem

The phenomenon of networks has advantages and also disadvantages for our life. The main advantage of a network is User Productivity. A network provides access to information for users. On the other hand, there are two concepts which are related to the Free Flow of Information. Users expect Privacy. Governments want Security. Both privacy and security are against the free flow of information. Moreover, producing any module in a software system produces a cost. There are four significant criteria for network systems: user productivity, privacy, security and economics. We think any network system will fail without considering the follow threads:

- Any network system which does not have User Productivity will not be welcome by users.
- Users are interested in systems which protect their information.
- Providing confidentiality, integrity and availability is important for any organization.
- Financial resources of any organization are limited, so they cannot support any software.

There needs to be a system with different components. Since the system is designed for a distributed environment, components can be done in separated places and communicate to each other in general, but we show this in one figure for simplicity.

There is a directional and acyclic graph $G = (V, E)$, where V is a set of nodes that represents computer systems and E is the set of edges that represent the connection between nodes (Table 1). This graph is a popular approach to model the network. Let there be 10 nodes as in Table 1 which represents the graph, where 0 represents that there is no connection between two nodes (in column and row) and 1 vice versa.

Each edge has a number of features:

Security: represents the degree of security. All members of the developing team vote to all paths according their experience of security.

Privacy: represents the degree of privacy. Users, based on their experiences recognize the degree of privacy.

User productivity: represents how much users can have access to their necessary information.

Cost: represents how much cost is needed for creating and maintenance a connection. There are 5 fuzzy sets to recognize the cost (See Figure 3).

Table 1: Graph for model of the network

0	1	1	1	0	0	1	1	1	0
1	0	1	0	0	1	1	0	0	0
1	0	0	0	1	1	0	0	1	1
0	1	1	0	1	0	0	1	1	0
0	1	0	1	0	1	1	0	0	1
0	0	0	0	1	0	1	0	1	1
1	0	1	1	0	1	0	1	1	0
1	0	1	1	0	1	1	0	0	1
0	1	0	0	1	0	1	1	0	1
1	1	0	0	1	0	0	1	1	0

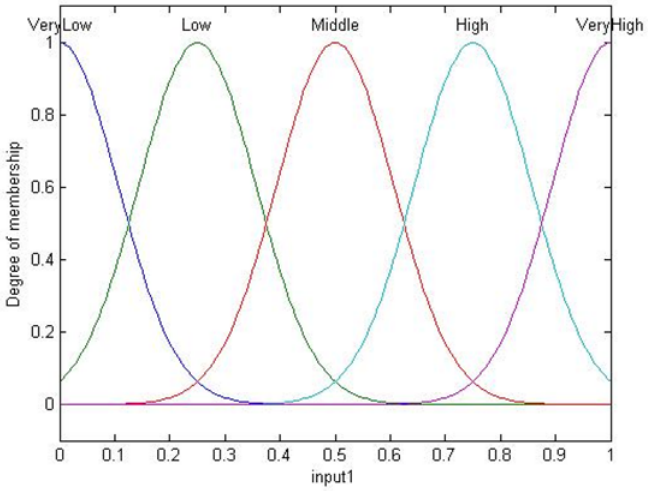


Figure 3: Fuzzy sets for cost

The main goal of this paper is to present an approach that finds an optimum path (which is created with the connection of a number of paths). Since there are many objectives at issue, we have to use multi-objectives or multi-task methods. Our research must have the following characteristics:

- Research must be based on a probabilistic view, because the behavior of the network (environment) is not predictable.
- Research must follow a bottom-up approach. This approach recognizes that the lack of security is the result of the interaction of complex nodes. In contrast, a top-down approach focuses on the whole of the system. A bottom-up approach is better than a top-down approach for presenting attack, because attack is performed on the interaction between nodes and not the whole of the system.
- Research must be based on an analysis of threat sources. Schneier states that the term "security" is meaningless if the question "secure from whom?" is not addressed [35].
- Research must be based on the grouping decision. The opinion of all corporate team members in developing the final decision needs to be sought.

This research has two major elements. One of them is a Polling System and another one is a Business Intelligence System. The polling system allows us to combine all stakeholders' opinion, so security will be became the responsibility of all system stakeholders and not only the administrator. However, providing privacy needs recognition of requirements. Actually the best way for recognition is indirection. Business Intelligence is a suitable approach to recognize user requirements.

5 Proposed Algorithm

In the proposed scheme we must consider all notices which are mentioned in the "problem" section. Firstly, there is need for a system to counter security. We propose a Polling System for this goal (See Figure 4). In our polling system, queues are used as a number for the types of stakeholders. Each stakeholder can vote in its queue. Since the stakeholder is a role, someone may vote in several queues. The core of the polling system is a Fuzzy System. We use a Product Inference Engine to combine the votes from queues,

$$f(x) = \frac{\sum_{l=1}^{M} \bar{y}^l (\prod_{(l=1)}^{n} \mu_{A_i^l}(x_i))}{\sum_{l=1}^{M} (\prod_{(l=1)}^{n} \mu_{A_i^l}(x_i))} \quad (4)$$

where $f(x)$ is output; M the number of rules; n the number of rules incipience and μ is the membership degree for input variable x in the fuzzy set A. In the proposed system, the input variable is the vote and the rule is a queue. The output is the final result from the polling system.

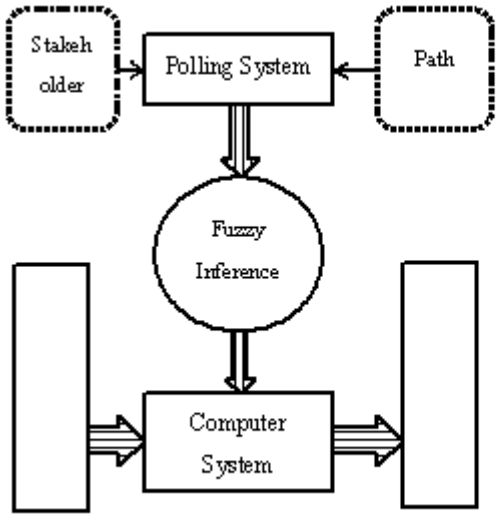


Figure 4: Proposed polling system

Secondly, there is need for a system to counter user productivity. We propose a Business Intelligence (BI) system for this goal. BI allows us to know the leaning of each user indirectly. Actually, the performance of BI is deeply dependent on the approach of information gathering. BI produces a number, with its fuzzy system as a result.

Finally, there is need for a multi-objective optimization formula to model the whole of the scheme. We propose the following optimization model for our problem:

$$\begin{aligned} &\text{Optimize } S, P, UP, C \\ &\text{Subject to: } C \leq LP, S \geq DS, P \geq DP, \end{aligned} \quad (5)$$

where S is security; P is privacy; UP is user productivity and C represents the total cost. Total cost must be less than or equal to the limited cost. Security and privacy must be greater than or equal to the default level of security and the default level of privacy, respectively. In other words, the model formula says that all parameters of a computer system (security, privacy, user productivity and cost) must be optimized simultaneously. Optimization means maximization of security, privacy and user productivity and minimization of cost. In this paper, we use tree algorithms to solve the multi-objective Problem (5):

1) Multi-objective simulated annealing;
2) Multi-objective genetic algorithm;
3) Multi-objective bee colony algorithm.

5.1 Multi-objective Simulated Annealing (AMOSA)

The basic concept in simulated annealing is the evolution of the solution by simulating decreasing temperature (tmp) in the material, where a higher temperature

denotes greater modification of the solution in a generation. If the temperature of a hot material decreases very quickly, its internal structure may change and the material could become hard and brittle. Decreasing the temperature slowly yields higher homogeneity and less brittle material. Evolution of the solution occurs at specific temperature profiles. In the first few iterations, a diverse set of initial solutions for the problem are produced at a higher temperature. These solutions are then evolved while the temperature decreases to obtain their local optima. In a multi-objective situation, there are non-dominated solutions that must be kept in the archive as candidates for the optimal solution.

AMOSA was proposed in [4]. During the execution of the AMOSA algorithm, two solutions exist: the current-so and new-so. Comparison of the two solutions yields one of three states: 1) current-so dominates new-so, 2) current-so and new-so are non-dominated with respect to each other, and 3) new-so dominates current-so.

If new-so is dominated by current-so, there may be solutions in the archive that dominate new-so. New-so is accepted into the archive based on the probability:

$$p = \frac{1}{1 + exp(\triangle \times tmp)} \tag{6}$$

where $\triangle$ is the difference between new-so and the other solutions that dominate new-so. If there are A solutions in the archive,

$$\triangle = \frac{\sum_{i=1}^{A} \triangle_i + \triangle}{A + 1} \tag{7}$$

Solutions can escape from the local optima and reach the neighborhood of the global optima by this probable acceptance. If the new-so is dominated by some solutions in the archive, Equation (7) is modified to:

$$\triangle = \frac{\sum_{i=1}^{A} \triangle_i}{A} \tag{8}$$

If the new-so is not dominated by any of the members in the archive, it is set to the current-so and is added to the archive. If the new-so dominates some solutions in the archive, it is set to the current-so and is added to the archive. In addition, any solutions in the archive that are dominated by the new-so, are removed. If the new-so is dominated by some solutions in the archive, Equation (6) is changed to:

$$p = \frac{1}{1 + exp(-\triangle)} \tag{9}$$

where $\triangle$ is the minimum difference between the new-so and the dominating solutions in the archive. The new-so is set to the current-so with Probability (9). If the new-so is not dominated by any of the solutions in the archive, it is set to the current-so and added to the archive. If the new-so dominates some solutions in the archive, it is set to the current-so and added to the archive, while all dominated solutions are removed from the archive.

5.2 Multi-objective Genetic Algorithm (MOGA)

The MOGA is based on a single-objective genetic algorithm [15, 25, 19], and comprises various stages. In the first stage, a population of individuals (chromosomes) is created. The number of individuals in the population (pop-size) is determined by the programmer. Each individual contains certain fields, where the number of fields in an individual is equal to the number of variables in the problem, which must be optimum. Each individual has the potential to reach an optimum point, at which optimal values are set in the corresponding fields in the individual. In the first stage of MOGA, all individuals in the population are initialized with random values. The algorithm runs until the stopping conditions are met. There are three types of stopping conditions. The first of these is special values; when the values of individuals are equal to the default values, the algorithm terminates. The second type of stopping condition occurs when the values of individuals no longer change. The last type of stopping condition is the number of iterations. When the number of iterations of the algorithm reaches the given threshold value (max-generation), the algorithm terminates.

Given that MOGA is an evolutionary algorithm, it is executed for a number of iterations, where each iteration of MOGA is called a generation, inspired by Darwinian evolutionary theory. The programmer can control the evolutionary nature of MOGA using the number of generations. This means that despite the deterministic optimization method, which is controlled by the number of inputs, the programmer can vary the number of generations. In the first generation, individuals are initialized with random values. The values of individuals are changed in each generation using two operators: mutation and cross-over. In mutation, one field of an individual is changed to a different value. There are a number of different methods for mutation, which describe the quality of the altered values. In cross-over, two individuals are combined to produce a new individual. After the genetic algorithm operators (mutation and cross-over) have been applied, several individuals are selected for the next generation. Selection is done stochastically according to the fitness of the individual.

The goal of the optimization algorithm is to find the optimal point. Optimal points can be divided into two categories: local optima and global optima. A local optimum can be any point that is the optimum of all points within a limited range, while a global optimum is a point that is the optimum of all points in an unlimited range. Because deterministic optimization methods compare the current point with points in a limited range, they may be trapped in a local optimum. The stochastic feature of MOGA allows the algorithm to escape from local optima and achieve the global optimum.

Based on the discussion above, MOGA has two advantages: the programmer can control the execution time and the algorithm has the potential to achieve a global

optimum point.

MOGA finds an optimum point according to the Pareto set; in other words, a point is optimum if it is not dominated by other points. Indeed, the Pareto principle allows a number of objectives to become optimum simultaneously. Each individual is checked for its domination in the population. Individual i is allocated a rank equal to one plus the number of individuals, n_i, dominating individual i. Once ranking has been completed, a raw fitness is assigned to each individual based on its rank using a linear mapping function.

$$F_i = N - \sum_{k=1}^{r_i-1} \mu(k) - 0.5(\mu(r_i - 1)) \tag{10}$$

where μ denotes the numbers of individuals in the rank. MOGA incorporates niching among individuals in each rank. The niche count with σ_{share} is found first. The distance metric is computed with the objective function values. Thus, the normalized distance between any two individuals i and j in a particular rank is calculated as:

$$d_{ij} = \sqrt{\sum_{k=1}^{M} (\frac{f_k^{(i)} - f_k^{(j)}}{f_k^{max} - f_k^{min}})^2} \tag{11}$$

The distance is computed for each pair of individuals. Therefore, the niche count is calculated by summing the shared function values:

$$SH(d_{ij}) = \begin{cases} 1 - \frac{d_{ij}}{\sigma_{share}}, & \text{if } d_{ij} < \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

The shared fitness is calculated as $F_i' = F_i / nc_i$, $nc_i = \sum d_{ij}$. Shared fitness is used as a basis for stochastically selecting individuals for the next generation. The above process continues until the stopping condition is satisfied. When the algorithm terminates, the remaining individuals represent the optimum.

5.3 Multi-objective Bee Colony (MOBC)

The foraging behavior of bees is characterized by various steps that are used in optimization. The first step is called the Waggle Dance, which is used by bees to convey information to other bees about the direction, distance, and quality of a food source. Upon finding a food source, a bee begins to dance in a figure of eight pattern. The second step in the foraging behavior is when follower bees that were waiting inside the hive, follow the dancer bee. The number of follower bees assigned to a path is directly proportional to the quality of the path. In the third step, these bees return to the hive. More bees are recruited to the source of the food if the path is still good enough. Bees stop collecting poor-quality food and adjust their strategy for finding food based on information about the location of good-quality food.

Foraging behavior can be used for optimization when it is divided into two phases. The first phase consists of path construction. In this phase, a bee explores the entire food source, but with the exploration limited by constraints. When a bee does a tour (which includes all possible variables), it performs the Waggle Dance. Other bees use this information, expressed as:

$$Pf_i = \frac{1}{L_i} \tag{13}$$

where Pf_i is the profitability of a bee_i and L_i is its tour. If a colony has n bees, the bee colony's average profitability is given by:

$$\begin{aligned} Pf_{colony} &= \frac{1}{n} \sum_{i=1}^{n} Pf_i \\ &= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{L_i} \end{aligned} \tag{14}$$

The dance duration of any bee is given by:

$$D_i = K \times \frac{Pf_i}{Pf_{colony}} \tag{15}$$

where K is the profitability rating and is adjusted according to the lookup table given in Table 2.

Table 2: Lookup table for adjusting profitability

Profitability Rating	K_i
$Pf_i < 0.9Pf_{colony}$	0.60
$0.9Pf_{colony} < Pf_i < 0.95Pf_{colony}$	0.20
$0.95Pf_{colony} < Pf_i < 1.15Pf_{colony}$	0.02
$1.15Pf_{colony} < Pf_i$	0.00

The second phase of the bee algorithm consists of path reconstruction. In this phase, bees in the hive, having received information from the explorer bee, utilize the path. Bees use a transition rule for choosing the appropriate path with the probability denoted by $P_{ij}(t)$, which measures the possibility of moving from $step_i$ to $step_j$ at time t. In a multi-objective sense, the discussed path must be examined for dominance over other paths. Formula (9) takes into consideration the fitness of all paths:

$$\rho_{ij}(t) = \begin{cases} \lambda, & j \in F_i(t) \\ \frac{1-\lambda|F_i(t) \cap A_i(t)|}{|A_i(t)| - |F_i(t) \cap A_i(t)|}, & j \notin F_i(t) \end{cases} \tag{16}$$

where λ is the value (less than one) assigned to the preferred path; $|A_i(t)|$ is the number of allowed next steps, and $|F_i(t) \cap A_i(t)|$ is the number of preferred next steps [1, 18, 31, 38].

Now, we can examine the dominance of all paths, after which each path is classified as conforming to one of three situations: 1) dominates another path(s); 2) is dominated by another path, and 3) is not dominated by any other path.

Table 3: Properties of network

-	0.3, 0.2, 0.5, 0.8	0.4, 0.4, 0.5, 0.8	0.6, 0.7, 0.5, 0.8	-	-	0.1, 0.8, 0.6, 0.7	0.9, 0.7, 0.9, 0	0.2, 0.2, 0.4, 0.5	-
0.8, 0.8, 0.7, 0.9	-	0.3, 0.5, 0.5, 0.6	-	-	0.7, 0.6, 0.5, 0.4	0.7, 0.8, 0.4, 0.2	-	-	-
0.4, 0.3, 0.5, 0.7	-	-	-	0.9, 0.1, 0.8, 0.2	0.2, 0.3, 0.4, 0.5	-	-	0.4, 0.5, 0.7, 0.9	0.3, 0.6, 0.6, 0.8
-	0.7, 0.4, 0.9, 0.8	0.9, 0.3, 0.7, 0.8	-	0.8, 0.7, 0.6, 0.5	-	-	0.6, 0.9, 0.7, 0.3	0.8, 0.2, 0.1, 0.1	-
-	0.8, 0.2, 0.8, 0.3	-	0.9, 0.7, 0.6, 0.8	-	0.9, 0.1, 0.7, 0.3	0.6, 0.7, 0.9, 0.4	-	-	0.8, 0.8, 0.5, 0.5
-	-	-	-	0.7, 0.8, 0.8, 0.9	-	0.9, 0.2, 0.3, 0.1	-	0.8, 0.9, 0.8, 0.8	0.9, 0.4, 0.3, 0.9
0.2, 0.3, 0.3, 0.3	-	0.4, 0.4, 0.5, 0.3	0.8, 0.4, 0.7, 0.6	-	0.7, 0.8, 0.9, 0.1	-	0.4, 0.2, 0.9, 0.7	0.3, 0.5, 0.6, 0.8	-
0.8, 0.4, 0.2, 0.7	-	0.2, 0.5, 0.8, 0.4	0.8, 0.4, 0.3, 0.1	-	0.6, 0.4, 0.8, 0.3	0.7, 0.5, 0.3, 0.2	-	-	0.9, 0.6, 0.4, 0.2
-	0.2, 0.6, 0.7, 0.7	-	-	0.3, 0.6, 0.7, 0.4	-	0.7, 0.5, 0.7, 0.9	0.8, 0.4, 0.3, 0.2	-	0.1, 0.4, 0.8, 0.2
0.9, 0.8, 0.4, 0.3	0.1, 0.4, 0.8, 0.3	-	-	0.5, 0.7, 0.7, 0.7	-	-	0.9, 0.7, 0.9, 0.8	0.6, 0.7, 0.3, 0.8	-

In the first situation, the path is stored in the archive. In the second situation, the path is destroyed, and in the third situation, the path is stored in the archive with the following probability:

$$P_{ij}(t) = \frac{[\rho_{ij}(t)]^{\alpha} \times [\frac{1}{d_{ij}}]^{\beta}}{\sum_{j \in A_i(t)} [\rho_{ij}(t)]^{\alpha} \times [\frac{1}{d_{ij}}]^{\beta}} \quad (17)$$

where d_{ij} is the distance between $step_i$ and $step_j$, α is a variable that influences the fitness, and β is a variable that influences the distance. A is a collection of all steps that can be reached from the previous step.

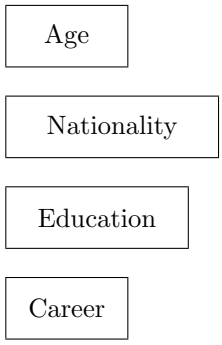


Figure 5: Proposed EI

6 Experimental Results

Let security and cost be set in [0,1] for each path. The value of security is determined by using a polling system, which allows stakeholders to give their opinions. The final result of the polling system is entered into a fuzzy system. In the fuzzy system the final value is produced with Equation (4).

However, there are two measures: User Productivity and Privacy. Both of them are related to the whole of the system (not for each selected path) and are set in [0,1]. User Productivity is determined with a Business Intelligence (BI) system. The performance of (BI) is fully dependent on indirect questions. The responsibility of BI is to mine the favorites of users from indirect questions. We focus on "indirect", because users do not usually like to state their favorite. Since this paper is just a proposal scheme, we prefer only four questions.

Suppose the qualities of system on the paths in Table 1, are shown in the following matrix (Table 3). Values in each cell represent "Security", "Privacy", "User Productivity" and "Cost" respectively.

We assume the range of values is [0,1]. This assumption does not limit the generalization. Suppose we want send a packet from node "1" to node "10" and $DS = 2$, $DP = 2$. The optimum values for this transmission based on the optimization algorithm, are represented in following table.

All of these algorithms are stochastic, so their result may be changed in different executions, but we can achieve a general outcome from a comparison of results.

Our research is different to previous works. For example, [33] focuses on the number of messages that are transferred between nodes, but our objectives is optimization of different system aspects simultaneously. Therefore comparison between presented work and other ones

Table 4: Final results

	Selected Path	Security	Privacy	User Productivity	Cost
AMOSA	$1 \to 2 \to 6 \to 10$	1.9	1.2	1.3	1.6
MOGA	$1 \to 3 \to 5 \to 10$	2.1	1.3	1.8	1.5
MOBC	$1 \to 7 \to 3 \to 9 \to 10$	1	2.1	2.6	2.1

is meaningless.

7 Conclusion

In this paper, we propose a scheme to model the network system in a real distributed environment according to security and privacy. Actually, security and privacy are critical concepts for all network systems, but in modelling we must consider their major parameters. In this paper, we recognize "security" and "privacy" based on their originality. Security is promoted by the head of an organization, but privacy is attractive for users. Indeed, users want free access to the assets of systems. Free access of users provides user productivity. Another important concept is that applying a security and privacy algorithm in real systems is an economic issue. We consider it as a "cost".

We optimize all major parameters of a network system (security, privacy, user productivity and cost) with three multi-objective optimization algorithms. AMOSA, MOGA and MOBC are used in this paper. Their results prove that the performances of AMOSA and MOGA are better that MOBC. The AMOSA algorithm can achieve a final result sooner that other algorithms, but the performance of MOGA is more stable.

Acknowledgments

The work in this paper has been supported by the National Natural Science Foundation of China (61272500), National High-tech R & D Program (863 Program) (2015AA017204) and the Beijing Natural Science Foundation (4142008).

References

[1] P. Agrawal, H. Kaur, D. Bhardwaj, "Analysis and synthesis of enhanced bee colony optimization with the traditional bee colony optimization to solve the traveling sales person problem," *International Journal of Computer & Technology*, vol. 2, no. 2, pp. 93–96, 2012.

[2] A. Badreddine, T. B. Romdhane, N. B. Amor, "A new process-based approach for implementing an integrated management system: Quality, security, environment," in *Proceedings of International Conference on Industrial Engineering (IMECS'09)*, pp. 1–6, Hong Kong, Mar. 2009.

[3] A. Badreddine, T. B. Romdhane, N. B. Amor, "A multi-objective risk management approach to implement an integrated management system: Quality, security, environment," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4728–4733, San Antonio, TX, USA, 2009.

[4] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269–283, 2008.

[5] A. Behnia, R. A. Rashid, J. A. Chaudhry, "A survey of information security risk analysis methods," *Smart Computing Review*, vol. 2, no. 1, pp. 79–94, 2012.

[6] W. Bux, "Local-area subnetworks: a performance comparison," *IEEE Transaction on Communications*, vol. 29, no. 10, pp. 1465–1473, 1981.

[7] R. Carvalho, M. Ferreira, "Using information technology to support knowledge conversion process," in *Proceedings of the 13th WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering*, pp. 176–1817, 2001.

[8] C. W. Choo, *The Knowing Organization*, Oxford University Press, 1998.

[9] W. Chung, W. Chen, J. F. Nunamaker, "Business Intelligence Explorer: A Knowledge Map Framework for Discovering Business Intelligence on the Web," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003.

[10] C. A. Coello, D. A. Van Veldhuizen, G. B. Lamont, *Evolutionary Algorithms for Solving Multi-objective Problems*, Springer, 2007.

[11] O. Etzioni, "The World Wide Web: Quagmire or Gold Mine?" *Communication of the ACM*, vol. 39, pp. 65–68, 1996.

[12] A. Federgruen, Z. Katalan, "The stochastic economic lot scheduling problem: cyclical base-stock policies with idle times," *Management Science*, vol. 44, pp. 989–1001, 1996.

[13] A. Federgruen, Z. Katalan, "Costumer waiting-time distributions under base-stock policies in single facility multi-item production systems," *Naval Research Logistics*, vol. 43, pp. 533–548, 1996.

[14] L. Flud, K. Sawka, J. Carmicheal, J. Kim, K. Hynes, *Intelligence Software Report 2002*, Cambridge, Flud & Company Inc., 2002.

[15] C. M. Fonseca, P. J. Fleming, "Genetic algorithm for multiobjective optimization: Formulation, discussion and generalization," in *Proceeding of 5th International Conference on Genetic Algorithms*, pp. 416–423, 1993.

[16] T. Gang, C. Kai, S. Bei, "The research & application of business intelligence system in retail industry," in *IEEE International Conference on Automation and Logistics*, pp. 87–91, 2008.

[17] D. Gupta, M. M. Srinivasan, "Polling systems with state-independent setup times," *Queueing Systems*, vol. 22, pp. 403–423, 1996.

[18] M. Gupta, G. Sharma, "An efficient modified artificial bee colony algorithm for job scheduling problem," *International Journal of Soft Computing and Engineering*, vol. 1, no. 6, pp. 303–315, 2012.

[19] A. Haidine, R. Lehnert, "Multi-case multi-objective simulated annealing (MC-MOSA): New approach to adopt simulated annealing to multi-objective optimization," *International Journal of Information Technology*, vol. 4, no. 3, pp. 197, 2008.

[20] S. M. Hashemi, J. He, "An approach for risk management of computer security base on polling system," in *The 16th IEEE International Conference on Communication Technology (ICCT'15)*, pp. 912–918, 2015.

[21] S. M. Hashemi, J. He, "BI-based approach for computer security," *The 3rd IEEE International Conference on New Media*, Indonesia, 2015.

[22] A. K. Jain, R. C. Dubes, *Algorithms for Clustering Data*, Englewood Cliffs, NJ, USA, Prentice-Hall, 1988.

[23] E. Kiesling, C. Strausss, C. Stummer, "A multi-objective decision support framework for simulation-based security control selection," in *Seventh IEEE International Conference on Availability, Reliability and Security*, pp. 454–462, 2012.

[24] L. Klienrock, H. Levy, "The analysis of random polling systems," *Operation Research*, vol. 36, no. 5, pp. 716–732, 1988.

[25] A. Konak, D. W. Konak, A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, vol. 91, pp. 992–1007, 2006.

[26] D. Kumar, D. Kashyap, K. K. Mishra, A. K. Misra, "Security vs. cost: An issue of multi-objective optimization for choosing PGP algorithms," in *IEEE International Conference on Computer & Communication Technology (ICCCT'10)*, pp. 532–535, 2010.

[27] H. Levy, M. Sidi, "Polling systems: Applications, modelling and optimization," *IEEE Transaction on Communication*, vol. 38, no. 10, pp. 1750–1760, 1990.

[28] T. Li, D. Logothetis, M. Veeraraghavan, "Analysis of a polling system for telephony traffic with application to wireless LANs," *IEEE Transaction on Wireless Communications*, vol. 5, no. 6, pp. 1284–1293, 2006.

[29] X. Lin, "Map displays for information retrieval," *Journal of the American Society for Information Science*, vol. 48, pp. 40–54, 1997.

[30] D. Micheal, Y. H. Yacov, "Influence diagrams with multiple objectives and tradeoff analysis," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 34, no. 3, pp. 293–304, 2004.

[31] R. Murugan, M. R. Mohan, "Artificial bee colony optimization for the combined heat and power economic dispatch problem," *ARPN Journal of Engineering and Applied Sciences*, vol. 5, no. 7, pp. 9–18, 2012.

[32] T. Neubauer, C. Stummer, E. Weippl, "Workshop-based multiobjective security safeguard selection," in *Proceedings of the First IEEE International Conference on Availability, Reliability and Security (ARES'06)*, pp. 366–373, 2006.

[33] T. Okimoto, N. Ikegai, T. Ribeiro, K. Inoue, H. Okada, H. Maruyama, "Cyber security problem based on multi-objective distributed constraint optimization technique," in *43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W'13)*, pp. 1–7, 2013.

[34] J. Raissi, "Performance impact of imitation in multi-objective security service provisioning," in *Proceedings of IEEE*, pp. 1–6, 2013.

[35] B. Schneier, *Secrets & Lies: Digital Security in a Networked World*, New York, NY: Wiley, 2000.

[36] R. Spence, *Information Visualization*, pp. 60–67, ACM press, 2001.

[37] W. Stallings, *Cryptography and Network Security Principles and Practices*, Pearson Education, 2004.

[38] S. Suriya, R. Deepalakshmi, S. Kannan, S. Shantharajah, "Enhanced bee colony algorithm for complex optimization problems," *International Journal on Computer Science and Engineering*, vol. 4, no. 1, pp. 72, 2012.

[39] V. Viduto, W. Huang, C. Maple, "Toward optimal multi-objective models of network security: Survey," in *Proceedings of the 17th International Conference on Automation & Computing*, pp. 6–11, 2011.

[40] V. Viduto, C. Maple, W. Huang, A. Bochenkov, "A multi-objective genetic algorithm for minimizing network security risk and cost," in *IEEE International Conference on High Performance Computing and Simulation (HPCS'12)*, pp. 462–467, 2012.

[41] L. Wang, *A Course in Fuzzy System and Control*, Prentice-Hall International, Inc., pp. 4–7, 1997.

[42] J. A. Weststrate, *Analysis and Optimization of Polling Systems*, Ph.D. Thesis, Tilburg University, 1992.

[43] A. Wierman, E. M. Winands, O. J. Boxama, "Scheduling in polling systems," *Performance Evaluation*, vol. 64, no. 9, pp. 1009–1028, 2007.

[44] L. Wu, G. Barash, C. Bartolini, "A service-oriented architecture for business intelligence," *IEEE International Conference on Service-oriented Computing and Applications (SOCA'07)*, pp. 279–285, 2007.

Biography

Seyed Mahmood Hashemi received his bachelor from Islamic Azad University (Qazvin Branch) in software engineering at 2001 his master from Islamic Azad University (Science and Research Branch) in artificial intelligence at 2003. He is currently PhD candidate in Beijing University of Technology (BJUT). His research interests are Internet of Things (IoT), network security and Artificial Intelligence (AI).

Jingsha He received his Master's and doctoral degrees in computer engineering from the University of Maryland at College Park in the US. He is currently a professor in the School of Software Engineering at Beijing University of Technology (BJUT) in Beijing, China. Prior to joining BJUT in 2003, Prof. He worked for several multi-national companies such as IBM Corp., MCI Communications Corp. and Fujitsu Labs in the US. where he published more than 10 papers and received 12 U.S. patents. Since joining BJUT in 2003, Prof. He has published nearly 240 papers in journals and international conferences, received nearly 40 patents and 30 software copyrights in China and co-authored 7 books. He has been the principal investigators of more than 20 research projects. Prof. He's research interests include information security, wireless networks and digital forensics.

Alireza Ebrahimi Basabi received his Master's degrees in Software engineering from the University of Beijing University of post and telecommunications (BUPT) in the china. He is currently a first year PhD student working under the supervision of Professor JingSha He in the Beijing University of Technology (BJUT). He has a background in system administration and software developer. His research interests include social media, IOT (Internet of Things), Ai (Artificial Intelligence), cloud computing, information assurance and Network security.

A Study of Relationship Among Goldbach Conjecture, Twin Prime and Fibonacci Number

Chenglian Liu
Department of Computer Science, Huizhou University
Huizhou 516007, P.R. China
(Email: chenglian.liu@gmail.com)
(Received Jan. 21, 2016; revised and accepted Apr. 17 & May 31, 2016)

Abstract

In 2015, Liu et al. proposed a relationship between RSA public key cryptosystem and Goldbach's conjecture properties. In this paper I will examine two other relationship's with Goldbach's conjecture: 1) Goldbach's conjecture and twin prime; 2) Goldbach's conjecture and Fibonacci number. I completely list all combination of twin prime in Goldbach conjecture, and propose a very simple method to recognize the prime in Fibonacci sequence. I also give a estimation formula to Goldbach's partition.

Keywords: Fibonacci number, Goldbach conjecture, twin prime

1 Introduction

The Goldbach conjecture and the twin prime issue are unsolved problems in Number Theory. It is well known that Chan [2] had a major discovery on Goldbach's conjecture by his "$1 + 2$" formal proof in 1973. Zhang [19] had a good result on the twin prime in 2014. Other articles [4–8, 10, 11, 18] also give good contributions. Liu, Chang, Wu and Ye [9] studied the relationship between RSA public key cryptosystem and Goldbach's conjecture properties. They found the RSA and Goldbach conjecture relationship, and also linked Goldbach's conjecture and twin prime. Liu et al.'s [9] listed two situations where there are twin prime numbers in Goldbach partition combinations such as Propositions 1 and 2. In addition to examining the relationship between Goldbach's conjecture and the twin prime and Fibonacci number, I will also make three major contributions:

1) Propose an estimating method which is better than Bruckman's method.

2) List all combinations of the twin prime in Goldbach's conjecture.

3) Propose a simple method to examine Fibonacci prime.

2 The Relationship Between Goldbach's Conjecture and the Twin Prime

In this section, I describe the relationship between Goldbach's conjecture and twin prime. This article is based on the work of Liu et al.'s [10] research. In Liu et al.'s article, they proposed 4 theorems, 6 propositions and 1 lemma. I continue that work and examine 6 additional situations of twin prime in Goldbach's partition. This issue is discussed in Section 2.3.

2.1 Literatures Reviews

To Goldbach's partition number, Bruckman's [1] estimated value was too large on the "number of error" range. Ye and Liu's [17] estimation is too vague, unclear and inaccurate. Based on this discussion, I give an estimating in which the number is closer to the true value. Constant [3] and Liu et al.'s [9] showed the relationship between the RSA cryptosystem and Goldbach's conjecture. Ye and Liu [17], and other articles [4, 10, 12] introduced Goldbach's conjecture and twin prime relationship. In Section 3 I will examine the relationship between Goldbach's conjecture and the Fibonacci number. The relationship between Goldbach's conjecture, twin prime, RSA and the Fibonacci number is a major topic and is shown in Figure 1. Notations are described in Table 1.

A variety of situations that may arise the twin primes in Goldbach conjecture, the all possible combination shown in Table 2.

2.2 The Goldbach Partition

Given a positive integer such as 480, there are 29 pairs to match Goldbach's rule, and 7 twin prime pairs. We say 29 is Goldbach's partition number. If randomly given an even number, it is easy to find Goldbach's partition, and we should say Goldbach's conjecture has been solved. However it is an unsolved problem today. Generally, to

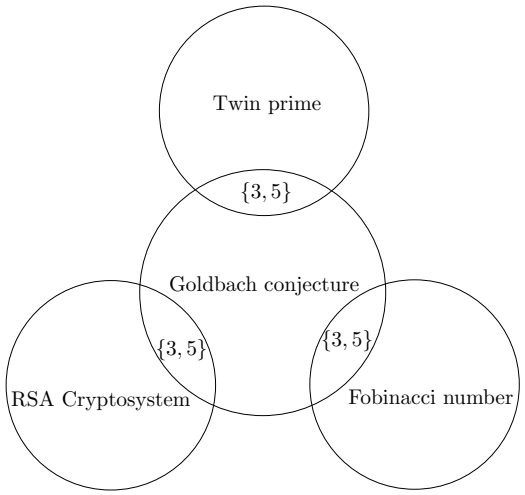


Figure 1: A relationship among Goldbach conjecture, twin prime, RSA and the Fibonacci number

Table 1: Notations

Symbols	Definition
GC	An even number for the Goldbach Conjecture (GC) number.
$GC(x)$	The number of Goldbach partition.
$GC'(x)$	The estimation number of $GC(x)$.
$GC \equiv 2\rfloor_4$	GC is congruent to two modulo four, we usually write $GC \equiv 2 \pmod 4$. But for convenience, we use $GC \equiv 2\rfloor_4$ instead here.

express GC number in the form of

$$GC = P_i + P_j \longmapsto (P_i - 2n) + (P_j + 2n), \tag{1}$$

where P_i and P_j are both primes. Let $R(n)$ be the number of representations of the Goldbach partition where $\prod_2$ is the twin prime constant [16], given $R(n) \sim 2\prod_2 \left(\prod_{P_k|n,k=2}\right) \frac{P_k-1}{P_k-2} \int_2^n \frac{dx}{(\ln x)^2}$. Ye and Liu [17] also gave the estimation formula $GC(x) = 2C\prod_{p\geq 3} \frac{(p-1)}{(p-2)} \cdot \frac{(Li(x))^2}{x} + \mathcal{O}(x \cdot e^{-c\sqrt{\ln x}})$.

In 2008, Bruckman [1] proposed a proof of the strong Goldbach conjecture, where the Goldbach function

$$\theta(2N) \equiv \sum_{k=3}^{2n-3} \delta(k)(2N-k) \tag{2}$$

is at least equal to one. Finally, the results

$$1 \leq \theta(2k+6) \leq k+1, \quad k = 0, 1, 2, \cdots. \tag{3}$$

When k approaches infinity, the error range becomes larger. For example:

$$\begin{aligned} \theta(32) &\leq 14, \ k = 13. \\ \theta(80) &\leq 38, \ k = 37. \\ \theta(138) &\leq 67, \ k = 66. \\ \theta(101200) &\leq 50598, \ k = 50597. \end{aligned}$$

Table 2: Twin prime appears probable in the Goldbach conjecture

item	even number				type
1	GC	$\equiv 0\rfloor_4$	$\equiv 0\rfloor_6$	$\equiv 4\rfloor_8$	$4n+2$
	$\frac{GC}{2}$	$\equiv 2\rfloor_4$	$\equiv 0\rfloor_6$	$\equiv 2\rfloor_8$	
		$\equiv 2\rfloor_4$	$\equiv 0\rfloor_6$	$\equiv 6\rfloor_8$	
2	GC	$\equiv 0\rfloor_4$	$\equiv 0\rfloor_6$	$\equiv 0\rfloor_8$	$4n$
	$\frac{GC}{2}$	$\equiv 0\rfloor_4$	$\equiv 0\rfloor_6$	$\equiv 0\rfloor_8$	
		$\equiv 0\rfloor_4$	$\equiv 0\rfloor_6$	$\equiv 4\rfloor_8$	
3	GC	$\equiv 0\rfloor_4$	$\equiv 4\rfloor_6$	$\equiv 4\rfloor_8$	$4n+2$
	$\frac{GC}{2}$	$\equiv 2\rfloor_4$	$\equiv 2\rfloor_6$	$\equiv 2\rfloor_8$	
		$\equiv 2\rfloor_4$	$\equiv 2\rfloor_6$	$\equiv 6\rfloor_8$	
4	GC	$\equiv 0\rfloor_4$	$\equiv 4\rfloor_6$	$\equiv 0\rfloor_8$	$4n$
	$\frac{GC}{2}$	$\equiv 0\rfloor_4$	$\equiv 2\rfloor_6$	$\equiv 0\rfloor_8$	
		$\equiv 0\rfloor_4$	$\equiv 2\rfloor_6$	$\equiv 4\rfloor_8$	
item	odd number				type
5	GC	$\equiv 2\rfloor_4$	$\equiv 0\rfloor_6$	$\equiv 2\rfloor_8$	$4n+1$
	$\frac{GC}{2}$	$\equiv 1\rfloor_4$	$\equiv 3\rfloor_6$	$\equiv 1\rfloor_8$	
		$\equiv 1\rfloor_4$	$\equiv 3\rfloor_6$	$\equiv 5\rfloor_8$	
6	GC	$\equiv 2\rfloor_4$	$\equiv 0\rfloor_6$	$\equiv 6\rfloor_8$	$4n+3$
	$\frac{GC}{2}$	$\equiv 3\rfloor_4$	$\equiv 3\rfloor_6$	$\equiv 3\rfloor_8$	
		$\equiv 3\rfloor_4$	$\equiv 3\rfloor_6$	$\equiv 7\rfloor_8$	
7	GC	$\equiv 2\rfloor_4$	$\equiv 4\rfloor_6$	$\equiv 2\rfloor_8$	$4n+1$
	$\frac{GC}{2}$	$\equiv 1\rfloor_4$	$\equiv 5\rfloor_6$	$\equiv 1\rfloor_8$	
		$\equiv 1\rfloor_4$	$\equiv 5\rfloor_6$	$\equiv 5\rfloor_8$	
8	GC	$\equiv 2\rfloor_4$	$\equiv 4\rfloor_6$	$\equiv 6\rfloor_8$	$4n+3$
	$\frac{GC}{2}$	$\equiv 3\rfloor_4$	$\equiv 5\rfloor_6$	$\equiv 3\rfloor_8$	
		$\equiv 3\rfloor_4$	$\equiv 5\rfloor_6$	$\equiv 7\rfloor_8$	

I obtained results from a large number of experimental data. I draw the curve from the data, and then calculates the formula from the two curves (see Figures 2 and 3). I found an interesting situation which GC is congruent to zero modulo six, or congruent to non-zero modulo six. An even number GC is randomly chosen, where $GC < 6$, if $GC \equiv 0 \pmod 6$, GC' is found where $GC'(x) \simeq \frac{1.8 \cdot GC}{11.931 \cdot GC^{0.2182}}$. Otherwise, I find other $GC'(x) \simeq \frac{1.9 \cdot GC}{6.2328 \cdot GC^{0.2144}}$. The expression shown in Equation (4).

$$GC \mapsto \begin{cases} \equiv 0 \bmod 6, \ GC'(x) \simeq \frac{1.8 \cdot GC}{11.931 \cdot GC^{0.2182}}. \\ \not\equiv 0 \bmod 6, \ GC'(x) \simeq \frac{1.9 \cdot GC}{6.2328 \cdot GC^{0.2144}}. \end{cases} \tag{4}$$

I compare my estimation with Bruckman's method based on the true value of Goldbach's partition. The results indicated that my method is better than Bruckman's method, see Table 3.

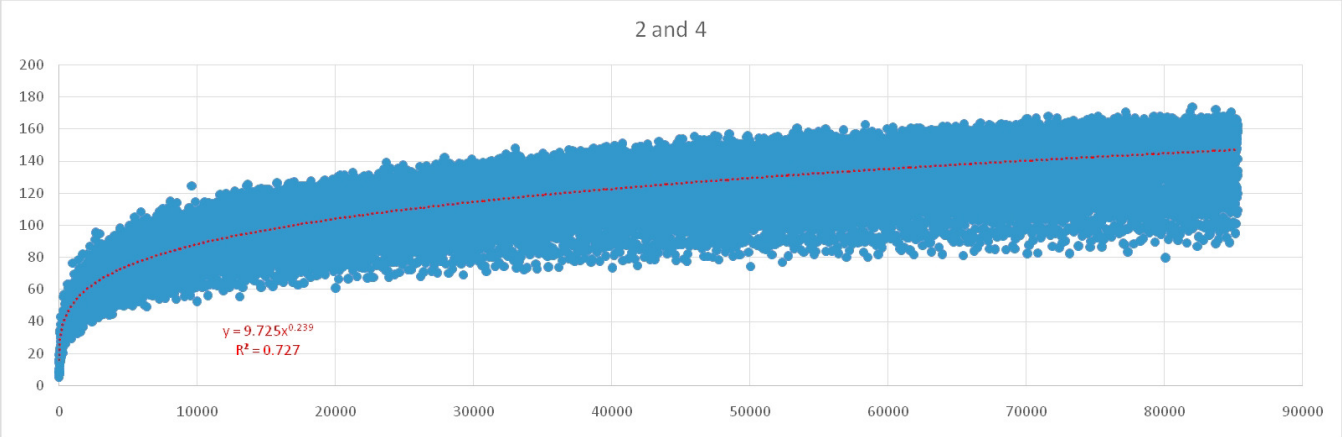


Figure 2: The curve of estimating, where $GC(x) \not\equiv 0 \pmod 6$

Table 3: Results of our method vs Bruckman's method

Item	Positive Integer	$GC(x)$	Our method	Bruckman's method	
			$GC'(x)$	k	$k+1$
1	12650	186	270	6322	6323
2	25300	314	464	12647	12648
3	50600	553	798	25297	25298
4	75900	1478	1970	37947	37948
5	101200	918	1372	50597	50598
6	126500	1140	1633	63247	63248
7	151800	2635	3396	75897	75898
8	177100	1802	2125	88547	88548
9	202400	1669	2359	101197	101198
10	227700	3688	4670	113847	113848
11	253000	2011	2808	126497	126498
12	278300	2130	3026	139147	139148
13	303600	4676	5854	151797	151798
14	318950	2059	3366	159472	159423
15	331600	2160	3470	165797	165798
16	344250	4652	6461	172122	172123
17	356900	2356	3675	178447	178448
18	369500	2321	3776	184747	184748
19	382200	6325	7015	191097	191098
20	394850	⋮	⋮	⋮	⋮
21	407500	⋮	⋮	⋮	⋮
22	420150	5264	7556	210072	210073

2.3 The Twin Prime

To help the description, I prefer to use corollary alternative propositions. These Corollaries 1 and 2 are original from Liu et al.'s [9] Propositions 1 and 2, I expand to examine 6 corollaries based on their work.

Corollary 1. *If* $P_i + P_j \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 4 \pmod 8$, *and* $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 2 \pmod 8$ *or* $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 6 \pmod 8$, *there may exist a twin prime where the* $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ *is of the form* $(4n+1)+(4n+3)$.

Proof. As known from assumption, $\frac{P_i+P_j}{2}$ is an even number, we have

$$\begin{cases} \frac{P_i+P_j}{2} - 1 \text{ is an odd number.} \\ \frac{P_i+P_j}{2} + 1 \text{ is an odd number too.} \end{cases}$$

Note that $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 6 \pmod 8$, we see the $\frac{P_i+P_j}{2}$ is of the form $4n+2$. Naturally, the $\frac{P_i+P_j}{2} - 1$ is $4n+1$ form, and $\frac{P_i+P_j}{2} + 1$ is $4n+3$ form. Otherwise, it is a contradiction.

Since $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 2 \pmod 8$, we know $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ is of the form $(4n+1)+(4n+3)$. □

Corollary 2. *If* $P_i + P_j \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 0 \pmod 8$, *and* $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 0 \pmod 8$ *or* $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 4 \pmod 8$, *there may exist a twin prime where* $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ *is of the form* $(4n+3)+(4n+1)$.

Proof. As known, the $\frac{P_i+P_j}{2}$ is an even number. Since $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 0 \pmod 8$. We see the $\frac{P_i+P_j}{2}$ is $4n$ form. Hence $\frac{P_i+P_j}{2} - 1$ is $4n+3$ form. Therefore $\frac{P_i+P_j}{2} + 1$ is $4n+1$ form.

Now, as $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 0 \pmod 6 \equiv 0 \pmod 8$, the $\frac{P_i+P_j}{2}$ is of the form $4n$ too.

Thus, the $\frac{P_i+P_j}{2}+1$ is of the form $4n+1$. This inference is consistent with the above statement. □

Corollary 3. *If* $P_i + P_j \equiv 0 \pmod 4 \equiv 4 \pmod 6 \equiv 4 \pmod 8$, *and* $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 2 \pmod 6 \equiv 2 \pmod 8$ *or* $\frac{P_i+P_j}{2} \equiv 2 \pmod 4 \equiv 2 \pmod 6 \equiv 6 \pmod 8$, *there may exist a twin prime where* $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ *is of the form* $(4n+3)+(4n+1)$.

Proof. As known from assumption, the $\frac{P_i+P_j}{2} \equiv 2 \pmod 4$ is an even number. Unsurprisingly, the $\frac{P_i+P_j}{2}-1$ is $4n+1$ form. Hence, the $\frac{P_i+P_j}{2}+1$ would be $4n+3$ form. Otherwise, it is a contradiction. This inference is consistent with the above statement. □

Corollary 4. *If* $P_i + P_j \equiv 0 \pmod 4 \equiv 4 \pmod 6 \equiv 0 \pmod 8$, *and* $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 2 \pmod 6 \equiv 0 \pmod 8$ *or* $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 2 \pmod 6 \equiv 4 \pmod 8$, *there may exist a twin prime where* $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ *is of the form* $(4n+3)+(4n+1)$.

Proof. As known, the $\frac{P_i+P_j}{2}$ is an even number. Since $\frac{P_i+P_j}{2} \equiv 0 \pmod 4 \equiv 2 \pmod 6 \equiv 0 \pmod 8$, the $\frac{P_i+P_j}{2}$ is an even number and a $4n$ form. Obviously, the $\frac{P_i+P_j}{2} - 1$ is $4n+3$ form, whereas the $\frac{P_i+P_j}{2} + 1$ is of the form $4n+1$. Otherwise, it is a contradiction. This inference is consistent with the above statement. □

Corollary 5. *If* $P_i + P_j \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 2 \pmod 8$, *and* $\frac{P_i+P_j}{2} \equiv 1 \pmod 4 \equiv 3 \pmod 6 \equiv 1 \pmod 8$ *or* $\frac{P_i+P_j}{2} \equiv 1 \pmod 4 \equiv 3 \pmod 6 \equiv 5 \pmod 8$, *there may exist a twin prime where* $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ *is of the form* $(4n+3)+(4n+1)$.

Proof. As known, the $\frac{P_i+P_j}{2} \equiv 1 \pmod 4$, the $\frac{P_i+P_j}{2}$ is $4n+1$ form. Since $4n+1$ and $4n+3$ are located on either side of the center point $4n+2$. Thus, the $(\frac{P_i+P_j}{2} + 2)$ is of the form $4n+3$. If not, it is a contradiction. □

Corollary 6. *If* $P_i + P_j \equiv 2 \pmod 4 \equiv 0 \pmod 6 \equiv 6 \pmod 8$, *and* $\frac{P_i+P_j}{2} \equiv 3 \pmod 4 \equiv 3 \pmod 6 \equiv 3 \pmod 8$ *or* $\frac{P_i+P_j}{2} \equiv 3 \pmod 4 \equiv 3 \pmod 6 \equiv 7 \pmod 8$, *there may exist a twin prime where* $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ *is of the form* $(4n+3)+(4n+1)$.

Proof. This proof is same with Corollary 5. I omit the proof here. □

Corollary 7. *If* $P_i + P_j \equiv 2 \pmod 4 \equiv 4 \pmod 6 \equiv 2 \pmod 8$, *and* $\frac{P_i+P_j}{2} \equiv 1 \pmod 4 \equiv 5 \pmod 6 \equiv 1 \pmod 8$ *or* $\frac{P_i+P_j}{2} \equiv 1 \pmod 4 \equiv 5 \pmod 6 \equiv 5 \pmod 8$, *there may exist a twin prime where* $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ *is of the form* $(4n+3)+(4n+1)$.

Proof. This proof is same with Corollary 5. I also omit the proof here. □

Corollary 8. *If* $P_i + P_j \equiv 2 \pmod 4 \equiv 4 \pmod 6 \equiv 6 \pmod 8$, *and* $\frac{P_i+P_j}{2} \equiv 3 \pmod 4 \equiv 5 \pmod 6 \equiv 3 \pmod 8$ *or* $\frac{P_i+P_j}{2} \equiv 3 \pmod 4 \equiv 5 \pmod 6 \equiv 7 \pmod 8$, *there may exist a twin prime where* $(\frac{P_i+P_j}{2} - 1, \frac{P_i+P_j}{2} + 1)$ *is of the form* $(4n+3)+(4n+1)$.

Proof. This proof is same with Corollary 5. I omit the proof here too. □

Exception:
There are 4 exceptions of even number between $[2, 1000]$ to the rule in Table 2.

$$402 \mapsto \begin{cases} 402 \equiv 2 \bmod 4 \equiv 0 \bmod 6 \equiv 2 \bmod 8, \\ 201 \equiv 1 \bmod 4 \equiv 3 \bmod 6 \equiv 1 \bmod 8. \end{cases} \quad (5)$$

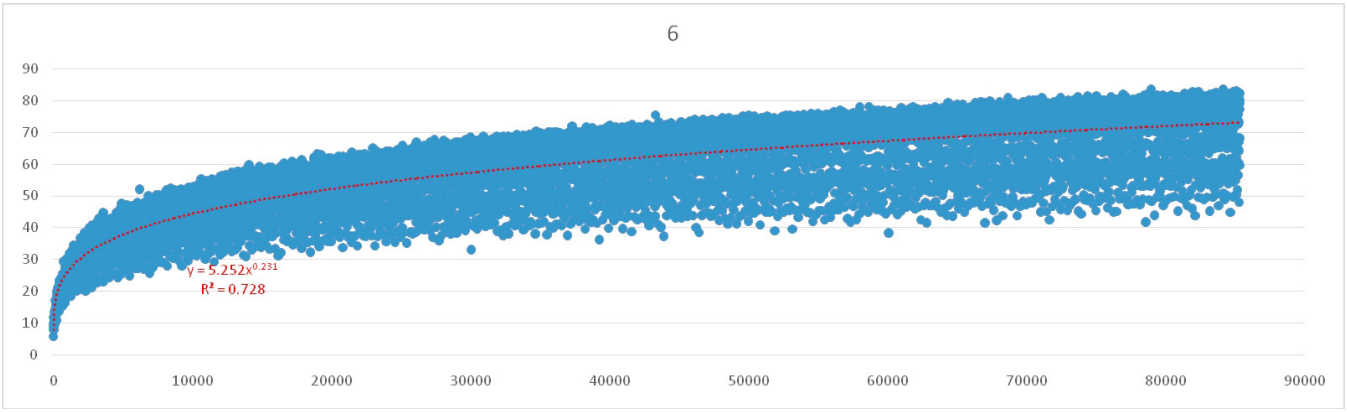


Figure 3: The curve of estimating, where $GC(x) \equiv 0 \pmod 6$

According from Table 2, the 402 matches item 5, however, there is no one twin prime in 17 prime pairs of Goldbach partition.

$$516 \mapsto \begin{cases} 516 \equiv 0 \bmod 4 \equiv 0 \bmod 6 \equiv 4 \bmod 8, \\ 258 \equiv 2 \bmod 4 \equiv 0 \bmod 6 \equiv 2 \bmod 8. \end{cases} \tag{6}$$

There are 23 prime pairs in Goldbach partition, but no one matches in the rule of item 1.

$$786 \mapsto \begin{cases} 786 \equiv 2 \bmod 4 \equiv 0 \bmod 6 \equiv 2 \bmod 8, \\ 393 \equiv 1 \bmod 4 \equiv 3 \bmod 6 \equiv 1 \bmod 8. \end{cases} \tag{7}$$

There are 30 prime pairs in Goldbach partition, but no one matches in the rule of item 5.

$$906 \mapsto \begin{cases} 906 \equiv 2 \bmod 4 \equiv 0 \bmod 6 \equiv 2 \bmod 8, \\ 453 \equiv 1 \bmod 4 \equiv 3 \bmod 6 \equiv 5 \bmod 8. \end{cases} \tag{8}$$

There are 34 prime pairs in Goldbach partition, but no one matches in the rule of item 5.

3 The Relationship of the Goldbach's Conjecture and the Fibonacci Number

This section will introduce about Fibonacci number [14, 15] and it's relationship with Goldbach's conjecture. Each positive number is the sum of the previous two integers, namely

$$F_n = F_{n-1} + F_{n-2}. \tag{9}$$

By Equation (9), we know the Fibonacci sequence as $\{0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ 55,\ 89, \cdots,\ \infty\}$. Wall [13] had good result in his article "Fibonacci Series Modulo M", a table was created in the appendix listing values for the function $k(n)$. This function is defined as the period of the Fibonacci numbers mod n before any repeats occur. For instance, $k(7) = 16$ since

$$F_n \bmod 7 = \{0, 1, 1, 2, 3, 5, 1, 6, 0, 6, 6, 5, 4, 2, 6, 1\}, \tag{10}$$

where F_n is the n-th Fibonacci number. Hence, the values in the sequence above are cyclic after 16 terms. On the other hand, the author notes another interesting property. The Fibonacci sequence has 'even-odd-odd' or 'odd-odd-even' rotation rules. The result shown in Table 4.

For n-th Fibonacci number, where $n \geq 1$, the F_n becomes an odd number if and only if $n \equiv 1 \pmod 3$ or $n \equiv 2 \pmod 3$, say

$$n \begin{cases} \equiv 0 \pmod 3, \text{ this is an even number.} \\ \equiv 1 \pmod 3, \text{ this is an odd number.} \\ \equiv 2 \pmod 3, \text{ this is an odd number.} \end{cases}$$

There is one example of the Fibonacci number matching the Goldbach's rule where the

$$F_6 = F_5 + F_4 \mapsto 3 + 5 = 8. \tag{11}$$

The Equation (11) is only one special case of Goldbach's conjecture in Fibonacci sequence nowaday. Since $F_{n \equiv 0 \pmod 3}$ has never been an prime that is an even number, we can say the $F_{n \equiv 1 \pmod 3}$ or $F_{n \equiv 2 \pmod 3}$ probable is a prime. There is an article by Wall [13] about Fibonacci prime in [14], but is a little different then what is discussed in this article.

Open Problems:

Can we find the second example of Goldbach's conjecture in Fibonacci sequence? In Fibonacci prime, I find an interesting phenomenon in my research.

1. If $n \equiv 3 \pmod 4$ and $F_n \equiv 1 \pmod 4$ where $n > 5$, the F_n probably be a prime, where

$$\begin{cases} F_{n \equiv 3 (\bmod 4)} \\ F_n \equiv 1 \pmod 4 \end{cases} \tag{12}$$

2. If $n \equiv 1 \pmod 4$ and $F_n \equiv 1 \pmod 4$ where $n > 5$, the F_n probably be also a prime, namely

$$\begin{cases} F_{n \equiv 1 (\bmod 4)} \\ F_n \equiv 1 \pmod 4 \end{cases} \tag{13}$$

We get following relationship as:

Goldbach's conjecture $\supseteq (odd + odd = even) \subset$ Fibonacci sequence.

Table 4: The special case of Fibonacci number matches the Goldbach's conjecture

				prime	prime	prime		prime				prime		prime	
n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
F_n	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377
		odd	odd	even	odd	odd	even	odd	odd	even	odd	odd	even	odd	odd
$F_n \underset{\text{mod}7}{\equiv} X$	0	1	1	2	3	5	1	6	0	6	6	5	4	2	6

			prime						prime						prime
n	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
F_n	610	987	1597	2584	4181	6765	10946	17711	28657	46368	75025	121393	196418	317811	514229
	even	odd	odd	even	odd	odd	even	odd	odd	even	odd	odd	even	odd	odd
$F_n \underset{\text{mod}7}{\equiv} X$	1	0	1	1	2	3	5	1	6	0	6	6	5	4	2

n	30	31	32	33	34	35	36	37	38	39
F_n	832040	1346269	2178309	3524578	5702887	9227465	14930352	24157817	39088169	63245986
	even	odd	odd	even	odd	odd	even	odd	odd	even
$F_n \underset{\text{mod}7}{\equiv} X$	6	1	0	1	1	2	3	5	1	6

				prime				prime		
n	40	41	42	43	44	45	...	81839		
F_n	102334155	165580141	267914296	433494437	701408733	1134903170	...	17103 digits		
	odd	odd	even	odd	odd	even				
$F_n \underset{\text{mod}7}{\equiv} X$	0	6	6	5	4	2		1		

4 Conclusions

I use Goldbach's conjecture as the center of interest. I then discusses the relationship among Goldbach's conjecture, twin prime, RSA cryptosystem and Fibonacci number and then makes three observations about the relationship:

1) The characteristics of twin prime in Goldbach's conjecture are analyzed, and then notes all situations of combination.

2) An estimate of Goldbach's partition is proposed where the result is more accurate than Bruckman's estimation.

3) Finally, I explore the relationship between Goldbach's conjecture and Fibonacci number. I mention a new discussion about searching the Fibonacci prime in its sequence.

As we can see, the authors is still working on these unsolved problems.

Acknowledgement

The authors would like to thank the anonymous reviewers for their useful comments. This work is partially supported from Huizhou University project under the number HZUXL201513, and HZUXL201514. This work also partially supported by student innovation training program under the grant number CX2016024, CX2016088 and CX2016089.

References

[1] P. S. Bruckman, "A proof of the strong Goldbach conjecture," *International Journal of Mathematical Education in Science and Technology*, vol. 39, pp. 1002–1009, Oct. 2008.

[2] J. R. Chen, "On the representation of a larger even integer as the sum of a prime and the product of at more two primes," *Scientia Sinica*, vol. 16, pp. 157–176, 1973.

[3] J. Constant, *Algebraic Factoring of the Cryptography Modulus and Proof of Goldbach's Conjecture*, July 2014. (`http://www.coolissues.com/mathematics/Goldbach/goldbach.htm`)

[4] J. Ghanouchi, *A Proof of Goldbach and De Polignac Conjectures*, July 16, 2016. (`http://unsolvedproblems.org/S20.pdf`)

[5] D. A. Goldston, J. Pintz, and C. Y. Yildirim, "Primes in tuples I," *Annals of Mathematics*, vol. 170, pp. 819–862, Sept. 2009.

[6] B. Green and T. Tao, "The primes contain arbitrarily long arithmetic progressions," *Annals of Mathematics*, vol. 167, pp. 481–547, 2008.
[7] B. Green and T. Tao, "Linear equations in primes," *Annals of Mathematics*, vol. 171, pp. 1753–1850, May 2010.
[8] G. Ikorong, "A reformulation of the Goldbach conjecture," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 11, no. 4, pp. 465–469, 2008.
[9] C. Liu, C. C. Chang, Z. P. Wu, and S. L. Ye, "A study of relationship between RSA public key cryptosystem and Goldbach's conjecture properties," *International Journal of Network Security*, vol. 17, pp. 445–453, July 2015.
[10] I. A. G. Nemron, "An original abstract over the twin primes, the Goldbach conjecture, the friendly numbers, the perfect numbers, the mersenne composite numbers, and the Sophie Germain primes," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 11, no. 6, pp. 715–726, 2008.
[11] K. Slinker, *A Proof of Goldbach's Conjecture that All Even Numbers Greater Than Four Are the Sum of Two Primes*, Jan. 2008. (`http://arxiv.org/vc/arxiv/papers/0712/0712.2381v10.pdf`)
[12] R. Turco, M. Colonnese, M. Nardelli, G. Di Maria, F. Di Noto, and A. Tulumello, *Goldbach, Twin Primes and Polignac Equivalent* RH*, the Landau's Prime Numbers and the Legendre's Conjecture*, July 16, 2016. (`http://eprints.bice.rm.cnr.it/647/1/`)
[13] D. D. Wall, "Fibonacci series modulo M," *The American Mathematical Monthly*, vol. 67, pp. 525–532, 1960.
[14] Wikipedia, *Fibonacci Number*, Feb. 2015. (`http://en.wikipedia.org/wiki/Fibonacci_number`)
[15] Wikipedia, *Fibonacci Prime*, Feb. 2015. (`http://en.wikipedia.org/wiki/Fibonacci_prime`)
[16] *Wolfram Research* Inc, *Goldbach Conjecture*, July 16, 2016. (`http://mathworld.wolfram.com/GoldbachConjecture.html`)
[17] J. Ye and C. Liu, *A Study of Goldbach's Conjecture and Polignac's Conjecture Equivalence Issues*, Cryptology ePrint Archive, Report 2013/843, 2013. (`http://eprint.iacr.org/2013/843.pdf`)
[18] S. Zhang, "Goldbach conjecture and the least prime number in an arithmetic progression," *Comptes Rendus-Mathematique*, vol. 348, pp. 241–242, Mar. 2010.
[19] Y. Zhang, "Bounded gaps between primes," *Annals of Mathematics*, vol. 179, no. 3, pp. 1121–1174, 2014.

Biography

Chenglian Liu received his B.S degree in Information Management from National Union University in 1992 and the MSc degree in National Defense from National Defense University in 2004. He studied his doctorate course at Royal Holloway, University of London from 2006 to 2009 under the supervised by Chris Mitchell. He is with a distinguished associate professor at Huizhou University since 2014. His research interests are in Key Agreement and Password Authentication, Number Theory and Cryptanalysis so on.

Cryptographically Imposed Model for Efficient Multiple Keyword-based Search over Encrypted Data in Cloud by Secure Index Using Bloom Filter and False Random Bit Generator

Devi Thiyagarajan, R. Ganesan
(Corresponding author: Devi Thiyagarajan)

School of Computing Science and Engineering, VIT University
Chennai Campus, Vandalur - Kelambakkam Road Chennai - 600 127, India
(Email: devi.t2013@vit.ac.in)
(Received Feb. 11, 2016; revised and accepted May 7 & June 5, 2016)

Abstract

Resources such as storage and network as a service to organizations and customers in reduced cost by cloud computing. The documents stored in cloud are of huge size and due to its sensitivity arises security and storage problems. Data is stored on remote location in cloud and cryptographic techniques resolve problems of data security. To ensure data to be secure in cloud, the client encrypts data and then stores on cloud by employing Hyper Elliptic Curve cryptography (HECC). Search over encrypted data makes the process of file retrieval from cloud more difficult. The paper proposes architecture of multiple keyword search by building index using Bloom filter and also pair key generation by false random bit generator. Bloom filter takes constant time for searching O (N) on large encrypted file systems without the need of document decryption thereby speeding up the process of ciphertext retrieval on user side. Bit array representing keyword information is only stored by data owner on cloud where the server is unable to find file content or query information. The experimental results show that Bloom filter based indexing is faster than traditional indexing schemes, the multiple keyword based search algorithm is effective in case of the response time of query and scalability of the system in case of size of data.

Keywords: Bloom filter, false random bit generation, HECC, secure index

1 Introduction

Cloud computing is one of the emerging field which replaces the burden of IT industry from spending huge expenditure on resources such as storage and network. Remote storage and easy accessibility of data combined with characteristics such as on-demand self-service, broad network access, resource pooling, rapid elasticity and measured services. Cloud is deployed as public, private, hybrid and community cloud with service delivery models such as SaaS (Software-as-a-Service), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). Resources in cloud are offered to both industries and individuals.

Though, cloud has many advantages certain issues such as security, privacy and interoperability do exist. The driving force for cloud computing is virtualization which enable multiple virtual machines to run with the help of single physical machine. As the infrastructure is being shared by several VMs, security issues do arise. Various surveys conducted portray security as one of the major challenge in cloud environment. Latest report on cloud computing challenges compares the issues and portrays that security stands as the first challenge.

Data confidentiality and privacy issues do arise due to multi-tenancy characteristic of cloud. To protect clients from such issues, cloud service provider need to follow certain mechanisms to keep data safe. But, malicious insider may act on behalf of the provider and send indelicate data. Such a situation makes a clear point that the security models cannot be build based on the trust of provider. Client needs to protect their data from malicious attacks both externally and also from internal adversaries. Clients outsource their files containing sensitive information to cloud for effective retrieval at the necessary time. Google search allows search over plaintext data. Such data is being stored in plaintext form in the cloud service which is vulnerable to attacks by adversaries. Protection of data from such malicious activities is prevented by storing encrypted data on cloud. Encryption methods are classified as symmetric (AES) [11, 21] and asymmetric (HECC) encryption algorithms [17, 23]. HECC is proven

to be the robust method of encrypting and decrypting files due to the hardness of hyperelliptic discrete logarithm problem. Incorporating such an algorithm in cloud enhances the data security in cloud environment.

Data owners share their files with authenticated users through retrieval mechanisms. Traditional methods retrieve the entire collection of files for a single search request from data user. This incurs more time for search reply and wastage of bandwidth. Selective retrieval of file based on data user request makes use of keywords. Search of plaintext data is not suited for cloud environment and encryption of data limits the search capability. Searchable encryption allows building an index with keywords and corresponding documents. Trapdoors along indexing enable data users to search over encrypted data in a secure manner and maintains privacy of document information as well as keywords. Such techniques remain unsuitable for cloud scenarios due to employment of symmetric encryption methods along with single keyword search. With these pitfalls it is necessary for an effective mechanism to download encrypted documents from cloud. The proposed model (Figure 1) incorporates asymmetric encryption such as HECC and position-based multiple keyword search scheme to maintain data security and privacy in cloud.

Retrieval of files from encrypted content need to be given attention as delivering the correct content to registered user is the ultimate goal. Search over encrypted data has to be done in an efficient way to reduce the overhead experienced by the users while decrypting their content from cloud. Hyperelliptic curve cryptography is employed for encryption of documents and search over encrypted files is carried by Bloom filter (BF) and False Random Bit Generator (FRBG).

Our contributions for providing data security and privacy includes:

1) Key pair generation by False Random Bit Generator (FRBG);
2) Building safe index using Bloom filter;
3) Search on encrypted files using Bloom Filter. Huge numbers of experiments were conducted to analyse the efficacy of the proposed architecture.

Paper organization is as follows: Literature review is presented in Section 2. Section 3 describes the architecture of proposed scheme. Section 4 discusses security analysis, Section 5 deals with implementation results and Section 6 concludes the paper.

2 Related Work

Searchable encryption [2, 3, 4, 5, 6, 7, 10, 27, 29] plays a major role in cryptography. Two-layer encryption of every keyword in a file is the first work on searchable encryption [27]. Usage of Bloom filter for construction of index of files along with trapdoor is stored on server. The request for search is followed by a trapdoor construction and the server on other hand tests with Bloom filters and sends the identifiers of files as response [10]. For effective search, one encrypted index was built for entire document collection where every entry in index contains keyword trapdoor along with file identifiers in encrypted state [6, 7]. Public-key based searchable encryption [4] played a major role in retrieval systems for privacy. Public key users store data on servers whereas authenticated users with the private key perform the operation of search. For efficient querying purpose conjunctive search [1], fuzzy keyword search [20, 31] and similarity keyword search [26] were proposed. A widely accepted retrieval technique namely private information retrieval [22] also helps in retrieving the items incurring complex computations.

Usage of k-nearest neighbour algorithm for searching documents along with ciphertext policy attribute-based encryption provides security and privacy for data [19, 30]. Hourglass function was utilised for the purpose of verifying the encrypted files [14] and identity based encryption is also employed for attaining data privacy [16]. Elgamal encryption [12] supporting fuzzy keyword search over encrypted data prevents inconsistencies occurring during search [28]. Ranked keyword search returned top-k files by using multiple keywords along with homomorphic encryption [32]. Attacks against ciphertexts are being analysed keyword search in cloud environment [15]. Hierarchical predicate encryption [24] along with access control also achieved keyword based search over encrypted data. The problems in bog specific search engines have been identified to optimize the search mechanism in cloud [25].

Authenticated users [13] retrieve files from cloud storage as authentication plays a major role in preventing illegitimate user access due to the application of discrete logarithms [18]. Hyperelliptic curve cryptography is combined with Advanced Encryption Standard and MD-5 algorithm in digital envelope for securing e-commerce channel [8]. The usage of HECC in cloud enhances the security of sensitive data by preventing from exposure.

A notable point is that all the existing schemes lack in certain functionality and remains unsuitable for cloud environments. Our work focusing on retrieving the exact document from cloud server based on Bloom filter and false random bits add robustness to the data security framework.

3 Construction of Secure Keyword Search Scheme

3.1 Background

1) **False Random Bit Function.**
$F : \{0,1\}^{Fi} \times \{0,1\}^{Fj-n} \rightarrow \{0,1\}^n$ be a false random function where sequence of F_i bits key is taken along with random F_j-n bit string and mapped to random n-bit string which is publicly known to all users.
2) **Bloom Filter.**

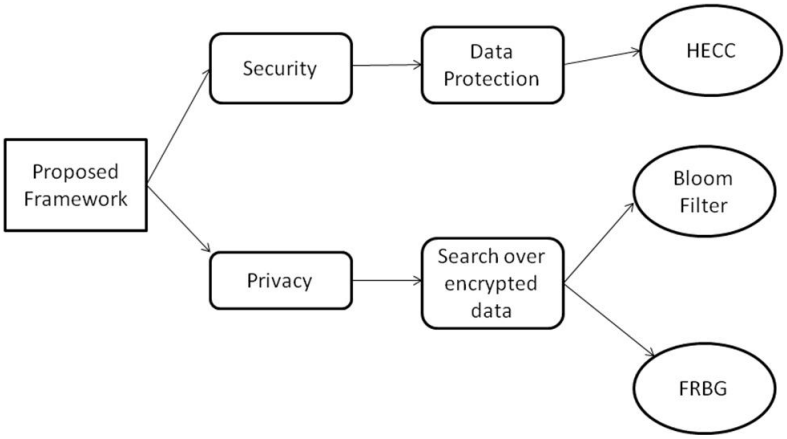


Figure 1: Proposed framework

A data structure named Bloom filter is used to identify the membership of a file in the file collection F and is used for the purpose of querying. A Bloom filter consists of set of 'm' elements $S_t = \{x_1.\ x_2, \cdots, x_m\}$ with 'n' hash functions $H_f = \{h_1, h_2, \cdots, h_n\}$. The r-bit Bloom filter is initially set to 0. Every element $x_k \in S$, $k = 1, 2, \cdots, m$, $h_i(x_k)$ for $i = 1, 2, \cdots, n$ is attained where $0 \leq \mathrm{h}_i(\mathrm{x}_k) < \mathrm{r}$ thereby respective bits to hash index is set for element $BF[h_i(x_k)]$. With input 'a', 'n' hash indices $h_i(a)$ for $i = 1, 2, \cdots, n$ is attained where $0 \leq h_i(a) < r$. Usage of hash functions may provide a positive answer while querying for an element which may not be the member of set [9]. Such condition is called as false positive.

3) **Hyperelliptic Curve Discrete Logarithm Problem.**
Given hyperelliptic curve of genus g over finite field F_q, point $P \in JC(K)$ of order n, point $Q\varepsilon < P >$, obtain integer $l\varepsilon[0, n-1]$ so $Q = lP$ where l is integer and is discrete logarithm of Q to base P, represented as $l = \log QP$.

4) **Multiple Keyword Search Scheme.**
With set of encrypted files C, MKS scheme returns the file identifiers (file_id) of those requested files to authenticated users. Multiple keyword set along with indexing (ID_{BF}) by Bloom filter speeds up the retrieval process of the users.

5) **Trapdoors.**
By the application of one-way hash functions, trapdoors are generated. With secret key (s_k) and keyword KW_i, trapdoor of KW_i is calculated as $T_{KWi} = f(s_k, KW_i)$.

3.2 System Model

Three participants of model are client (upload), cloud server (storage) and data users (file retrieval). Set of files after encryption by HECC ($C = C_1, C_2, \cdots, C_n$) are stored in cloud server along with keyword set $KW = \{KW_1, KW_2, \cdots, KW_{mo}\}$. Users registered with clients are authenticated and provided with the access over encrypted files C. In order to retrieve selective files, data user provides the multiple keywords of interest. The mapping between the search request from multiple users and files is the responsibility of CS as every file gets indexed with an unique file identifier (ID_f) and set of keyword. Multiple keyword based search scheme proceeds files containing specified keywords as the result of search to the authenticated users. Figure 2 depicts construction of multiple keyword search scheme.

3.3 Construction of Multiple Keyword Search Scheme

The multiple keyword search scheme performs the following three steps:

1) publicly known false random bit generation (FRBG) algorithm for pair_ key generation

2) index building using Bloom filter IB_{BF} with keyword set and

3) index search algorithm IS_{BF}.

The steps involved in the proposed scheme are:

1) Client sends the set (0, 1) along with files and keywords to FRBG.

2) Authenticated users send set (1, 0) to FRBG.

3) If the two different random bit sets matches, it returns the pair_ key and sets the bit as 1 else rejected.

4) Role of FRBG is string matching along with bit (0, 1) sets.

5) The cloud server CS contains the file identifier (id), bit matching and keywords.

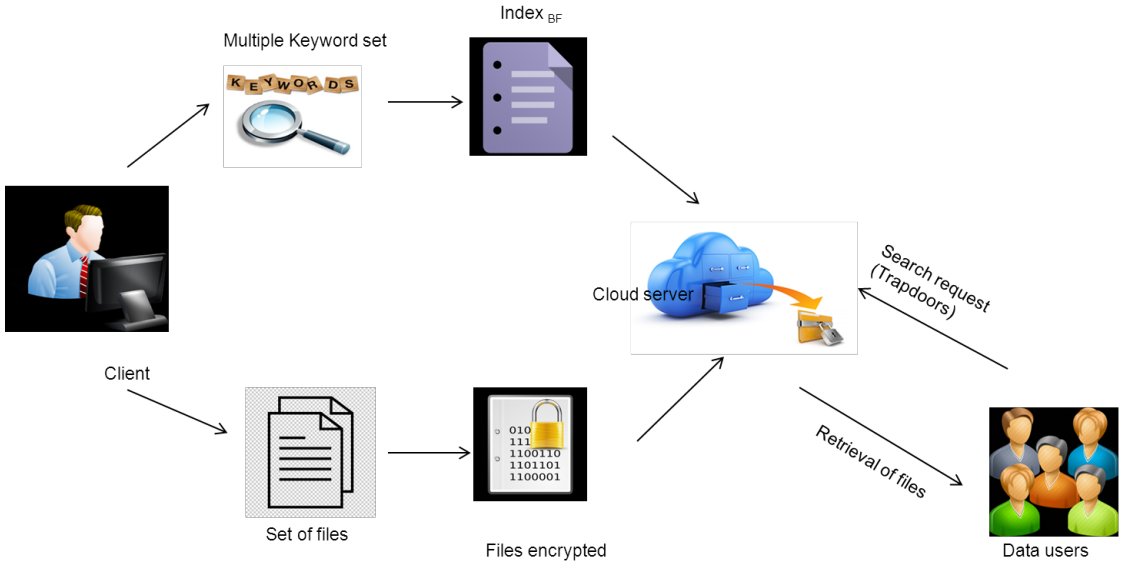


Figure 2: Construction of multiple keyword search scheme

6) Client runs the IndexBuild algorithm to build the secure index with Bloom filter and stores in server.

7) User on requesting the file, server runs the IndexSearch algorithm and in response returns the corresponding file in a secure manner.

Publicly Known False Random Bit Generation (FRBG).
The algorithm takes set of files F_i, message given by client which is used for mapping with strings of random bits (0, 1) and 8-bit key as input (See Algorithm 1). Output generated by the algorithm is sequence of false random bits $(y_1, y_2, \cdots, y_n)$. Suppose the input is in collection of files, it is followed by key pairing for client acknowledgement. The string along with the message is processed. Suppose if the paired key matches with the string, sequence of false random bits $(y_1, y_2, \cdots, y_n)$ are generated. By taking two different sets $\{0,1\}^s$ ant $\{1,0\}^{n-m}$ mapping is done. If mapping is correct, positions of bit are set to 1, else 0. Finally $y_{ij} = 1$ position of bit is set to 1 And false random numbers are generated (Figure 3).

Index Building with Bloom Filter.
Followed by pair_ key generation, secure index is constructed by means of Bloom filter. Formalization of the proposed scheme is as follows:

- Pub_{key} (P_k): Represents the public key of the given user (i.e) $P_k \in \{0,1\}^{Fi}$ which is kept publicly known.
- Secret Key (s_k): Represents the secret key for string matching which is to be kept known only to authenticated users.
- False Random Bit Function: $F : \{0,1\}^{Fi} * \{0,1\}^{Fj-n} \rightarrow \{0,1\}^n$ be false random function where sequence of F_i bits key is taken along with random F_j-n bit string and mapped to string of random n-bit which is widely identified by all users.
- Trapdoor: Let T_{ij} be a trapdoor whose inputs are public key (P_k), secret key (s_k) and keywords (KW) and generates the trapdoors for files F_i. T_{ij} $(P_k, KW) = E_{Pk}$ (KW) where E is encryption function. Encryption is performed by Hyperelliptic Curve Cryptography for security reasons.
- IndexBuild $_{BF}$ (F_i, P_k, KW, H): The algorithm takes files F_i, pubic key P_k, keywords KW and hash functions H as input. It generates string using false random bit generator (FRBG) and outputs the index for file F. Finally index is built for searching files (F_i) by using the sequence of random numbers generated by FRBG algorithm.

Algorithm 1 Publicly Known False Random Bit Generation (FRBG)

```
Input: Collection of files F_i, message m, pair_key;
Output: Set of false random numbers (y_1, y_2, ..., y_n);
for (x ∈ F_i U pair_key)
Generate String S_i message;
do (pair_key = = string)
Generate false random bits y_ij;
if (F : {0,1}^Fi * {1,0}^(Fj-n) → {0,1}^n) then
   while (y_ij == matching done) do
      return as 1;
   end while
else
   return 0;
end if
End
```

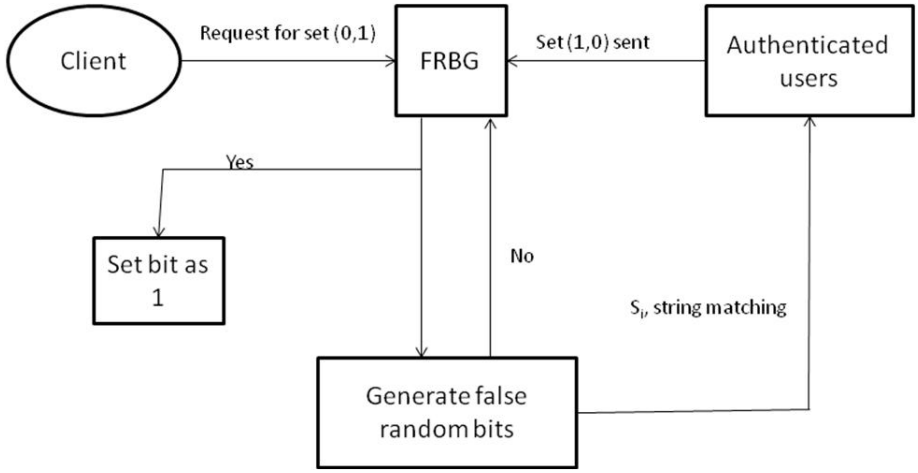


Figure 3: Pair_key generation by FRBG

- IndexSearchBF $_{BF}$ (IB_{BF}, T, key_pair, H): With the index built by using Bloom filter, trapdoor T, key_pair generated by FRBG and hash functions H the algorithm IndexSearch$_{BF}$ outputs the file containing the desired keyword KW.

  Finally index is built for searching files (F_i) by using the sequence of random numbers generated by FRBG algorithm (Figure 4).

Hash functions used in Bloom filter are publicly known thereby revealing the contents of files. The proposed scheme allows hash to be not applied directly on keywords KW but indexes each keyword KW by its encrypted version $E_{Pk}(KW_i)$. Hence, Bloom filter is built with hash values $H_f(E_{Pk}(KW_i))$, $f = 1, 2, \cdots, n$. The positions of bits in BF which are positioned to 1 equivalent to KW are similar for each and every file that contain KW. Due to vulnerability of frequency-based attacks, file identifier (ID) is used. $H_f(E_{key\_pair}(id(F_i), T(KW_i)))$, $f = 1, 2, \cdots, n$ is calculated as the hash function for Bloom filter. Suppose if the same keyword occurs in several different files, only the file with priority is retrieved. An attacker can find the specified file only if the trapdoor T is offered. The algorithm for index building using Bloom filter (IB_{BF}) is in Algorithm 2.

Each file in file collection F_i is assigned an unique identifier ID. The index constructed maps keywords to ID of file that actually contains the KW, thereby reducing search complexity. This index is then encrypted and stored on cloud. Search on inverted index with KW returns all the id of files containing the KW without looking into the original file collection F.

3.4 Index Search Algorithm IS_{BF}

Search with multiple keywords is done in the following manner. Suppose the user needs to search for keyword KW, trapdoor $(KW_i) = E_{Pk}(KW_i)$ is sent to cloud server CS. CS runs the $IndexSearch_{BF}$ algorithm on each file F in the set and returns the specified ones (See Algorithm 3.

Algorithm 2 Index Building using Bloom Filter (IB_{BF})

1: Input: Set of Files (F_i), public key P_k, keywords KW, $H(h_1, h_2, \cdots, h_n)$;
2: Output: IB_{BF} /* Index built for files using bloom filter */;
3: **if** ($BF_{Fi} \neq empty$) **then**
4: **for** all $KW_i \in F_i$ **do**
5: Generate trapdoor $T_{ij}(P_k, KW_i) = E_{Pk}(KW_i)$;
6: Generate string-matching $S_i = E_{key\_pair}(id(F_i), T(KW_i))$;
7: **for** $f = 1$ to n do **do**
8: Calculate pos_bit $(Pb_f) = H_f(S_i)$;
9: Set $IB_{BF}[Pb_f] = 1$;
10: **end for**
11: **end for**
12: Return IB_{BF};
13: **end if**
14: End

Algorithm 3 Index Search IS_{BF}

1: Input: Index built with Bloom filter IB_{BF}, Trapdoor $T(KW_i)$, key_pair, $H(h_1, h_2, \cdots, h_n)$;
2: Output: Specified file F or ϕ;
3: Compute $y = E_{key\_pair}(id(F), m(KW_i))$
4: **for** $f = 1$ to n do **do**
5: **if** $IB_{BF}[H_f(y)] \neq 1$ then **then**
6: Return ϕ;
7: **end if**
8: **end for**
9: Return F;
10: End

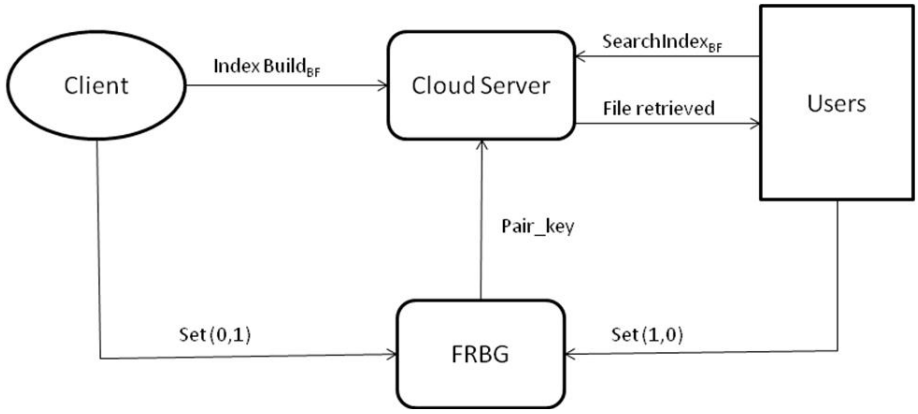


Figure 4: Secure search scheme with bloom filter and FRBG

4 Security Analysis

The security analysis of the proposed model is discussed.

Theorem 1. *Given the collection of files F_1, F_2, F_3, F_n, if $F_1 \in F_n$, then perform index building using Bloom Filter and also retrieve false random bits with FRBG. For the proof of this theorem, we have n number of files with index building $IndexBuild_{BF}(F_i, P_k, KW, H) = TrapdoorT_{ij}(P_k, KW_i) = E_{Pk}(KW_i) + F : \{0,1\}^{Fi} * \{0,1\}^{Fj-n} \rightarrow \{0,1\}^n$. Since the hash value position is greater than 0 (i.e) 1, if the theorem generates false random bits and the cryptographic algorithm over secured access using Bloom filter is preserved. To measure the time taken for index building IB_{BF} and index search IS_{BF} employing Bloom filter is $O(n)$ which is much less compared to brute-force approach. As shown in Figure 5, X-axis represents time taken for search in milliseconds and Y-axis represents file size in KB.*

Claim 1: Use of Bloom filter imposes reduced false positives.
The presence or absence of a file in file collection is straightforwardly verified by usage of Bloom filter. False positives may occur as a result of querying for a file which may not be in the collection. Since every related bit in BF is positioned to 1 for hash indices, there are less chances of false positives thereby reducing them totally.

$$
\begin{aligned}
&BF \in \pm 1|0\{\\
&\quad If\ pos\_1 :\ values\\
&\qquad Generate\ bits\ FRBG\\
&\quad Else\\
&\qquad Return\ pos\_0\}
\end{aligned}
$$

Claim 2: The proposed scheme prevents remote attacks.
An adversary can obtain the information stored remotely on servers through remote attacks. The proposed scheme employs FRBG for key matching process and bloom filter for indexing and searching process. $F : \{0,1\}^{Fi} * \{0,1\}^{Fj-n} \rightarrow \{0,1\}^n$ provides security through matching the perfect keyword and usage allowed only for authenticated users. These schemes are proven to be robust and provide two level security for data storage and retrieval on cloud. Thus an adversary is prevented by obtaining access on the server information and remote attacks are averted.

Claim 3: The proposed scheme is secure with the usage of Bloom filter and false random bit generation function.
Two major variants employed in search over encrypted data are Bloom filter and false random bit generation function. Pair_key generated by FRBG secures the model by authenticating users ($T_{ij}(P_k, KW_i) = E_{Pk}(KW_i)$) and the usage of Bloom filter enhances the search process to be performed in minimum time $O(n)$.

Claim 4: Search over encrypted file is faster than any other schemes with reduced search time.
Bloom filters speed up search over encrypted data in cloud as availability of necessary data in the right time is one of the most important feature in cloud computing. Assume there are 'n' number of files to be stored on remote server $F_1, F_2, \cdots, F_n$ containing 'm' keywords. Search time by applying brute force method is $O(n * m)$. By using 'n' number of bloom filters, the search time is reduced as $O(n)$. To check whether the respective bit in BF is set to 1, it takes only constant time thereby reducing the time for search.

5 Implementation Results

The proposed model along with multiple keyword search scheme is implemented in Openstack for effective results. When compared with traditional techniques, the proposed framework worked faster with less time complexity (Figure 5). X-axis represents the time taken to retrieve

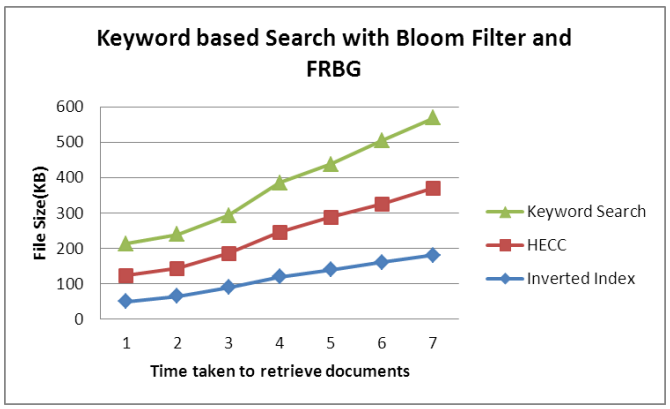


Figure 5: Comparison of proposed scheme with inverted index

documents from cloud server and Y-axis represents the file size uploaded on cloud server. With the increase in file size, the search time also reduces for our proposed scheme and vice-versa. Compared with HECC, inverted index incurs additional time for retrieving files. Using HECC algorithm, Bloom filter and false random bit generator the time taken for searching the document is less compared to existing works. When file size increases, Bloom filter has 0% false positive rate. When the data size increases, space occupied by inverted index much more compared to Bloom filter. Applications supporting data sets of larger size can prefer Bloom filter instead of inverted index allowing minimum amount of false positive rate.

Efficiency of retrieved files:

Let N_{kw} be the number of documents that contain keywords KW and N_Q be number of documents returned as a result of a query Q. The efficacy of retrieved files is given by the formula

$$E = N_{kw}/N_Q.$$

Such overhead of files retrieved presents the amount of unused files and the percentage is not high in case of the proposed model.

6 Conclusion

The proposed model achieves data security and privacy. The utilization of HECC for encryption/decryption purpose, false random bit generator for pair_key generation and Bloom filter for index building and searching files in cloud enables the proposed model to be efficient. Faster encryption and decryption time is achieved by HECC and lesser search time for retrieving encrypted files from cloud is attained by FRBG and Bloom filter add advantages to the data security model. Minimal key size and the hardness of discrete logarithmic problem of HECC prevent the adversaries from attacking the model. Two-layer protection with techniques such as FRBG and Bloom filter avoids unauthorized users from retrieval of data from cloud servers. The future work is focused on providing security to data-at-rest and also ranking the search results based on relevance.

References

[1] M. Azraoui, K. Elkhiyaoui, M. Onen and R. Molva, "Publicly verifiable conjunctive keyword search in outsourced databases," in *IEEE conference on Communications and Network Security (CNS'15)*, pp. 619–627, 2015.

[2] F. Bao, R. Deng, X. Ding, and Y. Yang, "Private query on encrypted data in multi-user settings," in *Information Security Practice and Experience*, LNCS 4991, pp. 71–85, Springer, 2008.

[3] M. Bellare, A. Boldyreva, and A. ONeill, "Deterministic and efficiently searchable encryption," in *Annual International Cryptology Conference*, pp. 535–552, Springer, 2007.

[4] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology (EUROCRYPT'04)*, LNCS 3027, pp. 506–522, Springer, 2004.

[5] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Theory of Cryptography Conference*, pp. 535–554, 2007.

[6] Y. C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *International Conference on Applied Cryptography and Network Security*, pp. 442–455, 2005.

[7] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.

[8] R. Ganesan, M. Gobi, and K. Vivekanandan, "A novel digital envelope approach for a secure e-commerce channel," *International Journal of Network Security*, vol. 11, no. 3, pp. 121–127, 2010.

[9] W. R. Ghanem, M. Shokir, and M. Dessoky, "Defense Against Selfish PUEA in Cognitive Radio Networks Based on Hash Message Authentication Code," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 12–21, 2016.

[10] E. J. Goh, *Secure Indexes*, Cryptology ePrint Archive, Report 2003/216, 2003. (`http://eprint.iacr.org/`)

[11] T. Gulom, "The encryption algorithm AES-PES16-1 and AES-RFWKPES16-1 based on network PES16-1 and RFWKPES16-1," *International Journal of Electronics and Information Engineering*, vol. 3, no. 2, pp. 53–66, 2015.

[12] M. S. Hwang, C. C. Chang, 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.
[13] M. S. Hwang and C. Yu Liu, "Authenticated encryption schemes: current status and key issues," *International Journal of Network Security*, vol. 1, no. 2, pp. 61–73, 2005.
[14] K. Hu and W. Zhang, "Efficient verification of data encryption on cloud servers," in *2014 Tewlfth Annual Conference on Privacy, Security and Trust (PST'14)*, pp. 314–321, 2014.
[15] F. G. Jeng, S. Y. Lin, B. J. Wang, C. H. Wang, T. H. Chen, "On the security of privacy-preserving keyword searching for cloud storage services," *International Journal of Network Security*, vol. 18, no. 3, 2016, pp. 597–600, 2016.
[16] T. Jung, X. Y. Li, Z. Wan and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 190–199, 2015.
[17] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.
[18] C. C. Lee, M. S. Hwang, Li H. Li, "A new key authentication scheme based on discrete logarithms," *Applied Mathematics and Computation*, vol. 139, no. 2, pp. 343–349, 2003.
[19] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," *IEEE Transactions On Emerging Topics in Computing*, vol. 3, no. 1, pp. 127–138, 2015.
[20] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling efficient fuzzy keyword search over encrypted data in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1467–1479, 2012.
[21] A. Mersaid, T. Gulom, "The encryption algorithm AES-RFWKIDEA32-1 based on network RFWKIDEA32-1," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 1–11, 2016.
[22] R. Ostrovsky, *Software Protection and Simulations on Oblivious Rams*, Ph.D. Dissertation, Massachusetts Institute of Technology, 1992.
[23] K. R. Santosh, C. Narasimham, and P. Shetty, "Cryptanalysis of multi-prime RSA with two decryption exponents," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 40–44, 2016.
[24] Z. Shen, J. Shu and W. Xue, "Keyword search with access control over encrypted data in cloud computing," in *International Symposium of Quality of Service (IWQoS'14)*, pp. 87–92, 2014.
[25] J. Singh, "Cloud based technique for blog search optimization," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 32–39, 2016.
[26] M. Strizhov and I. Ray, "Multi-keyword similarity search over encrypted cloud data," *IFIP International Information Security Conference*, pp. 52–65, Springer, 2014.
[27] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 44–55, 2000.
[28] Y. Wang, W. Bao, Y. Zhao, X. Hu, and Z. Qin, "An ElGamal encryption with fuzzy keyword search on cloud environment," *International Journal of Network Security*, vol. 18, no. 3, pp. 481–486, 2016.
[29] B. Waters, D. Balfanz, G. Durfee, and D. Smetters, "Building an encrypted and searchable audit log," in *Proceedings of 11th Annual Network and Distributed System*, 2004.
[30] H. Xu, S. Guo and K. Chen, "Building confidential and efficient query services in the cloud with rasp data perturbation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 1–18, 2014.
[31] Q. Xu, H. Shen, Y. Sang, H. Tian, "Privacy-preserving ranked fuzzy keyword search over encrypted cloud data," in *International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'13)*, pp. 239–245, 2013.
[32] J. Yu, P. Lu, Y. Zhu, G. Xue and M. Li, "Towards secure multi-keyword top-k retrieval over encrypted cloud data," *IEEE Transactions On Dependable And Secure Computing*, vol. 10, no. 4, pp. 239–250, 2013.

Biography

Devi Thiyagarajan is pursuing Ph.D in computer science from VIT University. Her areas of interest include cloud computing and cloud data security.

R. Ganesan is working as Associate Professor in VIT University. His area of expertise includes wireless sensor networks, cryptography and network security and cloud security. He has published several papers in international journals and conferences.

A Secure and Efficient Privacy-Preserving Attribute Matchmaking Protocol for Mobile Social Networks

K. Arthi[1], M. Chandramouli Reddy[2]
(Corresponding author: M. Chandramouli Reddy)
Department of Information Technology, Veltech Technical University[1]
Department of Computer Science & Engineering, Veltech Technical University[2]
42, Avadi-Vel Tech Road, Avadi, Chennai, Tamil Nadu 600062, India
(Email: mouli.veltech@gmail.com)
(Received Feb. 24, 2016; revised and accepted Apr. 25 & May 7, 2016)

Abstract

The advances in mobile and communication technologies lead to advancement of Mobile Social Networks (MSNs). MSN changed the way people communicate and exchange the private and sensitive information among the friend groups via mobile phones. Due to the involvement of private and sensitive information, MSN demands for efficient and privacy-preserving matchmaking protocols to prevent the unintended data (attribute) leakage. Many existing matchmaking protocols are based on user's private and specific data. Malicious participants may opt their attribute set arbitrarily so as to discover more information about the attributes of an honest participant. Hence, there is great chance of information leakage to a dishonest participant. In this context, Sarpong et al. had proposed a first of its kind of an authenticated hybrid matchmaking protocol that will help match-pair initiators to find an appropriate pair which satisfies the pre-defined threshold number of common attributes. Sarpong et al. had claimed that their protocol restricts attribute leakage to unintended participants and proved to be secure. Unfortunately, in Sarpong et al. scheme, after thorough analysis, we demonstrate that, their scheme suffers from data (attribute) leakage, in which the initiator and the participant can compute or achieve all the attributes of each other. Also we show that Sarpong et al. scheme requires huge computation and communication cost. As a part of our contribution we will propose an efficient and secure match making protocol which is light weight and restricts attribute leakage to the participants.

Keywords: Matchmaking protocols, mobile social networks, privacy-preserving attribute matchmaking protocol

1 Introduction

The advances in mobile and communication technologies lead to advancement of traditional online social network to Mobile Social Networks (MSNs). MSN facilitates real time personal and user specific data sharing and instant messaging among friend groups. Due to the exchange of private and shared information among the participants, finding a matching pair privately is a critical requirement in MSN.

A private matchmaking is a primary feature of private set intersection. Matchmaking protocol is a critical requirement for MSN in which two or more mutually mistrustful parties A and B consists of attribute sets SA and SB desire to compute together the intersection in such a way that both A and B should not take any information particular to the other opponent. A and B must learn only the common attributes among them i.e. SA∩SB nothing more.

In literature, many match making algorithms has been proposed based on various parameters. Few matchmaking protocols [1, 6, 7] has been proposed based on Certificate Authority C.A, in which C.A authenticates the entities attributes. Another matchmaking technique is fully distributed [9], which eliminates C.A. The participants perform the distribution of attributes among themselves, computing the intersection set. The initiator and the multi parties exchange their attributes using Shamir secret sharing scheme [3]. The Hybrid technique [12] is a commonly used technique in which the CA performs only the verification of attributes and managing the communication among the entities. The protocol participants will perform the attribute sharing and matchmaking opera-

tions. Recently Huang et al. [14] had proposed an Identity Based Encryption scheme for match making in social networks.

However, in this context in 2015, Chiou et al. [2] and Sarpong et al. [15] had proposed matchmaking protocols in which the initiator finds the best match among multiple participants who has the maximum similar attribute as the initiator. Sarpong et al. claimed that their scheme protects user's attributes from unnecessary leakage to unintended persons. In this manuscript after thorough analysis of Sarpong et al. scheme, we will demonstrate that in Sarpong et al. scheme, the participants can achieve the attributes of other participants and requires huge computation and communication cost.

As a part of our contribution, we will propose a secure and light weight matchmaking protocol for MSN, which resists the pitfalls in Sarpong and other related schemes.

The remaining of the paper is systematized as follows: In Section 2, we will give a brief review on system architecture. In Section 3, we will briefly discuss on Sarpong et al. [15] scheme. In Section 4, we discuss on the security pitfalls in Sarpong et al. scheme. In Section 5, the anomalies in Sarpong et al. scheme are discussed. Our proposed matchmaking protocol is presented in Section 6. In Section 7 we deliberate on informal security analysis of our proposed scheme. In Section 8, we deliberate on formal security analysis of our proposed scheme using widely accepted random oracle model. We discuss on Simulating experiments and performance evaluations are provided in Section 9.

2 System Architecture and Design Goals

Our system architecture consists of mainly three entities:a user (initiator) to find the best match among multiple participants (called participants) and a trusted certificate authority (CA), as depicted in Figure 1.

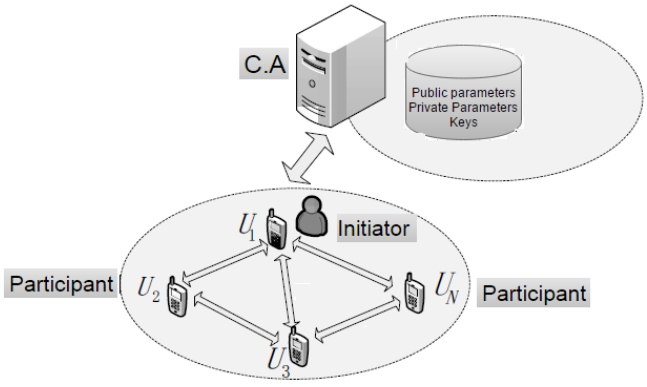


Figure 1: The system architecture

We will follow below mentioned privacy levels similar to [5].

Privacy Level 1: On completion of execution of matchmaking protocol,the initiator and each potential friend (participant)must identify only the intersection set and its size.

Privacy Level 2: On completion of execution of matchmaking protocol,the initiator and each potential friend (participant) must know only the ranking of the size of the intersection set mutually. Apart from these, no other information should be intercepted by the participants.

3 Brief Review of Sarpong et al. Matchmaking Algorithm

Assume Alice is the initiator of the protocol to find out the closest match among 'm' participant's (for brevity, we assume that Alice is communicating with a single participant Bob to find out the common attributes. The other participants also perform and exchange similar messages as Bob with Alice. Alice also exchanges same messages as it exchanges with Bob.) having portable devices and can connect with each other using PAN or Bluetooth or Wifi. $A_{Threshold}$ is the threshold value for the attribute matching set by the initiator Alice, i.e. to qualify as a match pair for initiator, there should be minimum of $A_{Threshold}$ number of common attributes between pairs. The initiator Alice consists of 'm' attributes, i.e. $a = \{a_1, a_2, \cdots, a_m\}$ and Bob consists 'p' attributes, i.e. $b = \{b_1, b_2, \cdots, b_p\}$. In the matchmaking, if two attributes are semantically same, then only they are treated as the same.

3.1 Key Generation

K1. Alice and Bob computes RSA key pairs (e_A, d_A), (e_B, d_B) respectively using p, q which are large prime numbers, where e_A, e_B are the public variables.

K2. CA computes RSA key pair is (e, d), where N=p*q.

K3. CA makes $< e, N >$ public.

3.2 Attribute Certification

A1. The attributes of Alice and Bob are $a = \{a_1, a_2, \cdots, a_m\}$ and $b = \{b_1, b_2, \cdots, b_k\}$.

A2. Alice exponentiates her attribute set using the public key of CA, i.e. 'e'. $a^e = \{a_1{}^e, a_2{}^e, \cdots, a_m{}^e\}$.

A3. Bob also exponentiates his attributes as $b^e = \{b_1{}^e, b_2{}^e, \cdots, b_k{}^e\}$.

A4. Alice to get the attributes certified by CA, forwards a message $E_e\{a^e||ID_A||UN_A||e, e_A\}$ to CA which contains the attribute set computed in A2, its identity, user name, its public key and CA public key. The message is encrypted with the CA public key, i.e. 'e'.

A5. Bob also to get the attributes certified by CA, forwards a message $E_e\{b^e||ID_B||UN_B||e||e_B\}$ to CA which contains the attribute set computed in A3, its identity, user name, its public key and CA public key. The message is encrypted with the CA public key, i.e. 'e'.

A6. The CA certifies the Alice attributes and returns $A = \{(a_1, s_1), (a_2, s_2), \cdots, (a_m, s_m)\}$ to Alice, where $s_i = H(ID_A||a_i)^d \bmod N$ using its private key 'd'.

A7. The CA also certifies the Bob attributes and returns $B = \{(b_1, \sigma_1), (b_2, \sigma_2), \cdots, (b_k, \sigma_k)\}$ to Bob, where $\sigma_1 = H(ID_B||b_1) \bmod N$.

3.3 Matchmaking Phase

M1. On getting the attributes certified by the CA, the private attributes of Alice and Bob becomes $A = \{(a_1, s_1), (a_2, s_2), \cdots, (a_m, s_m)\}$, $B = \{(b_1, \sigma_1), (b_2, \sigma_2), \cdots, (b_k, \sigma_k)\}$ respectively.

Challenge Phase:

M2. Alice picks 'm' arbitrary random numbers R_i for each attribute $i = \{1, 2, \cdots, m\}$ and computes $MA_i = S_i.g^{R_i} \bmod N$, i.e. $MA_1 = s_1.g^{R_1} \bmod N$, $MA_2 = s_2.g^{R_2} \bmod N$, $MA_3 = s_3.g^{R_3} \bmod N$ and sends $MES_1 = \{MA_1, MA_2, \cdots, MA_m\}$ to Bob.

M3. Bob also chooses an arbitrary numbers P_k for each attribute $k = \{1, 2, \cdots, k\}$ and computes $MB_k = \sigma_k.g^{P_k} \bmod N$, i.e. $MB_1 = \sigma_1.g^{P_1} = H(ID_B||b_1).g^{P_1} \bmod N$, $MB_2 = \sigma_2.g^{P_2} \bmod N = H(ID_B||b_2).g^{P_2} \bmod N$, and sends $MES_2 = \{MB_1, MB_2, \cdots, MB_k\}$ to Alice.

3.4 Encoding Phase

M4. Alice chooses an arbitrary number R_a and computes $Z_A = g^{e.R_a} \bmod N$, $MB_k^* = (MB_k)^{e*R_a} = \{{MB_1}^{e.R_a}, {MB_2}^{e.R_a}, \cdots, {MB_m}^{e.R_a}\} = \{(H(ID_B||b_1).g^{P_1}).^{e.R_a}, (H(ID_B||b_2).g^{P_2}).^{e.R_a}, \cdots, (H(ID_B||b_m).g^{P_m}).^{e.R_a}.\}$.

M5. Alice performs arbitrary permutation $RPA = \zeta\{a_1, a_2, \cdots, a_m\}^{R_a} = \zeta\{{a_1}^{R_a}, {a_2}^{R_a}, \cdots, {a_m}^{R_a}\}$ and sends $MES_3 = \{Z_A||MB_k^*||RPA\}$ to Bob.

M6. Bob also opts an arbitrary number R_b and computes $ZB = g^{e.R_b} \bmod N$, $(MES_1)^{e.R_b} = \{M1^{e.R_b}, M_2^{e.R_b}, M_3^{e.R_b}, \cdots, M_k^{e.R_b}\} = \{(s_1.g^{R_1})^{e.R_b}, (s_2.g^{R_2})^{e.R_b}, \cdots, (S_k.g^{R_m})^{e.R_b}.\}$.

M7. Bob chooses an arbitrary permutation $RPB = \zeta\{b_1^{R_b}, b_2^{R_b}, \cdots, b_k^{R_b}\}$ and sends $MES_4 = \{Z_k||(MES_1)^{e.R_b}||RPB\}$ to Alice.

3.5 Set Intersection Phase

M8. Alice sends her signed message $Sig_{d_A}(ID_A \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ to Bob.

M9. Bob also sends his signed message $Sig_{d_B}(ID_B \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ to Alice.

M10. Now Alice and Bob verify that received MES_1, MES_2, MES_3, MES_4 values are equivalent to the received or computed values in the previous steps.

M11. Alice share her random number to Bob by sending $Sig_{d_A}(ID_A||ID_B||R_a)$. Similarly Bob also shares his arbitrary number by sending $Sig_{d_B}(ID_B||ID_A||R_b)$.

3.6 Recovery Phase

M12. Alice computes a list $KA = \zeta A\{{a_1}^{R_aR_b}, {a_2}^{R_aR_b}, \cdots, {a_m}^{R_aR_b}\}$ and direct to Bob. Bob also computes $KB = \zeta B\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$ and send it to Alice.

M13. In order to know the actual common attributes, Alice sends her random permutation by encrypting with the Bob public key, i.e. e_B, E_{e_B} (ζA). Similarly, Bob sends his random permutations to Alice by encrypting with the Alice public key, i.e. e_a, $E_{e_A}(\zeta B)$.

M14. Alice already knowing ζB, can able to compute ζB^{-1} and retrieves $\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$, similarly Bob able to compute ζA^{-1} and recover $\{{a_1}^{R_aR_b}, {a_2}^{R_aR_b}, \cdots, {a_m}^{R_aR_b}\}$. Now both Alice and Bob know their actual common attributes.

4 Cryptanalysis of Sarpong et al. Algorithm

In this section we do a thoughtful security analysis of Sarpong et al. [15] scheme. Based on the actions perform by the attackers,to intercept the information exchanged among the protocol entities,the attackers in the system are classified into two types i.e.malicious and semi-honest.The malicious or active attackers deviate the protocol, and try to achieve the private information from the protocol participants by providing the forged attributes. The semi-honest or passive attackers are intrusive, follows the protocol rules as specified and try to achieve extra information from the messages exchanged in the protocol execution.

4.1 Failure to Resist Malicious Attack

In Sarpong et al. [15] scheme, in M13 of matching phase, Alice sends its random permutation, i.e. $E_{e_B}(\zeta A)$ by encrypting with the Bob public key. Similarly Bob sends its random permutation $E_{e_A}(\zeta B)$ by encrypting with the Alice public key. On receiving the encrypted message

$E_{e_A}(\zeta B)$, Alice perform following steps as depicted below:

Step 1: Decrypts $E_{e_A}(\zeta B)$ using its private key d_A, i.e. $D_{d_A}E_{e_A}(\zeta B) = \zeta B$.

Step 2: Alice performs inverse operation ζB^{-1} on KB, i.e. $\zeta B^{-1}(KB) = \zeta B^{-1}\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$ to retrieve original list, i.e. $\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$.

Step 3: In M11 of matching phase Bob sends the message $Sig_{d_B}(ID_B||ID_A||R_b)$ to Alice. Alice retrieves $\{ID_B, ID_A, R_b\}$ from the received message. Alice already knows her R_a, hence Alice can perform an inverse operation on each received value in $\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$, i.e. $\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}{R_b}^{-1}{R_a}^{-1} = \{b_1, b_2, b_3, b_4, \cdots, b_k\}$. Hence, Alice comes to know all the attributes of Bob, along with the common attributes. Similar is the case with Bob, in which Bob also comes to know all the attributes of Alice along with the common attributes by executing the above steps similar to Bob. Therefore we can conclude that, Sarpong et al. scheme fails to achieve the primary requirement of match making algorithm, in which the participant and initiator must know only the common attributes.

5 Pitfalls or Anomalies in Sarpong et al. Algorithm

5.1 Requires Huge Communication Cost

In M2 and M5 steps of match making process, Alice sends MES_1 and MES_3 to Bob respectively. In M8, Alice again forwards MES_1, MES_3 to Bob in a message $Sig_{d_A}(ID_A||MES_1||MES_2||MES_3||MES_4)$. Bob, on receiving the message $Sig_{d_A}(ID_A \;||\; MES_1 \;||\; MES_2 \;||\; MES_3 \;||\; MES_4)$, decrypts the message to get $\{ID_A, MES_1, MES_2, MES_3, MES_4\}$ and uses $MES_1, MES_2, MES_3, MES_4$ to validate, whether the transferred and received values are valid or not. To validate the messages transferred, a message digest operations like hash functions Eg: SHA-1 etc can be used, which outputs a fixed length data, hence reduces the necessitate to transfer full messages.

Similar is the case with the Bob. In M3 Bob sends MES_2, in M7 Bob sends MES_4 to Alice. In M9 Bob again sends these messages in the form of $Sig_{d_B}(ID_B||MES_1||MES_2||MES_3||MES_4)$ to Alice, which consumes huge communication cost.

5.2 Requires Huge Computation Cost

In M2 and M3 steps of match making process, Alice and Bob selects 'm' and 'p' arbitrary numbers respectively. Alice computes $MA_i = s_i.gR_i \bmod N$, where $1 <= i <= m$. Similarly Bob computes $MB_k = \sigma_k.gP_k = H(ID_B||b_k).gP_k \bmod N$, where $i <= k <= p$. Totally for one participant and one initiator, the Sarpong et al. schemes $k*m$ random numbers, which requires huge computation cost Alice.

6 Our Proposed Scheme

In this section we present our improved scheme over Sarpong et al. [15] scheme. The Key Generation and Attribute Certification phases of our proposed scheme are similar to Sarpong et al. scheme. We will start from matchmaking phase.Even though the protocol runs between Alice and 'm' participant, for brevity we consider only Bob as another participant of the protocol.

The main contributions of our work are:

1) An enhanced matchmaking protocol for MSN is proposed, which is based on the trusted certification authority (TCA) and provides a better privacy preserving by introducing the protocol's privacy levels.

2) The theoretical analysis is performed to prove the correctness and security of the protocol. Simulating experiments are conducted to evaluate the efficiency of the protocol.

3) We discuss the arbitration mechanisms for the protocol to detect malicious users who are cheating the others.

6.1 Matchmaking Phase

M1. On getting the attributes certified by the CA, the private attributes of Alice and Bob becomes $A = \{(a_1, s_1), (a_2, s_2), \cdots, (a_m, s_m)\}$, $B = \{(b_1, \sigma_1), (b_2, \sigma_2), \cdots, (b_k, \sigma_k)\}$, respectively.

Challenge Phase:

M2. Alice picks a single arbitrary random number R_1, and computes $MA_1 = S_i.g^{R_1} \bmod N$, i.e. $MA_1 = s_1.g^{R_1} \bmod N$, $MA_2 = s_2.g^{R_2} \bmod N$, $MA3 = s3.g^{R_3} \bmod N$ and sends $MES_1 = \{MA_1, MA_2, \cdots, MA_m\}$ to Bob.

M3. Each participant also chooses an arbitrary numbers P_1 computes $MB_k = \sigma_k.g^{P_1} \bmod N$, i.e. $MB_1 = \sigma_1.g^{P_1} = H(ID_B||b_1).g^{P_1} \bmod N$, $MB_2 = \sigma_2.g^{P_1} \bmod N = H(ID_B||b_2).g^{P_1} \bmod N$ and sends $MES_2 = \{MB_1, MB_2, \cdots, MB_k\}$ to Alice.

Encoding Phase:

M4. Alice chooses an arbitrary number R_a and computes $Z_A = g^{e.R_a} \bmod N$, $MB_k^* = (MB_k)^{e*R_a} = \{{MB_1}^{e.R_a}, {MB_2}^{e.R_a}, \cdots, {MBm}^{e.R_a}\} = \{(H(ID_B||b_1).g^{P_1})^{e.R_a}, (H(ID_B||b_2).g^{P_2})^{e.R_a}, \cdots, (H(ID_B||bm).g^{Pm})^{e.R_a}\}$.

M5. Alice performs arbitrary permutation $RPA = \zeta\{a_1, a_2, \cdots, ak\}^{R_a} = \zeta\{{a_1}^{R_a}, {a_2}^{R_a}, \cdots, {a_k}^{R_a}\}$ and sends $MES_3 = \{Z_A||MB_k^*||RPA\}$ to Bob.

M6. Bob also opts an arbitrary number R_b and computes $ZB = g^{e.R_b} \bmod N$, $(MES_1)^{e.R_b} = \{M1^{e.R_b}, {M_2}^{e.R_b}, {M_3}^{e.R_b}, ...{M_k}^{e.R_b}\} = \{(s_1.g^{R_1})^{e.R_b}, (s_2.g^{R_2})^{e.R_b}, \cdots, (s_k.g^{R_m})^{e.R_b}\}$.

M7. Bob chooses an arbitrary permutation $RPB = \zeta\{{b_1}^{R_b}, {b_2}^{R_b}, \cdots, {b_k}^{R_b}\}$ and sends $MES_4 = \{Z_k||(MES_1)^{e.R_b}||RPB\}$ to Alice.

6.2 Set Intersection Phase

M8. Alice computes $M_1 = ID_A \oplus h(MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$, $M_2 = h(ID_A \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ and forwards $\{M_1, M_2\}$ to Bob.

M9. Bob computes $M_3 = ID_B \oplus h(MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$, $M_4 = h(ID_B \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ and forwards $\{M_3, M_4\}$ to Alice.

M10. On receiving $\{M_3, M_4\}$ from Bob, Alice achieves ${ID_B}^* = M_3 \oplus h(MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$, computes $M_4^* = h({ID_B}^* \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ and compares the computed M_4^* with the received M_4. If both are equal Alice authenticates Bob. Similarly, Bob achieves ID_A* from M1, and computes $M_2^* = h({ID_A}^*||MES_1||MES_2||MES_3||MES_4)$. If computed M_2^* equals the received M_2, Bob authenticates Alice.

M11. Alice share her random number to Bob by sending an encrypted message using the Bob public key, so that the message can be decrypted only by Bob using his private key, i.e. d_B. Its $D_{d_B}(E_{e_B}(ID_A||ID_B||R_a||R_1)) = \{ID_A, ID_B, R_a, R_1\}$.

M12. Similarly Bob also shares his arbitrary numbers by sending an encrypted message using Alice public key, i.e. e_A, i.e. $E_{e_A}(ID_B||ID_A||R_b||P_1) = \{ID_B, ID_A, R_b, P_1\}$.

M13. Alice computes a random permuted list $KA = \zeta A\{h(a_1||R_1)^{R_aR_b}, h(a_2||R_1)^{R_aR_b}, \cdots, h(a_m||R_1)^{R_aR_b}\}$ and direct to Bob. Bob also computes $KB = \zeta B\{h(b_1||P_1)^{R_bR_a}, h(b_2||P_1)^{R_bR_a}, \cdots, h(b_k||P_1)^{R_bR_a}\}$ and send it to Alice.

M14. In order to know the actual common attributes, Alice sends her random permutation by encrypting with the Bob public key, i.e. $e_B, E_{e_B}(\zeta A)$. Similarly, Bob sends his random permutations to Alice by encrypting with the Alice public key, i.e. $e_A, E_{e_A}(\zeta B)$.

Recovery Phase:

M15. Alice already knowing ζB, can able to compute ζB^{-1} and retrieves $\{h(b_1||P_1)^{R_bR_a}, h(b_2||P_1)^{R_bR_a}, \cdots, h(b_k||P_1)^{R_bR_a}\}$, similarly Bob able to compute ζA^{-1} and recover $\{h(a_1||R_1)^{R_aR_b}, h(a_2||R_1)^{R_aR_b}, \cdots, h(a_m||R_1)^{R_aR_b}\}$.

M16. For each attribute $\{a_1, a_2, \cdots, a_m\}$, Alice computes $\{h(a_1||P_1)^{R_aR_b}, h(a_2||P_1)^{R_aR_b}, \cdots, h(a_m||P_1)^{R_aR_b}\}$ and compares with the attribute list $\{h(b_1||P_1)^{R_bR_a}, h(b_2||P_1)^{R_bR_a}, \cdots, h(b_k||P_1)^{R_bR_a}\}$. The comparison gives the Alice, the number of attributes in common and their actual values with Bob. Bob also perform same computations as Alice. As Alice and Bob uses hash function and session specific arbitrary numbers to compute $\{h(a_1||R_1)^{R_aR_b}, \cdots\}$, $\{h(b_1||P_1)^{R_aR_b}, \cdots\}$, if an attribute sent by Bob is not matching against any value in the Alice attribute list, it is computationally infeasible for Alice to achieve or compute the non-matching attribute, due to one way property of hash function, even the Alice knows P_1, R_a, R_b. Similar is the case with Bob.

M17. Hence in our scheme, there is no chance of leakage of attributes to opponent, in case of non-matching attributes.

7 Informal Security Strengths of The Proposed Scheme

7.1 Resists Malicious and Semi-Honest Participant Attack (Attribute Verification)

In our proposed scheme, in Attribute Certification phase, the initiator Alice and the participant Bob submit their attribute set $a = \{a_1, a_2, \cdots, a_m\}$ and $b = \{b_1, b_2, \cdots, b_k\}$ to CA. The CA certifies the attributes and returns $A = \{(a_1, s_1), (a_2, s_2), \cdots, (a_m, s_m)\}$ to Alice, where $s_i = H(ID_A||a_i)^d \bmod N$. Similarly for Bob, CA returns $B = \{(b_1, \sigma_1), (b_2, \sigma_2), \cdots, (b_k, \sigma_k)\}$ where $\sigma i = H(ID_B||b_i) \bmod N$. As CA binding the attributes with their hash value, the participants are restricted to change their attributes later. This step restricts the attacks by malicious and semi-honest participants.

7.2 Resists Malicious Participant Attack (Attribute Mapping) Scenario 1

In matchmaking phase of our scheme, i.e. M5, M7 the initiator Alice sends the randomly permuted attribute set, i.e. $RPA = \zeta\{a_1, a_2, \cdots, ak\}^{R_a} = \zeta\{{a_1}^{R_a}, {a_2}^{R_a}, \cdots, {a_k}^{R_a}\}$ to Bob. Similarly Bob also opts an arbitrary number R_b and computes an arbitrary permutation $RPB = \zeta\{{b_1}^{R_b}, {b_2}^{R_b}, \cdots, {b_k}^{R_b}\}$. Due to the random

permutations, even though the participant or malicious attacker achieves a_1R_a etc, it is impossible to map ${a_1}^{R_a}$ to an entry in the list $\zeta\{{a_1}^{R_a}, {a_2}^{R_a}, \cdots, {a_k}^{R_a}\}$. Also in M11, M12 the Alice and Bob exchange their random numbers by encrypting with the public key of the opponents. In M11 Alice share her random number to Bob by sending an encrypted message using the Bob public key, so that the message is decrypted only by Bob using his private key, i.e. d_B. its $D_{d_B}(E_{e_B}(ID_A||ID_B||R_a||R_1)) = \{ID_A, ID_B, R_a, R_1\}$. Similar is the case with the Bob. Hence, it is impossible for an attacker to achieve the attributes of the participants.

7.3 Resists Malicious Participant Attack (Dynamic Attributes) Scenario 2

In all the previous works including Sarpong et al, the initiator and the participants make their attribute set random by exponentiating the attributes with random number. If the random numbers are known to the malicious users, they can retrieve the attribute values which are static. Hence, it will leak the attribute information. In our proposed scheme, Alice computes a random permuted list $KA = \zeta A\{h(a_1||R_1)^{R_aR_b}, h(a_2||R_1)^{R_aR_b}, \cdots, h(a_m||R_1)^{R_aR_b}\}$ in which a hash of an attribute is concatenated with a random number and exponentiated. In this case, the same attribute value results in a different hash value each time it is sent. Hence, it is difficult for an attacker to achieve any information from the attribute set.

Due to space restrictions, we have discussed above attacks only. Our scheme resists all major cryptographic attacks and achieves attribute privacy.

8 Formal Security Strengths of The Proposed Scheme

We prove the security strengths of our proposed scheme by comparing what a malicious attacker can do in the real protocol execution against what the attacker can do in an ideal world. In the ideal-world execution, both participants would submit their attribute set to an imaginary trusted certificate authority i.e. TCA. The trusted TCA certifies the attributes submitted. Once the validations are done, the communicating parties compute the intersection set. If a protocol participant submits a message without proper validation from TCA, the other participants ignore or drop the message. Automatically, this confirms that the real-world attribute set intersection protocol is as secure as the protocol in the ideal world that depend on TCA.

We now formally outline the ideal functionality. The security definition involves the communication between TCA and malicious attackers.

Authorize: If TCA receives an authorization or verification request from participant P_i, TCA computes $\sigma_i = H(ID_i||b_i)$ where $1 <= i <= k$ for totally 'k' attributes and submits σ_i back to P_i.

Set Intersect: Initiator P_i sends a request to perform set intersection to party P_j. Similarly P_j sends a request to perform set intersection to party P_i. P_i and P_j now run an ideal set intersection protocol as below:

1) P_i sends a set S_i to P_j and P_j sends S_j to P_i. On receiving the entities set, both P_i and P_j checks whether each attribute in S_i and S_j has proper validations from TCA. Let ${S_i}^* \leq S_i$ and ${S_j}^* \leq S_j$ denote maximal subsets of S_i and S_j that have proper validations.
2) P_i and P_j compute the intersection set $I \leftarrow {S_i}^* \bigcap {S_j}^*$.

8.1 Formal Security Analysis

In this part, we demonstrate the security strengths of our scheme formally by using the random oracle model and we will illustrate that our scheme is strongly secure.

In the random oracle model, an ideal simulator 'S' is constructed and given a black box access to an attacker 'E'. The communication between an attacker 'E' and the simulator 'S'go through only via oracle queries that models attacker 'E' competence in a real attack. To break the security strong point of the private set intersection protocol, 'E' simulates subsequent queries.

Simulation of different random oracles:

Lemma 1. *Assume that the DDH (Decisional Diffie Hellman hypothesis) assumption holds for exponentiation, and hash function 'H' behaves like a random oracle, then the proposed hybrid protocol securely performs the 'Set Intersection' function described above.*

8.1.1 Simulation of Hash Query

Simulator 'S' maintains an initial empty hash list ${L^{List}}_h$ for the hash function h. The List maintains a tuple (x,P). On receipt of the hash query for an input 'x', 'S' will do a lookup operation.If the result exists, returns the same answer, else, it generates a random number g' ϵ G and returns g'. S' inserts (x,P) into the List.

8.1.2 Simulation of Authorize Query

Simulator 'S' on receiving the authorization or validation request i.e. to sign an element 'x' on behalf of certificate authority 'i' for a corrupted participant P_A (controlled by an attacker 'E'), 'S' makes a hash query on input (x, P_A) and on determining the hash value, 'S' computes the signature and returns the same to an attacker 'E'. (The simulator 'S' knows all the signing keys of all the protocol participants).

Table 1: Comparison of security features

Attacks/Protocols	Ours	[4]	[13]	[12]	[15]
Resists Semi-Honest Attack	Y	Y	Y	Y	Y
Resists Malicious Attack (Scenario 1)	Y	Y	Y	Y	N
Resists Malicious Attack (Scenario 2)	Y	N	N	N	N

Table 2: Comparison of complexity

Protocols	Computational Complexity	Communication Complexity
[10]	$O(m \log\log n)$	O(m+n)
[8]	O(R^2.n)	O(n^2)
[11]	O(R^2.n)	O(n.R)
[13]	2(N-1)(m+n)PM+2(N-1)DH	N-1)(m+n+5)
[12]	2(N-1)(m+n)PM	(N-1)(m+n+4)+6
[15]	2+m(n+1)+k(N+1)PM+(m(2+N)+n(2N+m+2)+2)EXP+(3N+1)Enc	O(m*n)
Proposed	2+m(n+1)+k(N+1)PM+(m(2+N)+n(2.N+m+2)+2)EXP+(3N+1)Enc	O(m*n)

8.1.3 Simulation of Set Intersection Query

Whenever an attacker 'E'submits a request to perform the set intersection protocol, S performs the following simulation. Assume that 'E'is imitating Alice as discussed in the above section.

'E' chooses an arbitrary random number some $A_1 \epsilon G$ and sends a set of encodings $MAS_1 = \{MA_1, MA_2, \cdots, MA_m\} = \{H(ID_A||a_1)^d.g^{A_1}, \cdots\}$ to S. S also chooses an arbitrary number B_1 computes the encodings, i.e. and directs $MBS_1 = \{MB_1, MB_2, \cdots, MB_k\}$ to 'E'. 'E'chooses an arbitrary number RA and computes the encodings $Z_A = g^{e.R_A} \bmod N$, $MBS_1{}^* = (MBS_1)^{e*R_A}$ = and submits $\{MB_1{}^{e.R_a}, MB_2{}^{e.R_a}, ...MB_m{}^{e.R_a}\}$ to S. 'E' performs arbitrary permutation $RPA = \zeta\{a_1, a_2, \cdots, a_m\}^{R_A} = \zeta\{a_1{}^{R_A}, \cdots, a_m{}^{R_A}\}$ and directs $MAS_2 = \{Z_A||MBS_1{}^*||RPA\}$ to S. 'S' chooses an arbitrary permutation $RPB = \zeta\{b_1{}^{R_B}, b_2{}^{R_B}, \cdots, b_k{}^{R_B}\}$ and sends $MBS_2 = \{ZB||MAS2^*||RPB\}$ to 'E'. It is not hard to see that S can compute all encodings, as it knows the secret signing keys ski for all the participants.'E'and S shares the random numbers used, i.e. A_1, B_1. Finally S and 'E' computes the intersection set.

It is clear that, except the attacker 'E' is able to fake or forge an encoding for an attribute, it does not possess a proper signature, then the joint output of all participants in the ideal world are identically distributed as similar to the proposed protocol. Assume that if 'E' did fake or forge an encoding for some element 'bi' which it is not validated or authorized by the TCA, then in the ideal world the protocol participants will filter out that attribute from the resulting set intersection, which results in the output distribution to be different in the ideal protocol from the proposed one.

9 Simulation and Experimental Evaluation

In this segment, we scrutinize the computational complexity of proposed and various related schemes through simulations.

9.1 Complexity Analysis

The computation cost is calculated based on the number of resource consuming 1024-bit multiplication, 1024-bit exponentiation and SHA-160 hash operations on mobile devices. The communication overhead is computed by the number of bits transmitted and received.

Table 1 confirms that our proposed scheme resists all major attacks both passive and active.

In Table 2, PM denotes a power modular; R denotes number of rounds; m, n denote number of Alice and participants attributes; EXP denotes an exponential operation; Enc denotes an encryption; N denotes number of participants.

Table 2 confirms that our scheme requires similar computational complexity compared to [7, 12] but negligibly higher complexity compared to the traditional schemes [8, 10, 11, 13]. But the overhead is perfectly valid, due to its security strengths. As discussed in the system architecture, we simulated our proposed scheme with a Samsung Galaxy J7mobile device consist of 1.5 GHz CPU. The simulation code is written in Java. We have considered the users $N = 5, 10, 20, 30, 40$ and each user is considered to contain varying attributes $k = 5, 10, 151$ (See Figure 2).

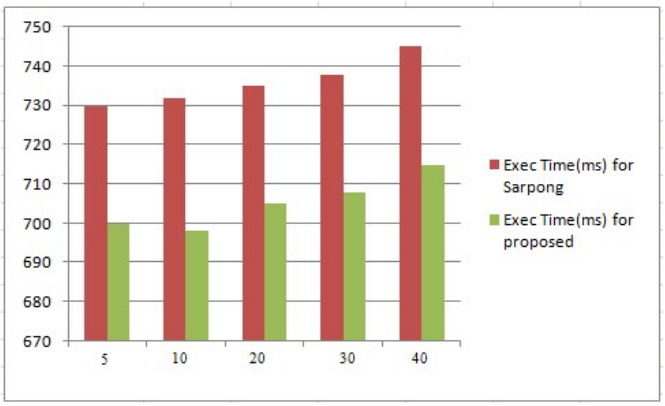


Figure 2: The simulation

10 Conclusion

The involvement of user's specific and sensitive data in MSN demands for a light weight and secure matchmaking algorithm, which resists attribute leakage to participants. Sarpong et al. had proposed first of its kind of matchmaking algorithm which selects the participants that contains the threshold level of attributes matching. We have cryptanalyzed Sarpong et al. scheme, and demonstrated that their scheme fails to achieve attribute privacy and requires huge storage and computation cost. We have proposed an efficient algorithm, which resists the pitfalls found in Sarpong et al. algorithm and other related schemes (static attribute representation). We also conducted experimental analysis of our scheme and illustrated the results.

References

[1] S. Y. Chiou, "Secure method for biometric-based recognition with integrated cryptographic functions," *BioMed Research International*, Vvol. 2013, Article ID 623815, 2013.

[2] S. Y. Chiou and C. S. Luo, "An authenticated privacy-preserving mobile matchmaking protocol based on social connections with friendship ownership," *Mathematical Problems in Engineering*, vol. 2014, Article ID 637985, 2014.

[3] X. Dong, "A multi-secret sharing scheme based on the CRT and RSA," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 47–51, 2015.

[4] A. Evfimievski, R. Agrawal, and R. Srikant, "Information sharing across private databases," in *Proceedings of ACM SIGMOD*, pp. 86–97, 2003.

[5] L. Guoy, J. Liu, R. Hao, B. Yangx, S. Jiang, X. Zhu, "Efficient private matching based on blind signature for proximity-based mobile social networks," in *IEEE International Conference on Communications (ICC'15)*, pp. 3246–3251, 2015.

[6] S. Huang, S. Griswold, K. Li, T. Sohn, "People-tones: A system for the detection and noti cation of buddy proximity on mobile phones," in *Proceedings of 6th IntConfon Mobile Systems (MobiSys'08)*, pp. 160–173, 2008.

[7] Y. H. Huang, S. H. Chiou, "Mobile common friends discovery with friendship ownership and replay-attack resistance," *Wireless Networks*, vol. 19, pp. 1839–1850, 2013.

[8] L. Kissner and D. Song, "Privacy-preserving set operations," in *Advances in Cryptology (CRYPTO'05)*, LNCS 3621, pp. 241–257, Springer, 2005.

[9] M. Li, N. Cao, S. Yu, W. Lou, "Findu: Privacy-preserving personal profile matching in mobile social networks," in *Proceedings of IEEE INFOCOM*, pp. 2435–2443, 2011.

[10] K. Nissim, M. Freedman, and B. Pinkas, "Efficient private matching and set intersection," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 1–19, 2014.

[11] J. Vaidya and C. Clifton, "Secure set intersection cardinality with application to association rule mining," *Journal of Computer Security*, vol. 13, no. 4, pp. 593–622, 2005.

[12] Y. Wang, T. Zhang, H. Li, L. He, and J. Peng, "Efficient privacy preserving matchmaking for mobile social networking against malicious users," in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 609–615, 2012.

[13] Q. Xie, U. Hengartner, "Privacy-preserving matchmaking for mobile social networking secure against malicious users," in *Ninth IEEE Annual International Conference on Privacy, Security, and Trust (PST'11)*, pp. 252–259, 2011.

[14] D. Xing, Y. Fang, H. Lin, S. S. M Chow and Z. Cao, *Privacy Preserving Friend Search over Online Social Networks*, Cryptology ePrint Archive, 2011. (`http://eprint.iacr.org/2011/445.pdf`)

[15] X. Zhang, S. Sarpong, C. Xu, "An authenticated privacy-preserving mobile matchmaking protocol based on social connections with friendship ownership," *International Jpurnal of Network Security*, vol. 17, no. 3, pp. 357–364, 2015.

Biography

K. Arthi have been working as a Associate professor of Department of Information Technology in VelTech Technical University, Avadi. He have completed my Ph.D in the area of Wireless Sensor Networks in Anna University, Chennai during 2015 and my ME (Embedded system technology) also from the same university during 2005. He completed my BE (IT) in Periyar University during 2002. He have published many National and International Journal Papers.

M. Chandramouli Reddy is a Research(Ph.D) Scholar, Department of Computer Science and Engineering in VelTech Technical University, Avadi. He have completed his

M. Tech (CSE) in JNT university, Hyderabad during 2006 and he completed his BE (CSE) in Madras University, Chennai during 1999. He have published many National and International Journal Papers.

Metamorphic Framework for Key Management and Authentication in Resource-Constrained Wireless Networks

Raghav V. Sampangi, and Srinivas Sampalli
(Corresponding author: Raghav V. Sampangi)
Faculty of Computer Science, Dalhousie University
6050 University Ave, PO Box 15000, Halifax, NS B3H 4R2, Canada
(Email: raghav.vs@ieee.org)
(Received Mar. 1, 2016; revised and accepted May 12 & June 10, 2016)

Abstract

The advent of the Internet of Things (IoT) has prioritized development in unique identification and sensing technologies, which facilitate IoT's automated and intelligent vision. Data security is critical to the success of such applications, more so with the communication over a wireless channel. However, IoT devices are resource limited and lack the ability to perform sophisticated computations without impacting their longevity requirements, or increasing the cost. This encourages creation of 'lightweight' security solutions for such low resource devices. We propose a new reconfigurable (or, metamorphic) framework for key management and authentication in this paper. Our framework deploys multiple lightweight algorithms and chooses one of them for each message exchange. We evaluate our work using assessment of key sequences, hardware resource utilization assessment and security analysis.

Keywords: Authentication, key management, reconfigurable security framework, resource-constrained wireless networks

1 Introduction

Several present day applications, including those in the military and healthcare, are built on the foundation of emerging wireless technologies. In particular, they employ radio frequency identification (RFID) and wireless body area networks (WBAN) as they facilitate unique identification and remote health monitoring. These technologies help with remote monitoring of military personnel, asset tracking in hostile territories and remote health monitoring of patients, among other applications. Advances in these technologies and in the capabilities of backend monitoring technologies such as cloud servers have further supported their use as central entities of the automated vision of the Internet of Things (IoT).

In RFID systems, electronic circuits called RFID tags store a unique identifier that helps uniquely identify any object, while details about such objects are stored in the backend server. These tags could be passive (without an on-chip power source) or active (with a power source). While the former are energized by the electromagnetic signals transmitted by an RFID reader (or interrogator) and respond to the reader queries, the latter can either respond to queries by the reader or initiate communications by themselves. A third category of tags, called semi-passive tags, have an on-chip power source, but still require the reader to initiate communication [12]. The lack of an on-chip power source imposes restrictions on the amount and type of computations that a passive tag can perform, making it resource-constrained. In WBAN systems, the on-body sensors that record data communicate with the hospital monitoring station through an on-body WBAN hub, and typically through a mobile device configured to be a personal server [17]. WBAN sensors could be required to stay on an individual for a long duration of time, depending on the application (especially in remote healthcare). This would require optimization techniques to ensure that the sensors can function for a longer time without frequent maintenance. One way to accomplish this is to configure the sensor nodes to 'sleep' and be woken up by the hub prior to communication [15]. This is similar to the function of passive RFID tags, and is also the main reason why WBAN sensors remain resource-constrained.

The restrictions on computational abilities of these devices necessitates a trade-off between cost, security and available resources. This further implies that data security solutions that can be deployed might be limited in their sophistication. Data security and privacy are critical in such applications, since the data can be uniquely identified with a specific individual [33]. Furthermore, with mode of communication being wireless, the communica-

tion can be 'snooped on'. A straightforward approach is to deploy cryptographic algorithms that are customized for application in low resource applications [20]. However, with most of the algorithms published and available in the public domain, the unpredictability and hence the security of an approach comes down to the strength of its keys.

We set out to determine *whether it was possible to create a framework that would constitute multiple algorithms for key management and authentication, and would enable dynamic choice of one of the deployed algorithms for each message communication – all this without any explicit communication phase for key exchange and agreement of the algorithm being used.* We wanted to explore the possibility of deploying such a set up in the context of resource-constrained wireless networks. In this paper, we discuss our metamorphic framework to accomplish such a functionality. Our approach (Section 3) is designed to support multiple algorithms for key management (and authentication), while facilitating context sensitive and deterministic choices of one of the available algorithms to accomplish several security goals. Although our framework is designed to be generic, applicable to all symmetric cryptosystems especially in resource-constrained wireless networks, we discuss a use case for the framework. The use case (Section 4) considers deploying two algorithms in the framework, namely — GeM2 key management and authentication mechanism based on gene mutation and transfer (Protocol B in [34]), and Butterfly1 (Butterfly key generation and encryption scheme [35]). To evaluate the framework with this use case, we consider key sequence assessment, hardware complexity assessment and security analysis (discussed in Section 5).

2 Related Work

The term *resource-constrained wireless networks* encompasses a wide variety of technologies and applications that are disparate, and have varying requirements. Such requirements could include resource (memory and computational ability) requirements, data storage requirements, but have a common security requirement. Since much of the resource-constrained wireless networks such as RFID systems, WBAN systems, Vehicular Ad-hoc Networks (VANETs) and the like, are created as autonomous systems to facilitate one independent activity in our lives, they are all in some way related to peoples' personal data. This places an emphasis on protecting data being communicated in such systems, thereby preserving the privacy of the individual(s) in question. We discuss some of the existing work in two resource-constrained systems, namely, RFID systems and WBAN applications.

Conventional systems rely on the following broad techniques for security — either on using shared secret keys and complex substitution/permutation functions as with symmetric cryptosystems, or on longer key-pairs and complex mathematical functions as with asymmetric cryptosystems [28]. However, restrictions in resource-constrained wireless networks limits the size of keys that can be used and the type of operations that can be performed, while ensuring security, longevity and keeping costs low. Each fundamental element of security, thus, needs to be customized and adapted for application in resource-constrained wireless networks. In this section, we discuss some of the mechanisms to accomplish key management and authentication resource-constrained wireless networks.

Pseudorandom number generators (PRNGs) are a popular choice for cryptographic algorithms for key generation in resource-constrained wireless networks. This is primarily due to their ability to generate unique sequences with different seeds. They can also generate sequences with large periods without repeating sequences [4]. Such algorithms have also been considered for deployment in RFID applications for key sequence generation [25, 26, 29]. Their work ranges from including nonlinear filter functions to ensure dispersion of bits in the pseudorandom sequence to varying feedback polynomials to generate pseudorandom sequences. It must be noted however, that PRNG-based techniques help accomplish only key generation and management, requiring them to be combined with other techniques such as hash-based or trusted-third party-based approaches for authentication.

Hash and key-ed hash algorithms are typically used with key generation systems, such as the ones discussed above, to accomplish authentication. One such approach [10] suggests the use of the new SHA-3 algorithm (Keccak algorithm) [3]. In their work, a combination of pseudorandom numbers, encryption keys and the RFID tag ID are used to compute message digests, and the encrypted key is updated on successful mutual authentication. Hashing algorithms are also employed by Hakeem et al. [13] for authentication in their proposal, where they use timestamps for key generation. Their work relies on two separate timestamps, one each generated at the server/reader and the tag. Their work also employs a linear feedback shift register (LFSR) to update keys. The first part of the protocol depends on each entity authenticating the other based on the difference in timestamps between the previous acknowledged timestamp and the current timestamp, and the XOR value of this timestamp difference with the secret tag key, k_t. Tag authentication by the server involves the tag sending a hash of its ID and the upper half of the secret key, K. Key updates at the server and the tag involves updating two secret keys and the timestamp, where the keys are updated using the previous values as seeds to the LFSR, while the current timestamp becomes the new stored timestamp value at the tag.

Hashes, especially keyed hashes, in particular are popular ways of accomplishing authentication in symmetric algorithms. Dong et al.'s work on RFID authentication [10] employs the new SHA-3 standard (Keccak algorithm) [3] to compute the message digests using a concatenation of pseudorandom numbers, keys and the tag ID. Pseudoran-

dom numbers are updated with each communication and are sent in the open, along with the hash containing an internally updated key. The key is updated on successful mutual authentication of the entities. The authors discuss various cases of operation, accounting for loss of tag acknowledgement messages and de-synchronization attempts. Hashing algorithms are also employed by Hakeem et al. [13] for authentication in their proposal, where they use timestamps for key generation.

Shi et al.'s work [36] exploits physical characteristics for security and (one-way) authentication in WBANs. Rather than having the sensors depending on cryptography for authentication, their work, BANA, considers using physical layer characteristics unique to the sensors; specifically, the variation in received signal strength (RSS) in the communication channel. The WBAN controller unit authenticates the on-body sensor nodes based on expected variations in received signal strengths of their individual responses and based on a threshold on the response time. The authors claim that attackers would experience "larger fluctuations due to multipath effect and Doppler spread than on-body sensors", making it a feasible authentication scheme. Mutual authentication among sensors or between sensors and the controller unit does seem to impact the limited resources, especially sensor battery life, in the long run, since BANA expects all sensors to compute the average RSS variations and authenticate other entities. This is mainly because authentication is an independent functionality in these sensors, which are required to include separate deployments of key management and encryption algorithms. Although the design of BANA is innovative in using physical channel characteristics for authentication, the need for separate implementation for key management implementation imposes an additional overhead on the resource-constrained sensors.

Message digests, digital signatures and third party certificates are common forms of accomplishing authentication among communicating entities. A different approach to accomplish this is a 'certificateless' manner, proposed by Liu et al. [23]. Their approach uses a trusted entity called the public key generator (PKG) that generates partial public-secret key pairs for each entity on the WBAN. Entities further request the PKG to generate the corresponding partial secret key using the entity's ID as the partial public key. The certificateless signature includes a message hash, the result of exponentiation operation applied on the public key of the PKG and the signer, its partial secret key and a random integer. This serves as a mechanism to accomplish authentication and message integrity verification. However, it is unclear whether this scheme is designed to authenticate entities on each update. This is because mutual authentication using public key infrastructure, regardless of how secure it is, will place an increased load on the already resource-constrained entity, whether it is a personal server or a sensor.

In a different take to key agreement and refresh, Zhu et al. [48] present a scheme that employs linked key updates, encrypted using the XTEA cipher (extended TEA [46]). In their work, keys are divided into 32-bit blocks and are updated block-wise on successful authentication. One thing to note is that their algorithm is prone to de-synchronization attacks, since the tag update is contingent on server authentication based on the server response, $m2$. If an adversary were to block $m2$ and transmit an unrelated $m2'$, the tag would not be able to authenticate the server, causing it to roll back its key update, thereby disrupting future communication.

Our discussion up to this point has focused on individual algorithms and combinations thereof to accomplish key management and authentication in RFID systems and WBAN applications. We next discuss algorithm frameworks, or a collection of algorithms used to accomplish a single or several security goals. A multi-algorithm encryption framework for active RFID tags has been proposed by Zhou et al. [47], an improvement of which is a generic optimized proposal for reconfigurable security co-processor work by Li et al. [21]. In their work, the control and data logic module chooses one of four encryption algorithms, namely AES, DES / 3DES (Data Encryption Standard), RSA (Rivest-Shamir-Adleman) public key cryptosystem, and ECC (Elliptic Curve Cryptography)-based cryptosystem. Their work is deployed in an FPGA (Field Programmable Gate Array)-based active RFID tag, where the design allows for reconfigurability and customization. The control / data logic module chooses the applicable encryption algorithm, in addition to the appropriate memory module, initializes the FPGA-based execution unit and performs the encryption. Their use of FPGA-based design is based on the reconfigurability rationale of the work by Jones et al. [18]. Jones et al. argue that a silicon-based implementation is not suggested for the design to be (re-)configurable. However, when we consider low resource devices such as passive RFID tags, one does not have any other option than implementing the custom security algorithms on silicon chips. Reconfigurability in such cases, can be accomplished by using hardware switches that can route data to the appropriate 'path' of the chosen algorithm for processing. This is the rationale we adopt in designing our framework for security.

The specified ISO/IEC 29167-1:2014 standard for RFID security [16] and IEEE 802.15.6 for WBANs [14] provide means for manufacturers of devices conforming to these standards to deploy multiple encryption algorithms on the devices as part of the respective security suites. From the available algorithms, the entities can select one for use for a particular session, during security association. When agreeing on the algorithm to be used, however, the choice is typically communicated in plaintext, available for an eavesdropper to learn about the system states. This reduces the overall uncertainty associated with the system. The approach we adopt, however, involves a random choice of one of the available algorithms, based on a previously agreed and synchronized timestamp, which increases the overall unpredictability and thus, the security of the system.

An appropriate mechanism to accomplish adaptive

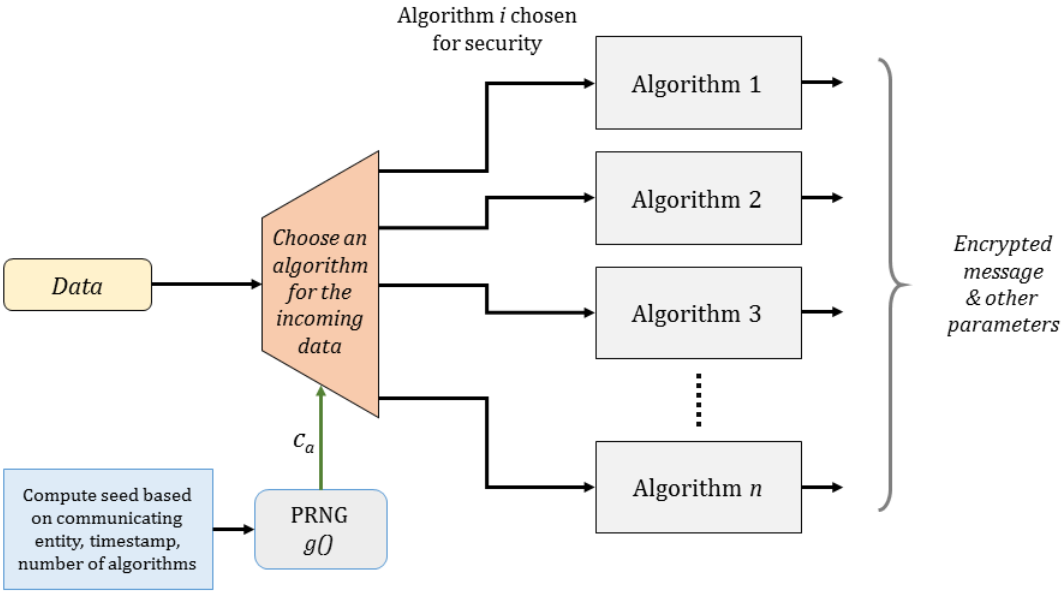


Figure 1: Overview of the multi-algorithm framework

computations to yield achieve high security, thus, is to use reconfigurable computing. Although reconfigurable computing is not new, with its applications being found in multimedia and other embedded systems [2, 24, 40], and even for securing the Internet (reconfigurable cryptography solutions for IPSec-based architectures [8]), we present its use in key management and mutual authentication. Our approach depends on the communicating entity and the timestamp, making it minimally context sensitive, and enables resource-constrained devices to dynamically choose one of the available algorithms for key management and authentication. Since it presents the perception of changing its structure with its dynamic algorithm choice, we refer to it as a *metamorphic* framework. We draw inspiration for our work from the functioning of a chameleon, which changes its color based on its surroundings. In the sections that follow, we discuss the proposed metamorphic framework, followed by presenting a use case for the same.

3 Proposed Metamorphic Framework for Key Generation and Authentication

Our work proposes a reconfigurable security framework for resource-constrained wireless networks with the main objective of accomplishing multiple security goals with simple logical operations. The central concept here is a mechanism for choosing one of the deployed algorithms for key generation and authentication, based on the contextual information. This approach is inspired by the functioning of a chameleon that changes its color based on the color of its surroundings.

Let us consider a conventional scenario with a system having one pre-defined algorithm for each aspect of security. Most systems use such an architecture and this works when all parameters other than encryption keys are pre-defined. In such cases, the uncertainty of the system operation remains limited. However, algorithms such as IPSec [11] are considerably better in the security than the former, with the entities choosing one of the pre-agreed algorithms with a security association phase just as the communication session begins. We derive motivation for our proposed framework from this aspect of being able to change algorithms and dynamically so, however, we remove the need for an explicit security association phase where the entities would agree on one of the available algorithms.

Figure 1 illustrates the overview of our proposed framework. Imagine that a system has N encryption schemes, each being a composite of algorithms to accomplish key management, encryption and authentication. Central to our framework is a mechanism to choose one of the available algorithms automatically and in a synchronous manner. We refer to this as the *algorithm choice logic*. This logic uses a unique combination of the ID (identifier) of the resource-constrained entity, the initial deploy-time timestamp (t_0) and an incrementing integer number, r_{ac}, in the range $0...(n-1)$, to determine which of the N schemes will be chosen to generate keys for data encryption and generate authentication parameters for a particular message transfer. The integer r_{ac} has a modulus of N, i.e. it 'wraps around' on reaching N (Equation (1)).

$$r_{ac} = (r_{ac} + 1) \bmod N. \quad (1)$$

The 'choice' aspect of the algorithm choice logic is accomplished by a pseudorandom number generator (PRNG), $g()$, that uses a combination of the ID, t_0 and r_{ac} as the seed. This seed, $seed_{ac}$, is generated as summarized by Equation (2). As the seed is always changing

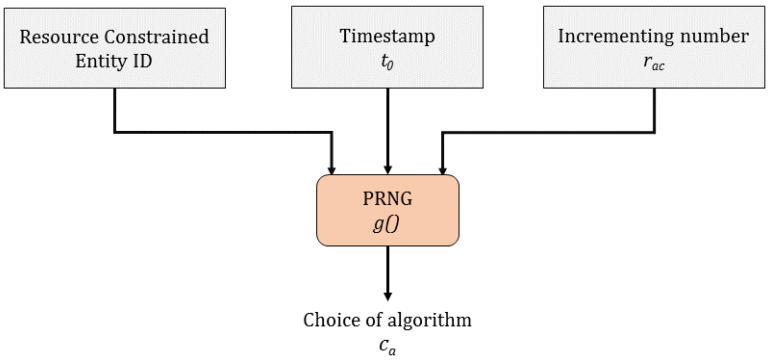


Figure 2: Illustration of algorithm choice process

in this approach, our approach ensures that the PRNG generates a new pseudorandom choice of the algorithm

$$seed_{ac} = ID \oplus t_0 \oplus r_{ac}. \quad (2)$$

Here, ID is the identification number associated with the resource-constrained entity (e.g. RFID tag or WBAN sensor); t_0 is the deploy-time timestamp; and, r_{ac} is the incrementing integer number, whose computation is explained by Equation (1). $\oplus$ represents the Exclusive-OR (XOR) operation.

The chosen algorithm, C_a, is determined by generating a pseudorandom number (PRN) using $seed_{ac}$ as the seed for $g()$. The number generated has a modulus of N.

$$C_a = g(seed_{ac}) \bmod N. \quad (3)$$

The deploy-time timestamp, t_0, is stored on the resource-constrained entity just prior to deploying it in its application environment. This is among the first pre-shared attributes, along with the ID and the initial encryption parameters associated with each algorithm. Furthermore, this will be updated during the course of operation as discussed later in this paper. This will ensure that the system state remains unpredictable to an observer. We employ the same timestamp in the algorithm chooser logic to re-use the stored and synchronized data, and to capitalize on the added uncertainty it provides. This specific combination of numbers, i.e. the timestamp, t_0, the incrementing number, r_{ac}, and the ID, changes continuously owing to increments in r_{ac} and at random with changes to t_0. This ensures that the N algorithms have a fair chance in being chosen for a specific encryption cycle. The operation of the seed generation and the corresponding PRN generation are illustrated by Figure 2. The function of the algorithm choice logic in Figure 1 is realized using C_a as the *select* input to choose one of the N algorithms that will be used to generate keys for message encryption and generate parameters for authentication.

When transmitting messages, our framework verifies the length of the message to be transmitted (M_{TI}). The length of the transmitted message is always considered to be the length (in bits, λ) of the longest message among all the algorithms. This ensures that messages are of consistent length and ensures uncertainty of the chosen algorithm.

If one entity cannot authenticate the other, it will start an internal counter to keep track of the number of erroneous messages or failed authentication attempts. If the next message results in a successful authentication, the internal counter is reset to 0, and communication proceeds as directed by the chosen algorithm. In case the counter reaches 3, flags the communication as erroneous and acts as required by the system implementation.

Our framework can be tweaked to include more (or reduced) choice in algorithms, depending on the application needs and the extent of constraints on the available resources. Thus, our framework has an implicit support to scalability, with minimal changes necessary to accommodate more algorithms in the framework. The changes would be in updating the algorithm chooser logic, specifically by updating N and a possible change to the circuit to extract C_a using modulo operation. Algorithm 1 summarizes the working of our framework algorithm (for a case when number of algorithms, $N = 3$).

4 Use Case

In this section, we discuss a use case for the framework, considering two previously published algorithms for key management and authentication [34, 35] as the constituent algorithms of the framework. We discuss customizing the framework for this use case and present a protocol of operation (that can be generalized for other use cases as well).

4.1 GeM2: Key Management and Authentication based on Gene Mutation and Transfer

GeM2 is a key management and authentication algorithm inspired by the mechanism of gene mutation and transfer in living organisms [34]. This features keys linked in a manner to the 'parent-child' relationship in organisms. Key update in GeM2 proceeds as follows — entities are

Algorithm 1 Algorithm choice & message transmission

BEGIN
Input: $rac \Leftarrow r_{ac}$, $t0 \Leftarrow t_0$, and $nAlg \Leftarrow 3$

$rac \Leftarrow (rac + 1)\ mod\ nAlg$ //Generate seed.

$seedAC \Leftarrow rac\ \oplus\ t0\ \oplus\ ID$

$CA \Leftarrow g(\ seedAC\ )\ mod\ nAlg$ //Choose algorithm.

if $CA = 0$ **then**
  Algorithm chosen = algorithm 1
  Perform any other related actions
else if $CA = 1$ **then**
  Algorithm chosen = algorithm 2
  Perform any other related actions
else if $CA = 2$ **then**
  Algorithm chosen = algorithm 3
  Perform any other related actions
end if

if $\lambda_{MTI} < \lambda_n$ **then**
  $MTX \Leftarrow MTI \parallel g(i)$
else if $\lambda_{MTI} = \lambda_n$ **then**
  $MTX \Leftarrow MTI$
else if $\lambda_{MTI} > \lambda_n$ **then**
  MTX consists of chunks of length, λ_{MTI}
end if

if $authentication = failed$ **then**
  $AuthFailCounter++$
  if $AuthFailCounter = 3$ **then**
    Flag error
  end if
else if $authentication = success$ **then**
  $transmit\_message\ (\ MTX\ )$
end if
END

initialized with an initial key and seeds for the PRNG. For each new key generation, a new seed is first computed using the linear recurrence formula [41], $seed_i = seed_{i-1} + seed_{i-2}$, i.e. by summing the previously used (or initially stored) seeds. Using this seed, a pseudorandom number ($numX$) is generated. A *mutation* pattern is then generated as follows — the '1' bits of the parent key (or initial key for the first key generation) are first 'preserved' by inverting the parent key, and performing an AND operation on this inverted parent and $numX$. This mutation pattern is then imposed on the parent by the XOR ($\oplus$) operation, i.e. $new\_key = parent\_key \oplus mutation$. With PRNG seeds being updated with each successful (acknowledged) authentication, $numX$ also used to generate the authentication-synchronization parameter (asv_i), computed as a hash of $numX$ XOR-ed with a special pattern called $pattern_{asv}$.

To add uncertainty to an otherwise straightforward key generation mechanism, GeM2 introduces random choices to update keys. The first random choice is at the beginning of key generation, where it checks whether to use the current parent key or update it (referred to as evolution of the parent). This is based on a random choice between 0 (continue with key generation) and 1 (parent evolution). Following this, a state identification parameter (g) is checked to see if it is equal to $genLimit$. $genLimit$ limits the number of 'child' keys for a parent to the specified number. If $genLimit = 5$, it means that a parent key can generate up to 5 keys before being forced to evolve. Of course, the uncertainty is in the combination of these two parameter checks, which implies that a parent key can either generate no child key or up to a maximum of $genLimit$ child keys. Each time a parent evolves, g is reset to 0, while the current parent number is identified by another state identifier (p). The parameters p and g for a given entity pair initialized with a common set of values, will specify the current state of the system, enabling only authorized entities to be able to determine the appropriate parameters given a specific state, (p_i, g_j). This facilitates key generation, as well as mutual authentication.

4.2 Butterfly1: Butterfly Key Management and Encryption

Butterfly encryption scheme [35], which we refer to as Butterfly1, employs a dynamic PRNG seed update technique that is naïvely based on the concept of Butterfly effect[1] [30]. In this seed update technique, the communicating entity chooses an integer, j at random ($0 \leq j \leq m$, with m being the length of the PRNG seed). The value of j determines the bit in the seed to be inverted. This single bit change will result in a completely new seed, which further results in a completely different set of pseudorandom sequences that are output from the PRNG.

The key management and authentication mechanism in Butterfly1 uses the Butterfly seed update mechanism, in addition to timestamp (t_i) and the updated value of the seed (s_j) to generate two keys for encryption. The first key, K_i, is computed as a pseudorandom number, generated using a seed that consists of an XOR combination of the updated seed and timestamp. The second key, K_T, is referred to as the transfer key and is computed as a pseudorandom number, generated using s_j as the seed.

Butterfly1 uses a multiple enveloping technique for enciphering the data. First, the data is XOR-ed with the updated seed, s_j, considered as the first envelope (resulting in an initial encrypted message, m_i). Following this, m_i is encrypted using K_i using a symmetric key algorithm (XOR is used in [35] due to its involutory property, i.e. it is its own inverse) generating the intermediate ci-

[1] *Butterfly effect* is a concept in chaos theory, defined by Polinar [30] as "hypersensitivity to perturbation". This means that in a non-linear deterministic system, if the initial conditions are changed ever so slightly, there will be drastic changes in the output of a later state.

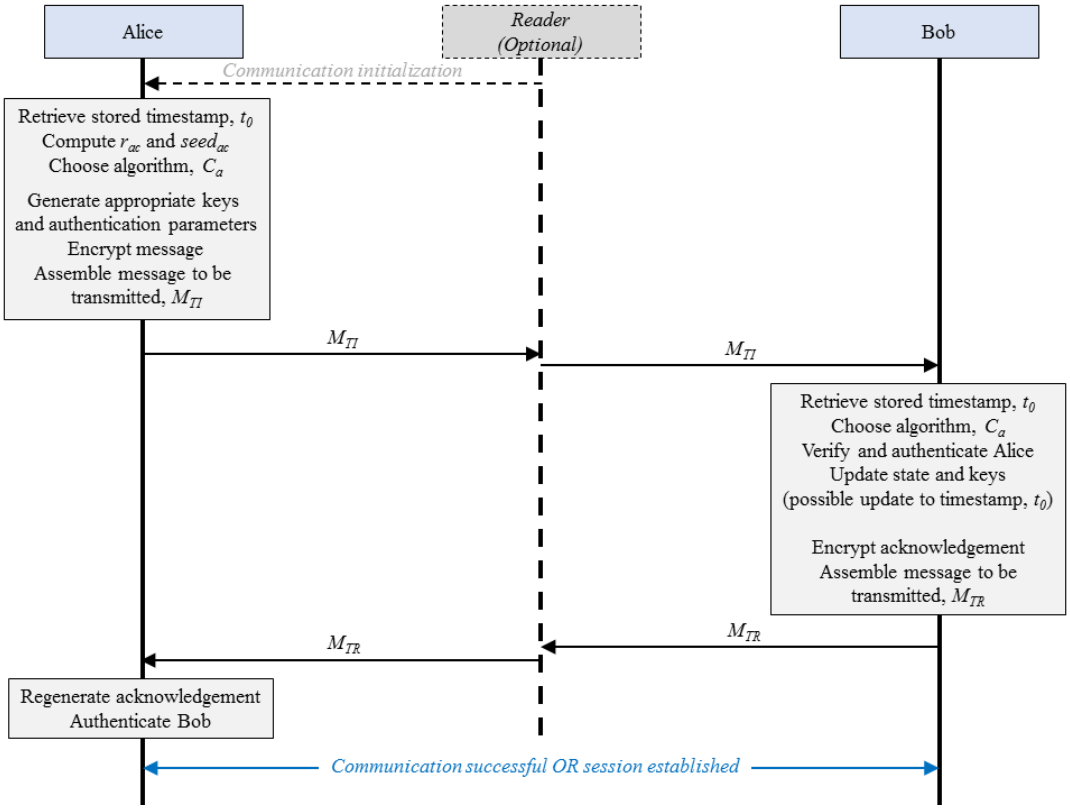


Figure 3: Framework: Protocol of operation

phertext, c_i. Butterfly1 also uses an XOR combination of the updated seed (s_j), the initial encrypted message (m_i) and the timestamp (t_i) as the seed for a PRNG, using the generated pseudorandom number as the message signature, θ_i. Next, the concatenation of c_i, t_i, the message sequence number (i), and θ_i, is encrypted using K_T (using the same symmetric key algorithm used for the earlier encryption using K_i). This becomes the final envelope over the data. Butterfly1 relies on changing values of the timestamp, the updated seed and hence the keys (and message signatures) to generate continuously changing parameters that are specific to the context (time and synchronized entity).

4.3 Using GeM2 and Butterfly1 in the Proposed Framework

We considered GeM2 and Butterfly1 as an initial attempt to explore the feasibility of our proposed framework. Although they are stand alone security proposals, with completely different rationales and mechanisms of operation, we explore this combination as a starting point to understand the behavior of the framework with such algorithms. This becomes a stepping stone towards exploring the the feasibility of using standard algorithms, Diffie-Hellman [9] for key generation, AES or DES for encryption and combinations thereof, in our framework.

In the algorithm choice logic (discussed in Section 3), the value of n will be set to 2 in the computation of r_{ac} (as defined in Equation (1)). The PRNG seed, $seed_{ac}$, would then depend on the ID of the entity (RFID tag, WBAN sensor, RFID server or WBAN hub), and t_0. The mechanism to update the timestamp t_0 is discussed as part of the protocol of operation. The protocol of operation is as follows for a communication of the i^{th} message between Alice and Bob (Figure 3):

Step 1: Alice retrieves the previously acknowledged and synchronized timestamp, t_0, and increments the stored value of r_{ac}, and computes the algorithm choice, C_a using Equations (1), (2) and (3).

Step 2: The algorithms are chosen as per Equation (4).

$$C_a = \begin{cases} 0 & \text{algorithm chosen} = \text{GeM2} \\ 1 & \text{algorithm chosen} = \text{Butterfly1} \end{cases} \quad (4)$$

Step 3: Alice computes the encryption keys and associated authentication parameters as explained in Sections 4.1 and 4.2 as specified by the chosen algorithm.

Step 4: Determining the length of the message (M_{TI}): In this implementation, the longest message transmitted is by Butterfly1. We consider this length as λ_n. The framework makes a decision on the transmitted message (MTX) based on Equation (5).

$$MTX = \begin{cases} M_{TI} & \text{if } \lambda_{M_{TI}} = \lambda_n \\ M_{TI} \parallel g(i) & \text{if } \lambda_{M_{TI}} < \lambda_n. \end{cases} \quad (5)$$

Here, $g(i)$ is a pseudorandom number generated using i as the seed (i being the sequence number in Butterfly1), only to generate numbers (that have no meaning for the algorithms) for padding the message to be transmitted so as to make it be of the same length as the longest message. In case messages in Butterfly1 (or GeM2) become longer than λ_n, they are broken into separate messages and transmitted using the same mechanism.

Step 5: On receiving the i^{th} message transmitted by Alice, Bob retrieves the timestamp, t_0 and performs the same computations as Alice to generate the keys and authentication parameters. The message integrity verification and entity authentication then proceeds as specified by the chosen algorithm.

Step 6: If Bob can authenticate Alice, Bob may respond with an encrypted acknowledgement/response as specified by the implementation, which could lead to session establishment or conclusion of a message transfer. On successful authentication, Alice and Bob have synchronized states.

Step 7: If Bob cannot authenticate Alice, Bob uses the internal counter to track of the number of erroneous messages or failed authentication attempts. This counter is reset on a successful communication, but Bob will flag the communication as erroneous if the counter reaches 3.

Step 8: Our framework also allows Alice and Bob to agree on the duration of each security association, or the duration for a single long-term communication session. The initial timestamp, t_0, is updated with the last synchronized timestamp at the end of such a security association. In this implementation, the security association period was set to be the choice of Butterfly1 algorithm. This means that whenever the framework chooses Butterfly1, t_0 is updated, i.e. $t_0 = t_i$. Therefore, when encrypting the current message, the processes to generate the key and other parameters would use the previously stored (and acknowledged) value of t_0, and update this value whenever the Butterfly1 algorithm is used. This ensures that there is no fixed duration of a security association, which adds another layer of uncertainty and consequently improves the security of the communication. However, if a system implementation requires a consistent duration for security association, the framework can be updated with minimal changes to set an internal timer that determines when the security association ends, thereby forcing a timestamp update.

Our framework is thus, able to utilize the best possible options from the available algorithms to ensure security. By retaining the length of the transmitted message to be a constant, our framework introduces an additional element of unpredictability to an observer.

5 Evaluation and Results

We evaluate our proposed metamorphic framework using three analyses. Our analyses are in the context of the use case discussed in Section 4, with GeM2 and Butterfly1 as constituent algorithms of the proposed framework. Although our framework proposal is generic, this gives us a way to provide some context to the assessment. Our evaluation of the framework involved three parts:

Key Sequence Evaluation: We implemented our framework (along with GeM2 and Butterfly1) using Java programming language to verify the working of the concept and generation of key sequences for further evaluation. We used the generated key sequences to test similarity between consecutive keys using Sörensen's Similarity Index (SSI) [37]. Using the results so obtained, we compared its performance with the RFID security proposals by Zhu et al. [48] and Dong et al. [10], and with an AES-based key generation approach for WBANs proposed by Liu et al. [22]. Note that we also implemented the proposals by Zhu et al., Dong et al., and Liu et al. using Java to generate key sequences for our assessment.

Hardware Complexity Evaluation: We estimated the approximate resource requirement for implementing the proposed framework (along with GeM2 and Butterfly1) encryption scheme on hardware.

Security Evaluation: We also performed a security assessment using Scyther protocol analyzer [6, 7] and qualitative security analysis to evaluate the security of our proposal.

5.1 Key Sequence Evaluation

We first implemented GeM2 and Butterfly1 using Java, followed by using these in the implementation of the framework. In our implementation of GeM2, the initial key was set to $92EB8D6ECF7F808A705D1A4566991AF0$, the initial seeds to compute the PRNG seed were set to 14930352 and 24157817. For Butterfly1, the PRNG used to choose the value of the variable j at random, which decides the state of the seed (s_j), was initialized to $192BC333250CCCFF$, while the seed (s) itself was initially set to 12345678. We used the Java Random class to introduce random delays (0 and 2 seconds) between consecutive key generations, as an attempt to emulate real-time communication, and used methods in the Random class to extract PRNG sequences. Furthermore, we used the Java method, $System.currentTimeMillis()$ to extract the timestamp. We extracted 10240 key sequences for our assessment.

A strong cryptosystem needs to have a key management mechanism that is strong and able to generate and refresh keys in a manner that can perplex adversaries, for

Table 1: Summary of similarity between keys

Configuration	Average SSI (SSI_{av})
Proposed framework* (K, K_i)	0.3437
GeM2* (K)	0.3040
Butterfly1* (K_i)	0.3809
Liu et al. [22]	0.3826
Zhu et al. [48]	0.4110
Dong et al. [10]	0.3815

$*$: Only keys in parentheses were considered for analysis.

a cryptosystem is accepted to be only as strong as the keys used (Kerchkoff's Principle [39]). This led us to verify how similar consecutive keys generated by each algorithm and the framework are, since GeM2 and Butterfly1 use varying logical operations to accomplish the desired functionality. To evaluate similarity between keys, we considered keys generated by the system and compare pairs of keys. We quantified the similarity between keys using SSI, which is a measure of how similar the various pairs of keys are, i.e. it is the ratio of twice the total similar characters in the two keys to the total size (in characters) of each key. Equation (6) summarizes the computation of SSI.

$$SSI = \frac{2 \times n(A \cap B)}{n(A) \ + \ n(B).} \tag{6}$$

Here, $n(A \cap B)$ represents the number of characters (or, numbers) in the key pair that are same, $n(A)$ and $n(B)$ represent the total number of characters (or, numbers) in each of the keys A and B of the key pair, respectively. The expectation is to have key similarity sufficiently low, with an average SSI value in the vicinity of 0.30, or at least less than 50%. This is to ensure that keys do not appear to have obvious similarity or patterns, which could be exploited by adversaries to derive keys.

Table 1 summarizes the average SSI values for the proposed framework, GeM2, Butterfly1, and the proposals by Liu et al. [22], Zhu et al. [48], and Dong et al. [10].

We can observe that our framework is able to generate keys that are less similar to each other, compared to the other algorithms, while GeM2 performs better than the framework.

One thing to note is that when used appropriately, the framework is able to combine the best attributes of the algorithms and contribute to making the system improve overall in terms of security. This can be observed with the slightly high similarity in keys generated by Butterfly1, which when combined with GeM2 in the framework, performs better. Furthermore, random choices of algorithms for each communication also ensures that the overall unpredictability remains high. This, in addition to low similarity between keys, improves the security. In our continuing work, we will work to ensure that the unpredictability of keys is not dependent on the best algorithm available in the framework, but that the framework itself will ensure high unpredictability of keys.

5.2 Hardware Complexity Evaluation

To test resource utilization, we developed a behavioral model of our framework using VHDL (Very High Speed Integrated Circuit - VHSIC - Hardware Description Language) [32] and deployed it on the Xilinx Spartan-6 FPGA (Field Programmable Gate Array) SP605 Embedded Kit [42]. Logical operations on an FPGA are accomplished sing configurable logic blocks (CLBs) and programmable interconnects. Spartan-6 has two slices, with each slice composed of "four logic-function generators (or look-up tables, LUTs) and eight storage elements" [43]. We implemented a modified version of the J3Gen PRNG (Melià-Seguí et al. [26]) and SHA-1 message digest algorithm (Rainier [31]). We used these implementations to realize the various functionalities to estimate resource consumption, and our proposal is generic, and designed to work with any encryption, PRNG and message digest algorithms.

Hardware resource estimation helps understand the overall resource utility when algorithms are implemented on application specific integrated circuits (ASICs). This is based on the logic block utilization (i.e. LUTs, Flip-Flops, etc.). As observed in Table 2, the resources consumed by our framework implementation are considerably low than available on the FPGA. The number of logic cells consumed can be used to determine the approximate logic gates for the implementation. A logic cell is a "logical equivalent of a classic four-input LUT and a Flip Flop" [44], resulting in 1.6 logic cells per LUT in Spartan-6 [43], or approximately 6.4 logic cells per slice. Using the formula, 1 *slice* $\approx$ 6.4 *logic cells*, we estimate the number of logic cells. Furthermore, it has been estimated that 1 *logic cell* $\approx$ 15 *ASIC gates* [45]. Our framework requires approximately 960 ASIC gates for a key size of 128 bits, where as PRESENT [5] and Grain128 [1], which are also included in the cryptographic suite specifications as part of ISO/IEC 29167-1:2014 [16], require 1570 and 1857 gates, respectively (as reported in their respective publications, results summarized in Table 3). This lets us ascertain that the overhead of our proposals on resource-constrained devices or other devices will be considerably less as compared to other approaches. Furthermore, our gate count estimate is very much within the range of 200 - 3000 gates, which is suggested to be the available gates for security on resource-constrained devices [19, 27].

5.3 Security Evaluation

We performed a security assessment of the protocol of operation (Figure 3) using Scyther protocol analyser [6, 7]. The Scyther tool allows us to perform a formal security analysis of the communication protocol [7], verifying it using the Dolev and Yao adversary model [6, 28], which assumes perfect cryptography, abstract messages and that the adversary has full control over the network. In this section, we discuss the results obtained from this analysis (summarized in Table 4), and also elaborate on how these

Table 2: Logic resource utilization

Device Utilization Summary	
Slice Registers (available:54,576)	17
Slice LUTs (available:27,288)	19
Occupied Slices (available:6,822)	10
MUXCYs used (available:13,644)	16
LUT Flip Flop pairs (/available)	13 (/22)
Bonded IOBs (available:296)	5
BUFG/BUFGMUXs (available:16)	1
Approximate Logic Cell and ASIC Gate Equivalent	
Logic Cells (available:43,661)	64
ASIC Gate Equivalent	960

Table 3: Gate count estimates

HiveSec	PRESENT [5]	Grain128 [1]
960	1570	1857

Table 4: Evaluation of the proposed framework using Scyther

Claim	Result	
	Initiator	Responder
Claim: *Secret*		
Secret Parameters‡	$NAWB^*$	$NAWB$
Claim: *Alive*	$NAV^\dagger$	NAV
Claim: *Weakagree*	NAV	NAV
Claim: *Nisynch*	$NAWB$	$NAWB$
Claim: *Niagree*	$NAWB$	$NAWB$

$NAWB^*$: No attacks, within bounds
$NAV^\dagger$: No attacks, verified
‡: The following parameters were expected to remain secret — all keys in the algorithms timestamp, message signatures, authentication-synchronization vector, and intended message

results impacts how the framework performs with respect to standard security goals [38] — confidentiality, integrity, authentication, non-repudiation (by association) and forward / backward secrecy.

Claim-1: Secrecy. Parameters expected to be secret in each configuration remain a secret, including and primarily the key generation parameters and intended messages. This claim holds true since none of these parameters are exchanged. This further implies that the framework configurations ensure *confidentiality* of all parameters. Furthermore, with the communicated parameters also including message signatures, the entities are able to verify *message integrity*.

Claims-2 and 3: Alive and Weakagree. Scyther validates our claims that the entities are running the same configuration (Weakagree) and all previous message sessions have used the proposed scheme (Alive).

Claim-4: Non-injective Synchronization (Nisynch). Our claim that the initiator and responder states are synchronized in the framework is also verified. Synchronization is dependent on the internal states of the system, which requires that each algorithm states be synchronized, and entities are able to recognize any attempts of de-synchronization. This guarantees protection against *replay* and *de-synchronization* attacks.

Claim-5: Non-injective Agreement (Niagree). The initial deploy-time parameters, such as the initial seeds and timestamp, are never exchanged in the open. Each synchronized update of the system states imply that these parameters are automatically updated, at times through encrypted messages. Thus, the internal parameters that are essential in computing the key materials are always in agreement in both entities, as long as they are synchronized and authenticated. Scyther validates that the framework configurations are synchronized, and that they are in agreement.

Authentication and Non-repudiation. Both algorithms used in the framework facilitate authentication, using message signatures and the authentication synchronization vector. This ensures that the system states are synchronized (established by Scyther assessment) and that the entities are (mutually) authenticated. Furthermore, the use of pre-shared parameters and random choice of one of the available algorithms means that the sender of each message cannot deny that it was sent from that particular entity. Since keys are updated and potentially a different algorithm is chosen for each key generation/encryption, it means that the internal states of each entity (for each algorithm) are always updated to the latest (synchronized) version, as long as they are authenticated. This, in a naïve manner helps the framework accomplish non-repudiation by association[2].

Forward and Backward security. To an observer without the knowledge of the internal states, the framework as a whole appears as a black-box sequence generation engine. This means that it appears to be a sequence generator that generates various keys and other parameters required to encrypt and sign messages. Unless the observing entity has a knowledge of the internal states and the algorithms chosen, knowledge of a contiguous set of keys either from the past or in the future would not yield useful information about the future keys or previously used keys, respectively. This is primarily

[2]Non-repudiation by association implies accomplishing this goal by being associated with a backend server, which can be authenticated by another trusted entity using trusted third parties and digital signatures.

due to the dependency on timestamp in the choice of algorithms as well as updates to internal states of the framework. This re-configurable or metamorphic property of our framework not only provides high security, but guarantees forward and backward secrecy as well.

6 Concluding Remarks

Motivated by the need for security proposals that consume less resources for computation, while providing high security by means of increased unpredictability, our work proposed a metamorphic (or, reconfigurable) security framework. We draw inspiration for our work from the manner in which a chameleon changes its color in response to the color of its surroundings. Since resource-constrained devices would require circuits to be predefined at deploy time, our framework is based on using multiple security solutions that help in accomplishing reconfigurability in its operation. Our framework uses a synchronized value of the timestamp, an incrementing integer and the ID of the resource-constrained entity for choosing one of N available algorithms.

Unpredictability is an attribute that is perhaps central to our framework. This is an aspect that defines the security of our proposals, since security of a published cryptographic technique is dependent on the nature of the keys used. This is defined by how related keys are to preceding and subsequent keys. Our assessment (Section 5) helps us establish that the keys, although seemingly related in the individual algorithms, are made even more unpredictable by the presence of the framework. This added layer of uncertainty ascertains that it remains hard for unauthorized entities to crack the system keys, keeping the overall security high. Our assessment also supports our claim of the proposal being suitable for resource-constrained applications, although this will depend on the attributes of other algorithms that may be deployed as part of the framework in other configurations.

The presence of multiple algorithms in a framework also means that the effect of any sub-optimal performance by one algorithm in the framework can be mitigated by the presence of other algorithms. Our results support our claims of increased security through unpredictability, while requiring less resources for ASIC implementations of our algorithms and the framework.

With its ability to combine the attributes of the available algorithms and being able to choose one at random, our work removes the need for a separate algorithm agreement phase in communication. Security is critical in wireless communications, especially in resource-constrained wireless networks with their added restrictions. When configured appropriately, our framework presents each resource-constrained device with an opportunity to accomplish several security goals, including mutual authentication and non-repudiation by association.

Our work was motivated by a need to remove key exchange messages, while facilitating resource-constrained devices to be able to dynamically choose from a set of available algorithms for various aspects of security; considering that the presence of security suites is also being suggested by the updated standard specifications [16, 15]. Our approach not only facilitates multiple algorithms, but the facility to choose one based on the specific context of a message exchange. This ability of dynamically being able to choose algorithms is similar to the framework reconfiguring itself with each message. The reconfigurable behavior makes our framework *metamorphic*, giving an illusion as though the framework is changing its structure at random. In our continuing work, we will consider extending our framework to evaluate it with more (and diverse) algorithms and to perform a critical complexity evaluation of the framework. While our focus in this work has been to develop a framework for application in resource-constrained wireless networks, the design of the framework is generic, allowing it to be extended and adapted for use in other non-resource-constrained application environments as well.

Acknowledgments

This work has been funded by the Boeing Company. The authors gratefully acknowledge the support and the feedback given by the company.

References

[1] M. Ågren, M. Hell, T. Johansson, and W. Meier, "Grain-128a: A new version of grain-128 with optional authentication," *International Journal of Wireless Mobile Computing*, vol. 5, pp. 48–59, Dec. 2011.

[2] I. Beretta, V. Rana, M. D. Santambrogio, and D. Sciuto, "On-line task management for a reconfigurable cryptographic architecture," in *IEEE International Symposium on Parallel Distributed Processing (IPDPS'09)*, pp. 1–4, May 2009.

[3] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, *The Keccak Sponge Function Family*, July 15, 2016. (`http://keccak.noekeon.org/`)

[4] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on Computing*, vol. 15, no. 2, pp. 364–383, 1986.

[5] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems (CHES'07)*, LNCS 4727, pp. 450–466, Springer Berlin Heidelberg, 2007.

[6] Cas Cremers, *The Scyther Tool*, Apr. 4, 2014. (`https://www.cs.ox.ac.uk/people/cas.cremers/scyther/`)

[7] Cas Cremers, *Scyther User Manual*, Department of Computer Science, University of Oxford, 2014. (`http://documents.mx/documents/scyther-manual.html`)

[8] A. Dandalis and V. K. Prasanna, "An adaptive cryptographic engine for internet protocol security architectures," *ACM Transactions on Design Automation of Electronic Systems*, vol. 9, pp. 333–353, July 2004.

[9] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644–654, Nov. 1976.

[10] Q. Dong, J. Zhang, and L. Wei, "A SHA-3 based RFID mutual authentication protocol and its implementation," in *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC'13)*, pp. 1–5, Aug. 2013.

[11] B. A. Forouzan, *Data Communications and Networking*, McGraw-HilI Forouzan Networking Series, 2007.

[12] B. Glover and H. Bhatt, *RFID Essentials*, O'Reilly Media, First edition, 2006.

[13] M. J. Hakeem, K. Raahemifar, and G. N. Khan, "A novel key management protocol for RFID systems," in *9th International Wireless Communications and Mobile Computing Conference (IWCMC'13)*, pp. 1107–1113, July 2013.

[14] IEEE Standards Association, "IEEE 802.15: Wireless Personal Area Networks (PANs),", 2005.

[15] IEEE Standards Association, '*IEEE Standard for Local and Metropolitan Area Networks - Part 15.6: Wireless Body Area Networks*, IEEE Standard 802.15.6-2012, pp. 15–172, 2012.

[16] ISO/IEC, *Information Technology – Automatic Identification and Data Capture Techniques – Part 1: Security Services for RFID Air Interfaces*, International Standard ISO/IEC 29167-1: 2014, pp. 10, Aug. 2014.

[17] A. F. Jaimes and F. R. de Sousa, "A taxonomy for learning, teaching, and assessing wireless body area networks," in *IEEE 7th Latin American Symposium on Circuits Systems (LASCAS'16)*, pp. 179–182, Feb. 2016.

[18] A. K. Jones, R. Hoare, S. Dontharaju, S. Tung, R. Sprang, J. Fazekas, J. T. Cain, and M. H. Mickle, "An automated, FPGA-based reconfigurable, low-power RFID tag," *Microprocess. Microsystems*, vol. 31, pp. 116–134, Mar. 2007.

[19] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols,". in *Advances in Cryptology (CRYPTO'05)*, LNCS 3621, pp. 293–308, Springer, 2005.

[20] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[21] C. Li, J. Zhou, Y. Jiang, C. Chen, Y. Xu, and Z. Luo, "A reconfigurable and scalable architecture for security coprocessor," in *5th IEEE Conference on Industrial Electronics and Applications (ICIEA'10)*, pp. 1826–1831, June 2010.

[22] J. Liu and K. S. Kwak, "Hybrid security mechanisms for wireless body area networks," in *Second International Conference on Ubiquitous and Future Networks (ICUFN'10)*, pp. 98–103, June 2010.

[23] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 332–342, Feb. 2014.

[24] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Techniques for design and implementation of secure reconfigurable PUFs," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 2, pp. 5:1–5:33, Mar. 2009.

[25] H. Martin, E. S. Millan, L. Entrena, P. P. Lopez, and J. C. H. Castro, "AKARI-X: A pseudorandom number generator for secure lightweight systems," in *IEEE 17th International On-Line Testing Symposium (IOLTS'11)*, pp. 228–233, 2011.

[26] J. Melia-Segui, J. Garcia-Alfaro, and J. Herrera-Joancomarti, "Multiple-polynomial LFSR based pseudorandom number generator for EPC gen2 RFID tags," in *37th Annual Conference on IEEE Industrial Electronics Society (IECON'11)*, pp. 3820–3825, 2011.

[27] M. O'Neill (nee McLoone), "Low-cost SHA-1 hash function architecture for RFID tags," in *Proceedings of the Workshop on RFID Security (RFIDsec'08)*, pp. 1–11, 2008.

[28] C. Paar and J. Pelzl, *Understanding Cryptography*, Springer Berlin Heidelberg, 2010.

[29] P. Peris-Lopez, E. S. Millan, Jan C. A. van der Lubbe, and L. A. Entrena, "Cryptographically secure pseudo-random bit generator for RFID tags," in *International Conference for Internet Technology and Secured Transactions (ICITST'10)*, pp. 1–6, 2010.

[30] D. Poulin, "A rough guide to quantum chaos,", 2006. (`http://www.physique.usherbrooke.ca/poulin/utilisateur/files/enseignement/rgtqc.pdf`)

[31] J. Rainier, "SHA-1 sequential implementation," *Github*, 2014.

[32] C. H. Roth, Jr., *Digital Systems Design Using VHDL*, PWS Publishing Company, 1998.

[33] H. Saini, "1-2 skip list approach for efficient security checks in wireless mesh networks," *International Journal of Electronics and Information Engineering*, vol. 1, no. 1, pp. 9–15, 2014.

[34] R. V. Sampangi and S. Sampalli, "RFID mutual authentication protocols based on gene mutation and transfer," *Journal of Communications Software and Systems*, vol. 9, pp. 44, Mar. 2013.

[35] R. V. Sampangi and S. Sampalli, "RFID encryption scheme featuring pseudorandom numbers and butterfly seed generation," in *22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM'14)*, pp. 128–132, Sept. 2014.

[36] Lu Shi, M. Li, S. Yu, and J. Yuan, "BANA: Body area network authentication exploiting channel characteristics," *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 1803–1816, Sept. 2013.

[37] T. Sorensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons," *Biologiske Skrifter Kongelige Danske Videnskabernes Selskab*, vol. 5, no. 4, pp. 1–34, 1957.

[38] W. Stallings, *Cryptography and Network Security: Principles and Practice*, Pearson Prentice Hall, 2010.

[39] W. Trappe and L. C. Washington, *Introduction to Cryptography with Coding Theory*, Pearson Prentice Hall, 2006.

[40] N. S. Voros, M. Hübner, J. Becker, M. Kühnle, F. Thomaitiv, A. Grasset, P. Brelet, P. Bonnot, F. Campi, E. Schüler, H. Sahlbach, S. Whitty, R. Ernst, E. Billich, C. Tischendorf, U. Heinkel, F. Ieromnimon, D. Kritharidis, A. Schneider, J. Knaeblein, and W. Putzke-Röming, "MORPHEUS: A heterogeneous dynamically reconfigurable platform for designing highly complex embedded systems," *ACM Transactions on Embedded Computing Systems*, vol. 12, pp. 70:1–70:33, Apr. 2013.

[41] E. W. Weisstein, "Linear recurrence equation," *MathWorld–A Wolfram Web Resource*, 2012.

[42] Xilinx, *Getting Started with the Spartan-6 FPGA SP605 Embedded Kit*, Xilinx Inc., June 2010.

[43] Xilinx, *Spartan-6 FPGA Configurable Logic Block: User Guide*, Xilinx Inc., Feb. 2010.

[44] Xilinx, *7 Series FPGA Configurable Logic Block: User Guide*, Xilinx Inc., Nov. 2014.

[45] Xilinx, "All programmable low-end portfolio product selection guide,", 2014.

[46] J. Yu, G. Khan, and F. Yuan, "XTEA encryption based novel RFID security protocol," in *24th Canadian Conference on Electrical and Computer Engineering (CCECE'11)*, pp. 58–62, May 2011.

[47] J. Zhou, Y. Xu, and X. Li, "Reconfigurable and scalable security module of active RFID for security-sensitive applications," in *The 2nd IEEE International Conference on Information Management and Engineering (ICIME'10)*, pp. 135–140, Apr. 2010.

[48] G. Zhu and G. N. Khan, "Symmetric key based RFID authentication protocol with a secure key-updating scheme," in *26th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'13)*, pp. 1–5, May 2013.

Biography

Raghav V. Sampangi Dr. Raghav V. Sampangi is a Postdoctoral Fellow at the Faculty of Computer Science, Dalhousie University, Canada. His research interests include security, privacy, and usability in Context-Aware Systems and the Internet of Things. He has worked on reconfigurable security in emerging wireless networks such as RFID and wireless body area networks (WBAN). Currently, he focuses on key generation and authentication in resource-constrained devices, and usable security in Context-Aware Systems and the Internet of Things.

Srinivas Sampalli Dr. Srinivas (Srini) Sampalli is a professor and 3M National Teaching Fellow in the Faculty of Computer Science, Dalhousie University, Halifax. His research is in emerging wireless technologies, especially in the intersection of smartphones, near field communications (NFC) and mobile cloud computing. He has investigated protocol vulnerabilities, security best practices, risk mitigation and analysis, design of intrusion detection and prevention systems, and applications in healthcare and mobile commerce. His projects have been sponsored by NSERC, Industry Canada and NRC. Dr. Sampalli has received many teaching awards at the Faculty, University, provincial and national levels, including a named teaching award and 3M National Teaching Fellowship, Canada's most prestigious teaching acknowledgment.

A Publicly Verifiable Authenticated Encryption Scheme Based on Factoring and Discrete Logarithms

Cheng-Yi Tsai[1], Chi-Yu Liu[1], Shyh-Chang Tsaur[2,3], and Min-Shiang Hwang[1,4]
(Corresponding author: Min-Shiang Hwang)

Department of Computer Science and Information Engineering, Asia University[1]
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung 41354, Taiwan
Department of Electronic Engineering, National Chin-Yi University of Technology[2]
No.57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan
Department of Business Administration, Tunhai University[3]
No.1727, Sec.4, Taiwan Boulevard, Xitun District, Taichung 40704, Taiwan
Department of Medical Research, China Medical University Hospital, China Medical University[4]
No.91, Hsueh-Shih Road, Taichung 40402, Taiwan
(Email: mshwang@asia.edu.tw)
(Received May 12, 2016; revised and accepted July 21 & Aug. 4, 2016)

Abstract

In this article, we propose a publicly verifiable authenticated encryption scheme based on factoring and discrete logarithms. We point out that even if either factoring or discrete logarithms is broken, this scheme still could keep the authentication, integration, and confidentiality of the message.

Keywords: Authenticated encryption scheme, discrete logarithm problem, factoring problem

1 Introduction

Traditionally, we use a hand-written signature to manifest the validity of a document and the identity of a signer. In the wake of development network, the information could be easily transmitted through the network in the form of electronic types. According to the requirements of secure data, senders must provide a secure protocol in the environment. The signer could use his/her secret key to generate a signature for the given message, and the verifier then uses the signer's public key to verify the authentication function of a paper [10, 26, 31]. Most previously proposed digital signature schemes were based on the well-known public key systems such as RSA [1, 5, 12, 32, 34] system or ElGamal system [7, 9, 21, 30, 38]. According to those public key systems, the sender not only encrypts the documents which are taken in electronic form but protects the content security of the documents as well.

Although the original digital signature could verify the creator identity of a document, it is not enough in some special applications [20, 29, 36]. When a sender wants to securely transmit data to a particular receiver, he/she must ensure that nobody but the particular receiver could authenticate the signature [2, 25, 28]. For example, in the movie "mission impossible", when the government assigns a task to Tom Cruise, only he could know the message of the task. Since the third party is unable to know the content of the message, the third party could not authenticate the task sent from government. According to the requirements of the special application, we could get the criterion of the authenticated encryption scheme, so signing signature and protecting the security of a document could be made at the same time [16, 18]. We can infer that an authenticated encryption scheme corresponds with the following properties [3, 17, 37]:

Confidentiality. It must ensure that the secret information can only be obtained, by the sender and the receiver, but not anyone else.

Authentication. It must ensure the sender and the receivers' identities, and avoid the adversary to send a malicious message.

Non-repudiation. It must confirm the sender's identity, and the sender could not repudiate his signature and message.

All of the above are the basic requirements of the authenticated encryption scheme (AES for short). If the proposed scheme satisfies those characteristics, it will be called an authenticated encryption scheme.

Nyberg and Rueppel [27] proposed a signature with message recovery based on the discrete logarithm problem. In this scheme, there are some advantages with which the application without a hash function is possible to be achieved, such as a smaller bandwith for signatures of small messages, and direct use in other schemes like identity-based public key systems or key agreement protocols. Recently, Horster, Michels, and Petersen [14] (HMPs for short) proposed an authenticated encryption scheme based on a message recovery method which is the modification of Nyberg-Rupple's scheme [27]. In their scheme, a sender does not have to transmit a message to the receiver. Then, the receiver not only could verify the message authentication and the message integrity, but he/she could also get the original message from the information that he/she has received. Although HMPs provided the confidentiality, this scheme was not secure in use because it suffered from "known ciphertext-plaintext attack". Li and Chang proposed an improved scheme [24]. Then Wu and Hsu [15] pointed out Li and Chang's scheme [24] was not perfect when a dispute occurred, so they proposed a scheme to make up for the disadvantage. Next, Ma and Chen [4] proposed a new application in AES. Their scheme could provide the third party to verify the signature without knowing a plaintext, except the sender and the receiver. Many schemes have been proposed to achieve the properties of authenticated encryption schemes [8, 22, 35].

We have described that if someone wants to sign or encrypt a message, he/she will often use public key systems. The security of RSA system is constructed in factorization, and ElGamal system is built on solving the discrete logarithm problem. In general, to verify a signature or to decrypt an encrypted message which is based on factoring or discrete logarithms is not easy to achieve. However, it is optimistic to improve the computation capability of a computer. Nobody could ensure the perpetual security of a cryptography algorithm which is based on either the factoring or the discrete logarithm problem in the future. Therefore, He first proposed a signature scheme which is based on both factoring and discrete logarithm problems. The proposed scheme could improve the degree of the security and ensure the feasibility of the algorithm.

In 1976, Diffie-Hellman [6] presented a concept of public-key cryptography. Since then, the security of each public-key cryptosystem has been based on just one cryptographic assumption, either factoring or discrete logarithms. However, it is possible that efficient algorithms will be developed in the future to break one or more of these assumptions.

Harn [11] first proposed a new public-key cryptosystem based on factoring and discrete logarithms. Unfortunately, Shao [33] showed that Harn's scheme would be subject to substitution attacks without using any hash function. Then Shao proposed an improved scheme to resist such a substitution attack on the condition that the factoring problem would not be broken. However, Lee [23] showed that Shaos' [33] improved scheme is still insecure. When the factoring problem is broken, the adversary will obtain signer's secret key from a known signature. He [13] pointed out a common disadvantage of those schemes that have been proposed. Every user has his or her key pair and arithmetic module, so there exists the key management problem for those schemes. Then his scheme aimed at this disadvantage. Lastly, Hwang et al. [19] proposed an improved scheme for his scheme.

We base on the proposed scheme of Hwang et al., and propose a new authenticated encryption scheme based on FAC and DL. In this article, we present a scheme that could satisfy properties of authenticated encryption schemes based on factoring and discrete logarithms. In addition, the third party could verify the signature without divulging the receiver's private key.

Thus, the proposed authenticated encryption schemes are either based on factoring difficulty or discrete logarithm difficulty. According to the essence of He's scheme, we propose a new AES based on FAC and DL, which correspond to the characteristics of the traditional AES based on FAC or DL, and is securer than others. In this article, the third party could verify the signature without divulging the receiver's private key. In Section one, a brief development of AES and signature based on FA and DL is discussed. In Section two, we describe the main scheme. Then we point out in Section three some possible attacks and show that this scheme could avoid these attacks. Finally, we make a conclusion for this article.

2 The Proposed Scheme

In this section, we propose a new scheme could not only achieve authentication and encryption functions, but could increase security by solving two difficulties of factoring and discrete logarithms as well. This scheme could be divided into three phases. In the initialization phase, related system parameters should be defined. In the encryption and signature generation phase, the signer could create a signature with message recovery to a special recipient. In signature verification and message recovery phase, the special recipient will verify the correctness and integrity of the message, and recover the message. To prevent certain dispute later, the designated recipient may convert the encryption into an ordinary signature.

Initialization Phase.

In this phase, the trusted center of the system selects the following parameters:

- p_1, p_2, q_1, q_2: Four large primes where $p_1 = 2p_2 + 1$ and $q_1 = 2q_2 + 1$;
- P: A large prime where $P = 4p_1q_1 + 1$;
- R: $R = p_1q_1$;
- g: A generator of order p_1p_2 over $GF(P)$.

P, R, and g are published, and p_1, p_2, q_1, q_2 are all discarded. Then each user selects his/her private

key X_i in Z_R where $gcd(X_i^2, R) = 1$, and computes his/her public key y_i where $y_i = g^{X_i^2} \bmod P$.

Encryption and Signature Generation Phase.

If a sender wants to transmit a secure message, he/her should perform the following protocol to generate a ciphertext and a signature for a message m.

Step 1. He/she should randomly select an integer K in Z_R such that $\gcd(K^2, R) = 1$, and compute

$$\begin{aligned} r_1 &= g^{K^2} \bmod P \\ r_2 &= g^{K^{-2}} \bmod P. \end{aligned}$$

Step 2. And he/she could encrypt the message m to find a ciphertext C such that

$$C = mH(y_B^{K^2} \bmod P)^{-1} \bmod P,$$

where H is a one-way function.

Step 3. He/she signs the message m in order to convince the verifier of the validity of the original message. And then he/she performs the following equations:

$$X_A = SK + H(r_1, r_2, m)K^{-1} \bmod R.$$

The pairs (r_1, r_2, S) is a signature of the message m. Then the sender delivers (r_1, r_2, S, C) to the receiver.

Signature Verification and Message Recovery Phase.

When the receiver obtains those four messages, he could perform the following equation to recover the message m from the ciphertext C, $m' = CH(r_1^{X_B^2} \bmod P) \bmod P$. The receiver will check the format of the message m', and then he could decide whether to accept it or not. After recovering the message, he must check the validity and the correctness of the signature. He could use the following equation to verify the signature:

$$y_A = r_1^{S^2} r_2^{H^2(r_1, r_2, m)} g^{2SH(r_1, r_2, m)} \bmod P.$$

If the signature are valid, then the receiver could make the conclusion that the information is correct.

In case of dispute, the special recipient must reveal the content of the message m with (r_1, r_2, S, m) and send it to the third party. When the third party receives those information, he/she will verify the legality and the correctness with the following equation:

$$y_A = r_1^{S^2} r_2^{H^2(r_1, r_2, m)} g^{2SH(r_1, r_2, m)} \bmod P.$$

If the equation equals, the sender is not able to disclaim that he has sent the message m to the special recipient.

All the explanation above introduces our scheme that is based on factoring and discrete logarithms. We can easily find that our scheme is more secure and could achieve the purpose of authenticated encryption.

3 Security Analysis and Performance Analysis

In this section, we will present some securities of the proposed scheme. Then we will present the performance analysis of our proposed scheme.

3.1 Security Analysis

The following attack analysis methods are all general attack assumptions, and we will present other possible problems that may occur in the future. First, we assume that the well-known public key system based on discrete logarithm problems has been broken and shows whether our proposed scheme could resist those crunches or not. Next, we will assume that the cryptosystem security is based on factoring problems, and prove that our scheme could still resist the crisis.

Attack 1. The receiver wants to forge the sender's signature (r_1, r_2, S, C).

If a receiver wants to forge the sender's identity to sign a message m' which is chosen by him, he must know the sender's private key X_A. He could randomly chose a number K', and compute the ciphertext C' and the signature (r_1', r_2, S') of the message m'. However, he could not find the signature S', because he does not know the sender's private key X_A. And thus, he could not forge the sender's identity to generate a signature.

Attack 2. An intruder tries to obtain the private key X_i from a user $U_i's$ signature.

In our scheme, if the intruder wants to obtain user A's secret key X_i from the user's signature, he must solve the equation $X_A = SK + H(r_1, r_2, m)K^{-1} \bmod R$, even though he could know the public hash functions $H(.), R$, and the four values (r_1, r_2, S, C) which is transmitted through the network. He should know the secret random number K which is the same as each signature process and message m.

First, the intruder should find a message m' and compute the hash value which is equal to the original hash value such as $H(r_1, r_2, m') = H(r_1, r_2, m')$. Nerveless, he will face the difficulty of finding a hash value; a hash function is computationally infeasible to find any second input which has the same output as any specified input, i.e., given x, to find a $x' \neq x$ such that $h(x) \equiv h(x')$. At the same time, he could infer that the random number K' is same as K, and he could only obtain the sender's private key X_i.

Attack 3. An opponent tries to impersonate the signer to generate a valid signature of a message m'.

If an opponent wants to forge a valid signer's signature for a random message m', he could fix r_1, r_2 and computes a fake S' to pass the verification process. If he wants to find a legal S', he will confront

discrete logarithm problems. Because he must know the correct value K, he should compute the K from the $r_1 = g^{K^2} \bmod P$. Second, he could fix S which is obtained from the transmitted process and finds the corresponded r_1, and r_2. However, he must know the correct message m. Otherwise, he could find a hash value $H(r_1, r_2, m')$ which is same as $H(r_1, r_2, m)$.

Attack 4. An adversary without the special receiver B's private key X_B tries to decrypt the ciphertext C.

If an adversary wants to know the message m, he should get the receiver's private key X_B to decrypt the ciphertext C. Also, he could solve the discrete logarithm problems, and obtain the value K^2. However, the difficulty is not easy to be solved. The reason is just as those described above.

Attack 5. Suppose that either the difficulty of computing discrete logarithm problem or the factoring problem has been broken.

Given the ciphertext C, the adversary attempts to find the solution of three variables r_1, r_2, and S that satisfies the equation $y_A = r_1^{S^2} r_2^{H^2(r_1,r_2,m)} g^{2SH(r_1,r_2,m)} \bmod P$. He first fixes two variables and finds the solution of the other variable from the verification equation. Besides, given y_A, g, C, r_1, and r_2, finding S to satisfy verification equation is under the factoring and discrete logarithm assumptions. In another similar approach, those are under the factoring, the discrete logarithm, and hash function assumptions.

3.2 Performance Analysis

In this subsection, we will focus on the performance of our scheme and to analyze the efficiency. For convenience, we first define some notations to denote the performance time: T_{mul} is the time for multiplication; T_h is the time for executing hash function; T_{exp} is the time for exponentiation with modulo P; and T_{inv} is the time for inversion modulo P. We only consider those heavy computational cost of T_h, T_{exp}, T_{mul}, and T_{inv}.

And then we could dispute the computational cost over two phases, signature and ciphertext generation phase, and message recovery and verification phase. In the signature and ciphertext generation phase, the sender will perform $3T_{exp}, 2T_{inv}, 2T_h, 2T_{mul}$ to achieve the processes of this phase. In the message recovery and verification phase, the verifier should perform $4T_{exp}, 1T_h, 4T_{mul}$ to complete the processes of this phase. In the dispute phase, we will not take into consideration the computational cost.

4 Conclusions

Our proposed scheme is based on Nyberg and Rueppel's scheme [27] and Hwang et al. [19] proposed scheme. Nyberg and Rueppel's method could be applied to small message transmissions such as ID-based systems or key agreement systems. The other one, Hwang et al.'s scheme, could provide good security based on factoring and discrete logarithm assumptions. We extend both schemes to construct our algorithm. The proposed scheme is a publicly verifiable authenticated encryption scheme based on factoring and discrete logarithms. This scheme could ensure the security and authentication of the message by solving two difficult problems.

In the future, some subjects, especially the mobile environment, are worth considering in applications. In a mobile environment, short response time and efficient computation are very important. When a user requests a service to a provider with a payment way, he will considerably care about the transmitted time and cost. Since it costs quite much for the computation of authentication encryption schemes, more efforts should be made to improve the efficiency.

References

[1] C. C. Chang and M. S. Hwang, "Parallel computation of the generating keys for RSA cryptosystems," *IEE Electronics Letters*, vol. 32, no. 15, pp. 1365–1366, 1996.

[2] C. C. Chang, C. Y. Sun, and S. C. Chang, "A strong RSA-based and certificateless-based signature scheme," *International Journal of Network Security*, vol. 18, no. 2, pp. 201–208, 2016.

[3] T. Y. Chang, C. C. Yang, and M. S. Hwang, "Cryptanalysis of publicly verifiable authenticated encryption," *IEICE Transactions on Foundations*, vol. E87-A, no. 6, pp. 1645–1646, 2004.

[4] M. Changshe and C. Kefei, "Publicly verifiable authenticated encryption," *Electronics Letters*, vol. 39, no. 3, pp. 281–282, 2003.

[5] L. Deng, H. Huang, and Y. Qu, "Identity based proxy signature from RSA without pairings," *International Journal of Network Security*, vol. 19, no. 2, pp. 229–235, 2017.

[6] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[7] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 220–222, July 1985.

[8] L. H. Encinas, A. M. del Rey, and J. M. Masqué, "A weakness in authenticated encryption schemes based on tseng et al.'s schemes," *International Journal of Network Security*, vol. 7, no. 2, pp. 157–159, 2008.

[9] O. B. Fredj, "An automatic alert unification method for heterogeneous alert signatures," *International Journal of Network Security*, vol. 18, no. 6, pp. 1180–1191, 2016.

[10] Y. Gao, P. Zeng, K. K. R. Choo, F. Song, "An improved online/offline identity-based signature scheme

for WSNs," *International Journal of Network Security*, vol. 18, no. 6, pp. 1143–1151, 2016.

[11] L. Harn, "Public-key cryptosystem design based on factoring and discrete logarithms," *IEE Proceedings - Computers and Digital Techniques*, vol. 141, no. 3, pp. 193–195, 1994.

[12] D. He, M. K. Khan, and S. Wu, "On the security of a RSA-based certificateless signature scheme," *International Journal of Network Security*, vol. 16, no. 1, pp. 78–80, 2014.

[13] W. H. He, "Digital signature scheme based on factoring and discrete logarithms," *Electronics Letters*, vol. 37, no. 4, pp. 220–222, 2001.

[14] P. Horster, M. Michels, and H. Petersen, "Authenticated encryption schemes with low communication costs," *Electronics Letters*, vol. 30, no. 15, pp. 1212–1213, 1994.

[15] C. L. Hsu and T. C. Wu, "Authenticated encryption scheme with *(t, n)* shared verification," *IEE Proceedings - Computers and Digital Techniques*, vol. 145, no. 2, pp. 117–120, 1998.

[16] H. F. Huang, P. H. Lin, and M. H. Tsai, "Convertible Multi-authenticated Encryption Scheme for Data Communication," *International Journal of Network Security*, vol. 17, no. 1, pp. 40–48, 2015.

[17] M. S. Hwang and C. Y. Liu, "Authenticated encryption schemes: Current status and key issues," *International Journal of Network Security*, vol. 1, no. 2, pp. 61–73, 2005.

[18] M. S. Hwang, J. W. Lo, and S. Y. Hsiao, "Improvement of authenticated encryption schemes with message linkages for message flows," *IEICE Transactions on Information and Systems*, vol. E89-D, no. 4, pp. 1575–1577, 2006.

[19] M. S. Hwang, C. C. Yang, and S. F. Tzeng, "Improved digital signature scheme based on factoring and discrete logarithms," *Discrete Mathematical Sciences & Cryptography*, vol. 5, no. 2, pp. 151–155, 2002.

[20] J. Kar, "Provably secure online/off-line identity-based signature scheme for wireless sensor network," *International Journal of Network Security*, vol. 16, no. 1, pp. 29–39, 2014.

[21] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[22] C. C. Lee, M. S. Hwang, and S. F. Tzeng, "A new convertible authenticated encryption scheme based on the elgamal cryptosystem," *International Journal of Foundations of Computer Science*, vol. 20, no. 2, pp. 351–359, 2009.

[23] N. Y. Lee, "Security of shao's signature schemes based on factoring and discrete logarithms," *IEE Proceedings - Computers and Digital Techniques*, vol. 146, no. 2, pp. 119–121, 1999.

[24] W. B. Lee and C. C. Chang, "Authenticated encryption scheme without using a one way function," *Electronics Letters*, vol. 31, no. 19, pp. 1656–1657, 1995.

[25] C. Y. Liu, C. C. Lee, and T. C. Lin, "Cryptanalysis of an efficient deniable authentication protocol based on generalized elgamal signature scheme," *International Journal of Network Security*, vol. 12, no. 1, pp. 58–60, 2011.

[26] D. Liu, S. Zhang, H. Zhong, R. Shi, and Y. Wang, "An efficient ID-based online/offline signature scheme without key escrow ," *International Journal of Network Security*, vol. 19, no. 1, pp. 127–137, 2017.

[27] K. Nyberg and R. A.Rueppel, "A new signature scheme based on the dsa giving message recovery," In *ACM Computer & Communications Security*, vol. 1, pp. 58–61, 1993.

[28] N. Ojha and S. Padhye, "Weak keys in rsa over the work of blomer & may," *International Journal of Network Security*, vol. 14, no. 2, pp. 80–85, 2012.

[29] C. Pan, S. Li, Q. Zhu, C. Wang, and M. Zhang, "Notes on Proxy Signcryption and Multi-proxy Signature Schemes," *International Journal of Network Security*, vol. 17, no. 1, pp. 29–33, 2015.

[30] S. Qadir, M. U. Siddiqi, W. F. M. Al-Khateeb, "An investigation of the Merkle signature scheme for cryptographically generated address signatures in mobile IPv6," *International Journal of Network Security*, vol. 17, no. 3, pp. 311–321, 2015.

[31] Y. Ren, S. Wang, X. Zhang, M. S. Hwang, "An efficient batch verifying scheme for detecting illegal signatures," *International Journal of Network Security*, vol. 17, no. 4, pp. 463–470, 2015.

[32] K. R. Santosh, C. Narasimham, and P. Shetty, "Cryptanalysis of multi-prime RSA with two decryption exponents," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 40–44, 2016.

[33] Z. Shao, "Signature schemes based on factoring and discrete logarithms," *IEE Proceedings - Computers and Digital Techniques*, vol. 145, no. 1, pp. 33–36, 1998.

[34] G. Sharma, S. Bala, A. K. Verma, "An improved RSA-based certificateless signature scheme for wireless sensor networks," *International Journal of Network Security*, vol. 18, no. 1, pp. 82–89, 2016.

[35] S. F. Tzeng, Y. L. Tang, and M. S. Hwang, "A new convertible authenticated encryption scheme with message linkages," *Computers and Electrical Engineering*, vol. 33, no. 2, pp. 133–138, 2007.

[36] F. Wang, C. C. Chang, C. Lin, S. C. Chang, "Secure and Efficient Identity-based Proxy Multi-signature Using Cubic Residues," *International Journal of Network Security*, vol. 18, no. 1, pp. 90–98, 2016.

[37] Z. Yang, C. Liu, W. Liu, S. Luo, H. Long and S. Li, "A lightweight generic compiler for authenticated key exchange from non-interactive key exchange with auxiliary input," *International Journal of Network Security*, vol. 18, no. 6, pp. 1109–1121, 2016.

[38] Y. Zhang, H. Li, X. Li, and H. Zhu, "Subliminal-free Variant of Schnorr Signature with Provable Security," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 19–30, 2015.

Biography

Cheng-Yi Tsai received the B.S. degree in Department of Business Administration from Chaoyang University of Technology (CYUT), Taichung, Taiwan, in 2001; the M.S. degree in Computer Science & Information Engineering from Asia University, Taichung, Taiwan, in 2005. He is currently pursuing his PHD degree in Graduate Institute of Computer Science & Information Engineering from Asia University. His current research interests include applied cryptography and mobile communications.

Chi-Yu Liu received the B.S. degree in Information Management from Chaoyang University of Technology (CYUT), Taichung, Taiwan, Republic of China, in 2003. She is currently pursuing her M.S. degree in Graduate Institute of Networking and Communication Engineering from CYUT. Her current research interests include applied cryptography and mobile communications.

Shyh-Chang Tsaur received the B.S. in Electronic Engineering from National Taiwan University, Taiwan, in 1967; the M.A. in Physics from State University of New York at Stony Brook, USA, in1969; the Ph.D. in Electronic Engineering from Carnegie Mellon University, USA, in 1973. Dr. Tsaur with Dr. C Kuo jointly have received more than 10 US patents in Semiconductor Memories during his work in Texas Instruments, USA from 1973 to 1981. From 1981 to 1996, Dr. Tsaur has been in computer industries for 15 years including owning a PC store, employed as CIO in CMS, CA, USA, Information consultants, etc. Since 1996, Dr Tsaur has been employed as the Special Assistant to HCG Chairman for 5 years successfully to reengineer MIS department; hired as an information consultant of TSANN KUEN 3C Group to accomplish a real time EIS system of 150 chain stores in one year; a professor in CSIE department of Asia University until he retired. In last ten years, in addition to teaching in Universities, Dr. Tsaur has co-authored 3 books: RFID principle, Application and Implementation; Database System Theory and Applications; Cloud Computing Introduction: Entering APP Software World.

Min-Shiang Hwang Min-Shiang Hwang received the B.S. in Electronic Engineering from the National Taipei Institute of Technology, Taipei, Taiwan, in 1980; the M.S. in Industrial Engineering from the National Tsing Hua University, Taiwan, in 1988; and a Ph.D. in Computer and Information Science from the National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at the National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications. He was also the chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2002. He was a professor and chairman of the Graduate Institute of Networking and Communications, CYUT, during 2002-2003. He was a professor and chairman of the Department of Management Information Systems, National Chung Hsing University (NCHU), during 2003-2009. He was also a visiting professor of UC. Riverside and UC. Davis (USA) during 2009-2010. He was a distinguished professor of the department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Excellent Research Award of National Science Council (Taiwan). He was a dean of College of Computer Science, Asia University (AU). He is currently a Chair Professor of the department of Computer Science & Information Engineering, AU. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include information security, electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 200+ articles on the above research fields in international journals.

Verifiable Outsourcing Computation of Modular Exponentiations with Single Server

Jianxing Cai, Yanli Ren, and Chunshui Huang
(Corresponding author: Yanli Ren)

School of Communication and Information Engineering, Shanghai University
Shanghai 200444, China
(Email: renyanli@shu.edu.cn)

(Received Dec. 5, 2015; revised and accepted Jan. 23 & Feb. 5, 2016)

Abstract

Verifiable computation (VC) allows a computationally weak client to outsource evaluation of a function on many inputs to a powerful but untrusted server. In this paper, we propose an algorithm of verifiable outsourcing computation with single server on modular exponentiation, which has wide applications in public key cryptosystems. We also extend the algorithm to verifiable outsourcing of simultaneous modular exponentiations. The proposed two algorithms improve checkability based on one server compare with the previous ones, where the outsourcer can detect the failure with probability close to 1 if the server misbehaves. The experiments show that our algorithms are the implementations of secure and verifiable outsourcing for single modular exponentiation and simultaneous modular exponentiations.

Keywords: Checkability, modular exponentiation, single server, verifiable outsourcing computation

1 Introduction

Outsourcing computation allows a computation-limited client to outsource the operations on the private data to a powerful server [5]. The rise of cloud computing in recent years has made outsourcing of storage and computation reality [20, 21]. Outsourcing computation will provide an ideal way of freeing up the resources of the client as one of the advantages of cloud computing model [18, 22]. By outsourcing work load to the cloud server, cloud users can use unlimited resources provided by cloud to complete the high cost of computing.

Despite of the tremendous benefits, outsourcing computation also brings an unprecedented security challenge [4, 12]. First of all, due to the opaque of cloud server's internal operational details, there are various malicious motives. It makes the server cannot be completely trusted. For instance, cloud server may be Jerry-rigged in a computation task in order to save resources if clients cannot judge the correctness of the output. Secondly, there may be malicious attacks from external or internal of server such as some software bugs. All of these can make the server return computationally invalid results [5]. Therefore private security and content security of the cloud environment are important problems in our research.

In general, an effective secure outsourcing scheme should have the following properties:

1) The client's data is private for the server;

2) The client can verify the correctness of the output returned by the server.

3) The client can carry out the computation correctly using substantially less effort than computing the result on its own [5, 14, 17].

Dijk et al. [7] proposed outsourcing algorithms of variable-exponent fixed-base and fixed-exponent variable-base exponentiations in a model with one untrusted server. In these algorithms, no variable parts are public before sending to the only untrusted server. They also considered an algorithm of outsourcing variable-exponent variable-base exponentiations. However, in this algorithm, the outsourced base is known to the server. Ma, Li and Zhang [13] described securely outsourcing algorithms of two types of exponentiations in two non-collusion untrusted servers. Hohenberger and Lysyanskaya [11] presented outsource-secure algorithms of variable-exponent variable-base exponentiations in one-malicious model of two untrusted servers. Chen et al. [5] also proposed outsourcing algorithm of modular exponentiation based on two servers, and improved the checkability and efficiency for the outsourcer. Wang et al. [19] constructed an efficient algorithm for batch modular exponentiation based on an untrusted server, but the outsourcer need to execute one modular exponentiation itself when verifying the outsourced result and the checkability is only $1/(n+1)$, where n is the number of modular exponentiations.

In this paper, we first propose a secure verifiable outsourcing algorithm of single modular exponentiation with single server. We also present another outsourcing algorithm for simultaneous modular exponentiations. In the proposed algorithms, the outsourcer could detect any failure with probability close to 1 if the server returns the fault result. The experiments show that the proposed algorithms improve checkability without decreasing efficiency for the outsourcer compare with the previous ones.

The organization of this paper is described as follows: in Section 2, the definitions and security requirements of our algorithms are given. A verify outsourcing algorithm of modular exponentiation is proposed in Section 3. In the following section, we present another outsourcing algorithm for simultaneous modular exponentiations by using single server. The performance evaluation of the proposed algorithms is given in Section 5. We conclude the paper in Section 6.

2 Definitions and Security Requirements

In this section, we review definitions and security requirements of outsource-secure algorithms which have been used in [5, 11, 19].

An algorithm **Alg** includes a trusted part T and an untrusted program U which is invoked by T. We use T^{U} to denote the work that executed by T and U. An adversary A is simulated by a pair of algorithms A=(E,U'), where E denotes the adversarial environment and generates adversarial inputs for **Alg**, and U' represents an adversarial software written by E. The security model is shown in Figure 1.

Definition 1. (Algorithm with outsource-I/O) *An outsourcing algorithm **Alg** takes five inputs and generates three outputs. The first three inputs are generated by an honesty party, and the last two inputs are produced by an adversary environment E. The first input is called the honest, secret input, which is private for both E and U'; the second input is honest and protected, which may be known by E, but is private for U'; the third one is called honest and unprotected input, which is public for both E and U. The last two inputs are maliciously chosen by E, and thus they are known for E. One of them is adversarial and protected, which is protected from U'; the other one is public for E, and we call it the adversarial, unprotected input. Similarly, the first output is secret and unknown for both E and U'; the second one is protected, which means it may be public for E, but protected from U'; the last output is unprotected, which are public for both E and U'.*

As presented in [11], we assume that the two adversaries E, U' can only make direct communication before the execution of T^{U}, and in other cases, they are not be able to communicate directly and must pass message through the outsourcer T. The reason is explained as follows. In the real world, we assume E is a malicious manufacture, and U' is a malicious software produced by E. It is obvious that U' is controlled by E and they can exchange the messages directly before it is sold to T. However, E cannot send instructions to U' directly once T begins to invoke U' as the outsourcing program. During the execution of T^{U}, E can only establish an indirect communication channel with U' through the unprotected inputs and outputs of **Alg**, which means that all messages they communicate with each other must pass through T.

The following outsource-security definition ensures that both E and U' cannot learn nothing about the private inputs and outputs of T^{U}, even if T uses the malicious software U' written by E.

Definition 2. (Outsource-security) *Let **Alg** be an algorithm as defined above. A pair of algorithms (T, U) is called to be outsource-security if the following conditions holds.*

1) Correctness: T^{U} is a correct execution of **Alg**.
2) Security: For any probabilistic polynomial-time (PPT) adversary A=(E,U'), there exist two PPT simulators $(\mathrm{S}_1, \mathrm{S}_2)$ such that the results of real and ideal experiment are computationally indistinguishable.

Pair One: $EView_{\mathrm{real}} \sim EView_{\mathrm{ideal}}$, which means that the malicious environment E cannot learn anything interesting about the secret inputs and outputs during the execution of T^{U}. Both of the real process and the ideal process proceed in rounds. The notation "$\leftarrow$" denotes the outputs of the procedure in the right hand side.

- The i-th round of real process consists of the following steps. The three honest inputs $(\mathrm{x_{hs}}, \mathrm{x_{hp}}, \mathrm{x_{hu}})$ are chosen by an honest stateful process I where the environment E cannot access. Then E chooses $\mathrm{estate^i}$, $\mathrm{x^i_{ap}}$, $\mathrm{x^i_{au}}$, $stop^{\mathrm{i}}$ based on its view from the last round and honest inputs $(\mathrm{x^i_{hs}}, \mathrm{x^i_{hp}}, \mathrm{x^i_{hu}})$ given to T^{U}, where $estate^{\mathrm{i}}$ is a random number as reminding what it did next time it is invoked, $\mathrm{x^i_{ap}}$, $\mathrm{x^i_{au}}$ are two adversarial inputs, and $stop^{\mathrm{i}}$ is the Boolean variable which denotes whether round i is the last round. Next, the algorithm $\mathrm{T}^{\mathrm{U}'}$ is implemented on the inputs $(\mathrm{tstate^{i-1}}, \mathrm{x^{j^i}_{hs}}, \mathrm{x^{j^i}_{hp}}, \mathrm{x^{j^i}_{hu}}, \mathrm{x^i_{ap}}, \mathrm{x^i_{au}})$, and generates a new state $tstate^{\mathrm{i}}$ for T, and secret, protected, unprotected outputs $\mathrm{y^i_s}$, $\mathrm{y^i_p}$, $\mathrm{y^i_u}$, where $\mathrm{tstate^{i-1}}$ is T's previously saved state. The oracle U' saves its current state $ustate^{\mathrm{i}}$ based on its previously saved state $ustate^{\mathrm{i-1}}$.
  - $(\mathrm{istate^i}, \mathrm{x^i_{hs}}, \mathrm{x^i_{hp}}, \mathrm{x^i_{hu}}) \leftarrow \mathrm{I}(1^{\mathrm{k}}, \mathrm{istate^{i-1}})$;
  - $(\mathrm{estate^i}, \mathrm{j^i}, \mathrm{x^i_{ap}}, \mathrm{x^i_{au}}, \mathrm{stop^i}) \leftarrow E(1^k, EView^{i-1}_{real}, x^i_{hp}, x^i_{hu})$;
  - $(\mathrm{tstate^i}, \mathrm{ustate^i}, \mathrm{y^i_s}, \mathrm{y^i_p}, \mathrm{y^i_u}) \leftarrow T^{U' ustate^{i-1}}(tstate^{i-1}, x^{j^i}_{hs}, x^{j^i}_{hp}, x^{j^i}_{hu}, x^i_{ap}, x^i_{au})$.

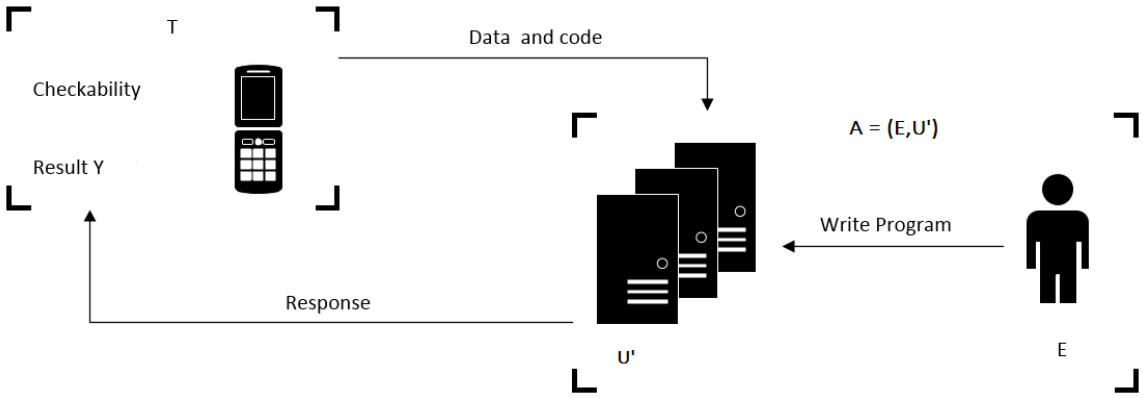


Figure 1: Security model

Thus, the view that the adversarial environment E gets in the i-th round of the real process is $\text{EView}^{\text{i}}_{\text{real}} = (\text{estate}^{\text{i}}, \text{y}^{\text{i}}_{\text{p}}, \text{y}^{\text{i}}_{\text{u}})$ and the overall view is the view in the last round, i.e., $\text{EView}_{\text{real}} = \text{EView}^{\text{i}}_{\text{real}}$ if $\text{stop}^{\text{i}} = \text{TRUE}$.

- The i-th round of ideal process consists of the following steps. In i-th round, the stateful simulator S_1 knows nothing about the secret input $\text{x}^{\text{i}}_{\text{hs}}$, but is given the protected and unprotected outputs that generated by **Alg**. Finally, S_1 outputs some values, either $(\text{y}^{\text{i}}_{\text{p}}, \text{y}^{\text{i}}_{\text{u}})$ or $(\text{Y}^{\text{i}}_{\text{p}}, \text{Y}^{\text{i}}_{\text{u}})$ captured by using a Boolean indicator ind^{i}. In the whole process, S_1 is allowed to access oracle U' and U' saves its state as in the real experiment.
  - $(\text{istate}^{\text{i}}, \text{x}^{\text{i}}_{\text{hs}}, \text{x}^{\text{i}}_{\text{hp}}, \text{x}^{\text{i}}_{\text{hu}}) \leftarrow \text{I}(1^{\text{k}}, \text{istate}^{\text{i}-1})$.
  - $(\text{estate}^{\text{i}}, \text{j}^{\text{i}}, \text{x}^{\text{i}}_{\text{ap}}, \text{x}^{\text{i}}_{\text{au}}, \text{stop}^{\text{i}}) \leftarrow \text{E}(1^{\text{k}}, \text{EView}^{i-1}_{\text{ideal}}, \text{x}^{\text{i}}_{\text{hp}}, \text{x}^{\text{i}}_{\text{hu}})$.
  - $(\text{astate}^{\text{i}}, \text{y}^{\text{i}}_{\text{s}}, \text{y}^{\text{i}}_{\text{p}}, \text{y}^{\text{i}}_{\text{u}}) \leftarrow \text{Alg}(astate^{i-1}, x^{j^{\text{i}}}_{\text{hs}}, x^{j^{\text{i}}}_{\text{hp}}, x^{j^{\text{i}}}_{\text{hu}}, \text{x}^{\text{i}}_{\text{ap}}, \text{x}^{\text{i}}_{\text{au}})$.
  - $(\text{sstate}^{\text{i}}, \text{ustate}^{\text{i}}, \text{Y}^{\text{i}}_{\text{p}}, \text{Y}^{\text{i}}_{\text{u}}, \text{ind}^{\text{i}}) \leftarrow \text{S}_1^{\text{U}'(ustate^{\text{i}-1})}(sstate^{\text{i}-1}, \text{x}^{\text{j}^{\text{i}}}_{\text{hp}}, x^{j^{\text{i}}}_{\text{hu}}, \text{x}^{\text{i}}_{\text{ap}}, \text{x}^{\text{i}}_{\text{au}}, y^{i}_{p}, y^{i}_{u})$.
  - $(\text{z}^{\text{i}}_{\text{p}}, \text{z}^{\text{i}}_{\text{u}}) = \text{ind}^{\text{i}}(\text{Y}^{\text{i}}_{\text{p}}, \text{Y}^{\text{i}}_{\text{u}}) + (1 - \text{ind}^{\text{i}})(\text{y}^{\text{i}}_{\text{p}}, \text{y}^{\text{i}}_{\text{u}})$.

  Thus, the view that E obtains in the i-th round of the real process is $\text{EView}^{\text{i}}_{\text{ideal}} = (\text{estate}^{\text{i}}, \text{z}^{\text{i}}_{\text{p}}, \text{z}^{\text{i}}_{\text{u}})$ and the overall view is the view in the last round, i.e., $\text{EView}_{\text{ideal}} = \text{EView}^{\text{i}}_{\text{ideal}}$ if $\text{stop}^{\text{i}} = \text{TRUE}$.

Pair Two: $\text{UView}_{\text{real}} \sim \text{UView}_{\text{ideal}}$, which means that the untrusted software U' written by E cannot get anything interesting about the inputs and outputs during the execution of T^{U}.

- On the basis of the real process described in Pair One, the view of the untrusted software U' in the real process is $\text{UView}_{\text{real}} = \text{ustate}^{\text{i}}$ if $\text{stop}^{\text{i}} = \text{TRUE}$.
- The i-th round of ideal process consists of the following steps. In i-th round, the stateful simulator S_2 is given the unprotected outputs that generated by **Alg**.
  - $(\text{istate}^{\text{i}}, \text{x}^{\text{i}}_{\text{hs}}, \text{x}^{\text{i}}_{\text{hp}}, \text{x}^{\text{i}}_{\text{hu}}) \leftarrow \text{I}(1^{\text{k}}, \text{istate}^{\text{i}-1})$.
  - $(\text{estate}^{\text{i}}, \text{j}^{\text{i}}, \text{x}^{\text{i}}_{\text{ap}}, \text{x}^{\text{i}}_{\text{au}}, \text{stop}^{\text{i}}) \leftarrow \text{E}(1^{\text{k}}, \text{estate}^{\text{i}-1}, \text{x}^{\text{i}}_{\text{hp}}, \text{x}^{\text{i}}_{\text{hu}}, \text{y}^{\text{i}-1}_{\text{p}}, y^{i-1}_{u})$.
  - $(\text{astate}^{\text{i}}, \text{y}^{\text{i}}_{\text{s}}, \text{y}^{\text{i}}_{\text{p}}, \text{y}^{\text{i}}_{\text{u}}) \leftarrow \text{Alg}(astate^{i-1}, x^{j^{\text{i}}}_{\text{hs}}, x^{j^{\text{i}}}_{\text{hp}}, x^{j^{\text{i}}}_{\text{hu}}, \text{x}^{\text{i}}_{\text{ap}}, \text{x}^{\text{i}}_{\text{au}})$.
  - $(\text{sstate}^{\text{i}}, \text{ustate}^{\text{i}}) \leftarrow \text{S}_2^{\text{U}'(ustate^{\text{i}-1})}(sstate^{\text{i}-1}, x^{j^{\text{i}}}_{\text{hu}}, \text{x}^{\text{i}}_{\text{au}})$.

  Thus, the view of the untrusted software U' in i-th round of the ideal process is $\text{UView}^{\text{i}}_{\text{ideal}} = (\text{ustate}^{\text{i}})$ and the overall view is the view in the last round, which means $\text{UView}_{\text{ideal}} = \text{UView}^{\text{i}}_{\text{ideal}}$ if $\text{stop}^{\text{i}} = \text{TRUE}$.

Finally, we give the following definition if T^{U} is a correct implementation of **Alg**.

Definition 3. **(α-efficient, secure outsourcing)** *A pair of algorithms (T, U) is called α-efficient if the running time of T is less than an α-multiplicative factor of that of **Alg** for any inputs x.*

Definition 4. **(β-checkable, secure outsourcing)** *A pair of algorithms (T, U) is called β-checkable if T detects the error with probability no less than β when U' deviates from its advertised functionality during the execution of $T^{U'}(x)$ for any inputs x.*

Definition 5. **((α, β)-outsource-security)** *A pair of algorithms (T, U) is said to be an (α, β)-outsource-secure execution of **Alg** if it is both α-efficient and β-checkable.*

3 Outsourcing Algorithm of Modular Exponentiation with Single Server

Similar to [5], a subroutine named Rand is invoked in our algorithm. The output is a random pair of the form $(\mathrm{b}, \mathrm{g}^{\mathrm{b}} \bmod \mathrm{p})$, and the input is a prime p and a base $\mathrm{g} \in \mathrm{Z}_{\mathrm{p}}^{*}$, where $\mathrm{b} \in \mathrm{Z}_{\mathrm{q}}$. To implement this subroutine, we can use a trusted server to generate a table of random for T and T retrieves a new pair in the table when an invocation is needed. We call this table-lookup method. Another method is generating those random pairs by using EBPV generator [15, 19].

The inputs of single server exponentiation (SgExp) include a base u and a power a, where $\mathrm{a} \in \mathrm{Z}_{\mathrm{q}}^{*}$ and $\mathrm{u} \in \mathrm{Z}_{\mathrm{p}}^{*}$, and $\mathrm{u}^{\mathrm{q}} = 1 \bmod \mathrm{p}$. Both a and u are secret for the server U. The output of SgExp is $\mathrm{u}^{\mathrm{a}} \bmod \mathrm{p}$ where p and q are two large primes and $\mathrm{q} | \mathrm{p} - 1$.

3.1 Outsourcing Algorithm

Here we propose our algorithm SgExp for secure outsourcing of exponentiations. One important security requirement for SgExp is that an adversary can get any useful information about the inputs and outputs of SgExp. Similar to [5] and [11], $\mathrm{U}(\mathrm{x}, \mathrm{y}) \rightarrow \mathrm{y}^{\mathrm{x}}$ express that we invoke the server to compute one time. And (x, y) are the inputs while $\mathrm{y}^{\mathrm{x}} \bmod \mathrm{p}$ are the outputs.

1) First, T invokes the subroutine Rand four times to create four blinding pairs $(\alpha, \mathrm{g}^{\alpha})$, $(\beta, \mathrm{g}^{\beta})$, $(\varepsilon, \mathrm{g}^{\varepsilon})$, $(\theta, \mathrm{g}^{\theta})$. We denote $\mathrm{A} = \mathrm{g}^{\alpha} \bmod \mathrm{p}$, $\mathrm{B} = \mathrm{g}^{\beta} \bmod \mathrm{p}$, $\mathrm{C} = \mathrm{g}^{\varepsilon} \bmod \mathrm{p}$, $\mathrm{D} = \mathrm{g}^{\theta} \bmod \mathrm{p}$.

2) The first logical divisions are

$$\mathrm{u}^{\mathrm{a}} = (\mathrm{Aw})^{\mathrm{a}} = \mathrm{g}^{\mathrm{a}} \alpha \mathrm{w}^{\mathrm{a}} = \mathrm{g}^{\beta} \mathrm{g}^{\gamma} \mathrm{w}^{\mathrm{a}} \bmod \mathrm{p},$$

where $\mathrm{w} = \mathrm{u}/\mathrm{A} \bmod \mathrm{p}$ and $\gamma = (\mathrm{a}\alpha - \beta) \bmod \mathrm{q}$.

In order to verify the correct of the consequence, we do the second divisions by using another two blinding pairs $(\varepsilon, \mathrm{g}^{\varepsilon})$ and $(\theta, \mathrm{g}^{\theta})$ as follows:

$$\mathrm{u}^{\mathrm{a}} = (\mathrm{Cv})^{\mathrm{a}} = \mathrm{g}^{\mathrm{a}\varepsilon} \mathrm{v}^{\mathrm{a}} = \mathrm{g}^{\theta} \mathrm{g}^{\tau} \mathrm{v}^{\mathrm{a}} \bmod \mathrm{p},$$

where $\mathrm{v} = \mathrm{u}/\mathrm{C} \bmod \mathrm{p}$ and $\tau = (\mathrm{a}\varepsilon - \theta) \bmod \mathrm{q}$.

3) T randomly selects i, j and $\mathrm{j} \neq \mathrm{i}$. Let $\mathrm{a}_1 = \mathrm{a} - 2^{\mathrm{i}}$, $a_2 = a - 2^{j}$, $2^{i} < a$, $2^{j} < a$.

4) T runs Rand to obtain eight pairs $(\mathrm{t}_1, \mathrm{g}^{\mathrm{t}_1})$, $(\mathrm{t}_2, \mathrm{g}^{\mathrm{t}_2})$, $(\mathrm{s}_1, \mathrm{g}^{\mathrm{s}_1})$, $(\mathrm{s}_2, \mathrm{g}^{\mathrm{s}_2})$, $(\mathrm{s}_3, \mathrm{g}^{\mathrm{s}_3})$, $(\mathrm{s}_4, \mathrm{g}^{\mathrm{s}_4})$, $(\mathrm{s}_5, \mathrm{g}^{\mathrm{s}_5})$, $(\mathrm{s}_6, \mathrm{g}^{\mathrm{s}_6})$. T then randomly chooses $\mathrm{m}_1, \cdots, \mathrm{m}_{\mathrm{i}-1}, \mathrm{m}_{\mathrm{i}+1}, \cdots, \mathrm{m}_{\mathrm{j}-1}, \mathrm{m}_{\mathrm{j}+1}, \cdots, \mathrm{m}_{\mathrm{n}} \in \mathrm{Z}_{\mathrm{p}}^{*}$.

5) Next, T queries U in random order as:

$$\begin{aligned}
U(\frac{\gamma}{t_1}, g^{t_1}) &\rightarrow g^{\gamma}, \\
U(a_1, wg^{s_1}) &\rightarrow R_{11} = w^{a_1} g^{s_1 a_1}, \\
U(\frac{s_1 a_1 - s_2}{s_3}, g^{s_3}) &\rightarrow R_{12} = g^{s_1 a_1 - s_2}, \\
U(\frac{\tau}{t_2}, g^{t_2}) &\rightarrow g^{\tau}, \\
U(a_2, vg^{s_4}) &\rightarrow R_{21} = v^{a_2} g^{s_4 a_2}, \\
U(\frac{s_4 a_2 - s_5}{s_6}, g^{s_6}) &\rightarrow R_{22} = g^{s_4 a_2 - s_5}, \\
U(2, m_1) &\rightarrow m[1] = m_1^2, \\
U(4, m_1) &\rightarrow m[2] = m_2^4, \\
&\cdots \\
U(2^i, w) &\rightarrow m[i] = w^{2^i}, \\
&\cdots \\
U(2^j, v^{-1}) &\rightarrow m[j] = v^{-2^j}, \\
&\cdots \\
U(2^n, m_n) &\rightarrow m[n] = m_n^{2^n}.
\end{aligned}$$

Note 1. *Considering both the client and the server computing devices are stored in binary, so we don't think the binary conversion cost computing time.*

6) T computes:

$$\begin{aligned}
w^{a_1} &= R_{11}(R_{12} g^{s_2})^{-1}, \\
v^{a_2} &= R_{21}(R_{22} g^{s_5})^{-1},
\end{aligned}$$

and then checks that whether U produces the correct outputs, i.e.,

$$\mathrm{Bg}^{\gamma} \mathrm{w}^{\mathrm{a}_1} \mathrm{m[i]m[j]} \bmod \mathrm{p} = \mathrm{Dg}^{\tau} \mathrm{v}^{\mathrm{a}_2} \bmod \mathrm{p}. \quad (1)$$

If not, T outputs "error", otherwise, T computes:

$$\mathrm{u}^{\mathrm{a}} = \mathrm{Bg}^{\gamma} \mathrm{w}^{\mathrm{a}_1} \mathrm{m[i]} \bmod \mathrm{p}.$$

3.2 Security Analysis

Lemma 1. (Correctness): *In a single untrusted model, the algorithm (T, U') presented in Section 3.1 is a correct implementation of SgExp.*

Proof. As described in Section 3.1, we know that $\mathrm{R}_{11} = \mathrm{w}^{\mathrm{a}_1} \mathrm{g}^{\mathrm{s}_1 \mathrm{a}_1}, \mathrm{R}_{12} = \mathrm{g}^{\mathrm{s}_1 \mathrm{a}_1 - \mathrm{s}_2}$. So, $\mathrm{R}_{11}(\mathrm{R}_12\mathrm{g}^{\mathrm{s}_2})^{-1} = \mathrm{w}^{\mathrm{a}_1}$. In addition, $\mathrm{w}^{\mathrm{a}} = \mathrm{w}^{\mathrm{a}_1 + 2^{\mathrm{i}}} = \mathrm{w}^{\mathrm{a}_1} \mathrm{w}^{2^{\mathrm{i}}} = \mathrm{w}^{\mathrm{a}_1} \mathrm{m[i]}$. Therefore,

$$\begin{aligned}
\mathrm{u}^{\mathrm{a}} &= g^{\beta} g^{\gamma} w^{a} \bmod p \\
&= g^{\beta} g^{\gamma} w^{a_1} m[i].
\end{aligned}$$

Similarly, T compute:

$$\begin{aligned}
v^{a_2} &= R_{21}(R_{22} g^{s_5})^{-1}, \\
u^{a} &= g^{\theta} g^{\tau} v^{z} \bmod p \\
&= g^{\theta} g^{\tau} v^{a_2} m[j]^{-1}.
\end{aligned}$$

□

Theorem 1. (Security): *In single untrusted program model, the algorithms (T, U) are an outsource-secure implementation of SgExp, where the inputs (a, u) may be honest, secret; honest, protected; or adversarial, protected.*

Proof. Firstly, we show Pair One $\mathrm{EView_{real}} \sim \mathrm{EView_{ideal}}$ holds, which means the adversarial environment E leans nothing during the execution of (T, U'). □

If the input (a, u) is honest, protected or adversarial, protected, the simulator $\mathrm{S_1}$ behaves same as in the real execution. Thus, it needs only to consider the case where (a, u) is an honest, secret input.

So, suppose (a, u) is an honest, secret input. The simulator $\mathrm{S_1}$ in the ideal experiment behaves as follows. When receiving the input in the i-th round, $\mathrm{S_1}$ ignores it and instead submits 6 random queries with the form $(\mathrm{a_j}, \mathrm{u_j})$ and n random queries with the form $(2^{\mathrm{k}}, \mathrm{u'_j})(\mathrm{k} \in 1, 2, \cdots, \mathrm{n})$ to U'. Then $\mathrm{S_1}$ tests 6 outputs $(\mathrm{u_j^{a_j}})$ and 2 random outputs $(\mathrm{u'^{2^k}_j})$ from U'. If an error is detected, $\mathrm{S_1}$ outputs $\mathrm{Y^i_p}$ = "error", $\mathrm{Y^i_u}$ = "ϕ", ind^i=1. If no error is checked, $\mathrm{S_1}$ verifies the remaining outputs. If all checks pass, $\mathrm{S_1}$ outputs $\mathrm{Y^i_p}$ "ϕ", $\mathrm{Y^i_u}$ = "ϕ", $ind^i = 0$; else, $\mathrm{S_1}$ chooses a random element $\mathrm{r} \in \mathrm{Z^*_p}$, and outputs $\mathrm{Y^i_p}$ = "r", $\mathrm{Y^i_u}$ = "ϕ", $ind^i = 1$. In either condition, $\mathrm{S_1}$ saves its own states and those of U'.

As same as [11], we need to show that the input distribution to U' in the real experiment is computationally distinguished from that in the ideal one. In the ideal experiment, the inputs are all chosen randomly and uniformly distributed. While in the real experiment, each part of all queries that T makes to U' is generated by invoking the subroutine Rand and thus computationally indistinguishable from random. Thus, we have three possible cases to consider. If U' behaves honest in the i-th round, $\mathrm{EView^i_{real}} \sim \mathrm{EView^i_{ideal}}$, because the outputs of SgExp in the ideal experiment are not replaced. If U' misbehaves in the i-th round, it will be caught by both T and $\mathrm{S_1}$ with probability $1 - 1/n^2$, and then the experiment outputs "error"; otherwise, the outputs of SgExp are corrupted by U'. In the real experiment, the outputs generated by U' is multiplied together with random values produced by invoking Rand, thus the corrupted outputs of SgExp look random to E. While in the ideal experiment, the outputs of U' are replaced by a random value r. So, we conclude that $\mathrm{EView^i_{real}} \sim \mathrm{EView^i_{ideal}}$ also holds even if U' is dishonest in the i-th round. In all, by the hybrid argument we have $\mathrm{EView_{real}} \sim \mathrm{EView_{ideal}}$.

Secondly, we show Pair Two $\mathrm{UView_{real}} \sim \mathrm{UView_{ideal}}$ holds, that is to say, the adversarial software U' leans nothing during the execution of (T, U').

In the ideal experiment, the simulator $\mathrm{S_2}$ behaves as follows. On receiving the inputs in the i-th round, $\mathrm{S_2}$ ignore them but submits 6 random queries of the form $(\mathrm{a_j}, \mathrm{u_j})$ and n random queries of the form $(2^{\mathrm{k}}, \mathrm{u'_j})$ $(k \in \{1, 2, \cdots, n\})$ to U', and then $\mathrm{S_2}$ saves its own states and that of U'. Since all three kinds of inputs are unknown to U', the simulator $\mathrm{S_2}$ is applicable to all those cases. As we know, E can easily distinguish the real and ideal experiments since the outputs of the ideal experiment are never corrupted, but he cannot send the information to U' since they cannot communicate each other during the execution of $\mathrm{T^U}$. In addition, the input distribution to U' in the real experiment is computationally indistinguishable from that in the ideal experiment randomly chosen by $\mathrm{S_2}$. Thus, $\mathrm{UView^i_{real}} \sim \mathrm{UView^i_{ideal}}$ holds for each round i. In all, we conclude that $\mathrm{UView_{real}} \sim \mathrm{UView_{ideal}}$.

Theorem 2. *In one untrusted model, the algorithm (T, U) presented in Section 3.1 is an* $(\mathrm{O}((\log^2 \mathrm{n})/\mathrm{n}), 1 - 1/n^2)$*-outsource-secure implementation of SgExp, where m is the bit length of a.*

Proof. In order to compute $\mathrm{u^a}$ mod p, SgExp requires 12 calls to Rand, 22 modular multiplication and 8 modular inverse. It takes $O(1)$ or $O(\log^2 n)$ modular multiplication by using table-lookup method or the EBPV generator, where n is the bit of q. As we know, it takes roughly $1.5n$ modular multiplication to compute $\mathrm{u^a}$ mod p by the square-and-multiply method. Therefore, the algorithm (T, U) is an $O((\log^2 n)/n)$-efficient execution of SgExp. □

On the other side, U cannot cheat the outsourcer to accept an error result unless that he knows i and j. Since i and j are randomly chosen from $1, 2, \cdots, \mathrm{n}$, the outsourcer can verify an outsourcing result with probability $1 - 1/n^2$.

3.3 Comparison

We make a comparison between our scheme SgExp and the schemes in [5, 19] in Table 1, where n is the bit length of q.

Note that the scheme [19] needs one modular exponentiation operation and has no advantage for single modular exponentiation. The proposed scheme is superior in checkability, and it is more efficient than that of the scheme in [19]. Our scheme only uses one server but the scheme in [5] is based on two untrusted server though we need more modular multiplications and modular inversions for the outsourcer and more queries to U. Therefore, the proposed scheme improves the checkability based on only one server.

4 Outsourcing Algorithm of Simultaneous Modular Exponentiations

We also extend our scheme to an outsourcing algorithm of simultaneous modular exponentiations (SmExp) $\mathrm{u_1^a u_2^b}$ mod p which has important applications in many cryptographic schemes such as trapdoor commitment [3, 8, 9, 16] and chameleon hashing [1, 2, 6, 10].

4.1 Outsourcing Algorithm

There are two bases $\mathrm{u_1, u_2} \in \mathrm{Z^*_p}$ and two powers $\mathrm{a, b} \in \mathrm{Z^*_q}$. We need to compute $\mathrm{u_1^a u_2^b}$ mod p.

Table 1: Comparison of outsourcing single exponentiation

	GExp[8]	Exp[1]	SgExp
Rand	7	5	12
Modular Multiplications	12	7	22
Modular Exponentiation	1	0	0
Modular Inversions	4	3	9
Queries to U	4	6	n+6
Privacy	√	√	√
Checkability	1/2	2/3	$1-1/n^2$
The number of servers	Single server	Two servers	Single server

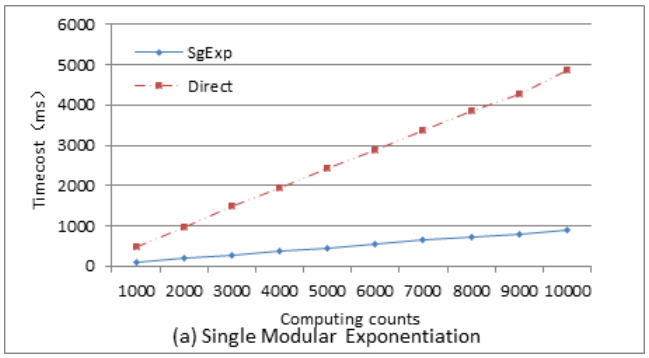


Figure 2: Simulation for SgExp algorithm

1) First, T invokes the subroutine Rand four times to create four blinding pairs (α, g^{α}), (β, g^{β}), $(\varepsilon, g^{\varepsilon})$, (θ, g^{θ}). We denote:

$$\begin{aligned} A &= g^{\alpha} \bmod p, \\ B &= g^{\beta} \bmod p, \\ C &= g^{\varepsilon} \bmod p, \\ D &= g^{\theta} \bmod p. \end{aligned}$$

2) The first logical divisions are represented as below:

$$\begin{aligned} u_1^a u_2^b &= (Aw_1)^a (Aw_2)^b \\ &= g^{\beta} g^{\gamma} w_1^a w_2^b, \end{aligned}$$

where $\mathrm{w_1 = u_1/A}$, $\mathrm{w_2 = u_2/A}$, and $\gamma = \alpha(\mathrm{b+a}) - \beta$. Similarly, T executes the second divisions as follows:

$$\begin{aligned} u_1^a u_2^b &= (Cv_1)^a (Cv_2)^b \\ &= g^{\theta} g^{\tau} v_1^a v_2^b, \end{aligned}$$

where $\mathrm{v_1 = u_1/C, v_2 = u_2/C}$, and $\tau = \varepsilon(\mathrm{a+b}) - \theta$.

3) T randomly selects k, h, i, j and $\mathrm{k < h < i < j}$. Let $\mathrm{a_1 = a - 2^k, a_2 = a - 2^h, b_1 = b - 2^i, b_2 = b - 2^j}$.

4) T runs Rand for ten times to obtain ten pairs $(\mathrm{t_1, g^{t_1}})$, $(\mathrm{t_2, g^{t_2}})$, $(\mathrm{s_1, g^{s_1}})$, $(\mathrm{s_2, g^{s_2}})$, $(\mathrm{s_3, g^{s_3}})$, $(\mathrm{s_4, g^{s_4}})$, $(\mathrm{s_5, g^{s_5}})$, $(\mathrm{s_6, g^{s_6}})$, $(\mathrm{s_7, g^{s_7}})$, $(\mathrm{s_8, g^{s_8}})$. T randomly generated n-4 integers $\mathrm{m_1, m_2, \cdots, m_{k-1}, m_{k+1}, \cdots, m_{h-1}, m_{h+1}, \cdots m_{i-1}, m_{i+1}, \cdots, m_{j-1}, m_{j+1}, \cdots, m_n} \in \mathrm{Z_p^*}$.

5) T queries U in random order as:

$$\begin{aligned} U(\frac{\gamma}{t_1}, g^{t_1}) &\rightarrow g^{\gamma}, \\ U(a_1, w_1 g^{s_1}) &\rightarrow R_{11} = w_1^{a_1} g^{s_1 a_1}, \\ U(a_1, w_1 g^{s_1}) &\rightarrow R_{11} = w_1^{a_1} g^{s_1 a_1}, \\ U(\frac{s_1 a_1 - s_2}{s_3}, g^{s_3}) &\rightarrow R_{12} = g^{s_1 a_1 - s_2}, \\ U(b_1, w_2 g^{s_1}) &\rightarrow R_{21} = w_2^{b_1} g^{s_1 b_1}, \\ U(\frac{s_1 b_1 - s_2}{s_4}, g^{s_4}) &\rightarrow R_{22} = g^{s_1 b_1 - s_2}, \\ U(\frac{\tau}{t_2}, g^{t_2}) &\rightarrow g^{\tau}, \\ U(a_2, v_1 g^{s_5}) &\rightarrow R_{31} = v_1^{a_2} g^{s_5 a_2}, \\ U(\frac{s_5 a_2 - s_6}{s_7}, g^{s_7}) &\rightarrow R_{32} = g^{s_5 a_2 - s_6}, \\ U(b_2, v_2 g^{s_1}) &\rightarrow R_{41} = v_2^{b_2} g^{s_1 b_2}, \\ U(\frac{s_5 a_2 - s_6}{s_8}, g^{s_8}) &\rightarrow R_{42} = g^{s_5 a_2 - s_6}, \\ U(2, m_1) &\rightarrow m[1] = m_1^2, \\ U(4, m_1) &\rightarrow m[2] = m_2^4, \\ &\cdots \\ U(2^k, w_1) &\rightarrow m[k] = w_1^{2^k}, \\ &\cdots \\ U(2^h, v_1^{-1}) &\rightarrow m[h] = v_1^{-2^h}, \\ &\cdots \\ U(2^i, w_2) &\rightarrow m[i] = w_2^{2^i}, \\ &\cdots \\ U(2^j, v_2) &\rightarrow m[j] = v_2^{2^j}, \\ &\cdots \\ U(2^n, m_n) &\rightarrow m[n] = m_n^{2^n}. \end{aligned}$$

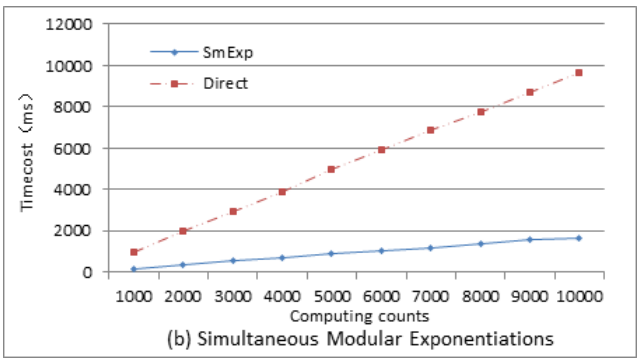


Figure 3: Simulation for SmExp algorithm

6) T computes:

$$\begin{aligned} w_1^{a_1} &= R_{11}(R_{12}g^{s_2})^{-1}, \\ w_2^{b_1} &= R_{21}(R_{22}g^{s_2})^{-1}, \\ v_1^{a_2} &= R_{31}(R_{32}g^{s_6})^{-1}, \\ v_2^{b_2} &= R_{41}(R_{42}g^{s_6})^{-1}. \end{aligned}$$

7) Finally, T checks whether U produces the correct outputs, i.e.,

$$Bg^{\gamma}w_1^{a_1}w_2^{b_1}m[k]m[i]m[h] = Dg^{\tau}v_1^{a_2}v_2^{b_2}m[j] \bmod p.$$

If not, T outputs "error", otherwise, T computes:

$$u_1^a u_2^b = Bg^{\gamma}w_1^{a_1}w_2^{b_1}m[k]m[i] \bmod p.$$

4.2 Security Analysis

Lemma 2. (Correctness): *In a single untrusted program model, the algorithm (T, U) presented in Section 4.1 is a correct implementation of SmExp.*

As described in Section 4.1, we know that

$$\begin{aligned} R_{11} &= w_1^{a_1}g^{s_1a_1}, \\ R_{12} &= g^{s_1a_1-s_2}, \\ R_{21} &= w_2^{b_1}g^{s_1b_1}, \\ R_{22} &= g^{s_1b_1-s_2}. \end{aligned}$$

So,

$$\begin{aligned} R_{11}(R_{12}g^{s_2})^{-1} &= \mathrm{w}_1^{\mathrm{a}_1}, \\ R_{21}(R_{22}g^{s_2})^{-1} &= \mathrm{w}_2^{b_1}. \end{aligned}$$

In addition,

$$\begin{aligned} w_1^a &= w_1^{a_1+2^k} \\ &= w_1^{a_1}w_1^{2^k} \\ &= w_1^{a_1}m[k], \\ w_2^b &= w_2^{b_1+2^i} \\ &= w_2^{b_1}w_2^{2^i} \\ &= w_1^{b_1}m[i]. \end{aligned}$$

Therefore,

$$\begin{aligned} u_1^a u_2^b &= g^{\beta}g^{\gamma}w_1^a w_2^b \bmod p \\ &= g^{\beta}g^{\gamma}w_1^{a_1}w_2^{b_1}m[k]m[i] \bmod p. \end{aligned}$$

Similarly, we compute:

$$\begin{aligned} v_1^{a_2} &= R_{31}(R_{32}g^{s_2})^{-1}, \\ v_2^{b_2} &= R_{41}(R_{42}g^{s_2})^{-1}, \\ u_1^a u_2^b &= g^{\theta}g^{\tau}v_1^a v_2^b \bmod p \\ &= Dg^{\tau}v_1^{a_1}v_2^{b_1}m[j]m[h]^{-1} \bmod p. \end{aligned}$$

As same as Theorems 1 and 2, we can easily prove the following theorems.

Theorem 3. (Security): *In one untrusted model, the algorithm (T, U) presented in Section 4.1 is an outsource-secure implementation of SmExp, where the inputs* $(\mathrm{a}, \mathrm{b}; \mathrm{u}_1\mathrm{u}_2)$ *may be honest, secret; honest, protected; or adversarial, protected.*

Theorem 4. *In one untrusted model, the algorithm (T, U) presented in Section 4.1 is an $(O((\log^2 n)/n), 1 - 1/n^4)$-outsource-secure execution of SmExp, where n is the bit length of q.*

Proof. As Theorem 2, the algorithm (T, U) presented in Section 4.1 is an $O((\log^2 n)/n)$-efficient implementation of SmExp. On the other hand, U cannot cheat the outsourcer to accept an error result unless that he knows k, h, i and j. Since k, h, i and j are randomly chosen from $\{1, 2, \cdots, \mathrm{n}\}$, the outsourcer can verify an outsourcing result with probability $1 - 1/n^4$. □

4.3 Efficiency

We also make a comparison among the proposed SmExp algorithm and other outsourcing algorithms of simultaneous exponentiation. The comparison is given in Table 2.

Note that the proposed SmExp algorithm is more efficient then the GExp algorithm since no modular exponentiation is needed. For the outsourcer, the SmExp algorithm improves the checkability based on only one server though it needs more modular multiplications and modular inversions.

5 Performance Evaluation

The experimental evaluation of the proposed outsourcing algorithms will be provided in this section. Our experiment is simulated on two Windows machines with Intel(R) Core(TM) i5-3470 CPU running at 3.20 GHz and 4G memory (local user), and Intel(R) Core(TM) i7-3770 CPU running at GHz and 16G memory (cloud server), respectively. We choose C^{++} as the programming language. The Multiple Precision Integers and Rational Library (MPIR) are used in our experiments.

Table 2: Comparison of outsourcing simultaneous exponentiation

	GExp[8]	SExp[1]	SmExp
Rand	8	5	14
Modular Multiplications	17	10	38
Modular Exponentiation	1	0	0
Modular Inversions	4	4	11
Queries to U	6	8	n+10
Privacy	√	√	√
Checkability	1/3	2/3	$1-1/n^4$
Security Model	Single server	Two servers	Single server

The parameters of p and q are same to Federal Information Processing Standards for DSA (FIPS-186-2). That is, p is a 512-bit prime and q|p − 1 is a 160-bit prime.

$$p = 8df2a494492276aa3d25759bb06869cbeac0d83a\ fb8d0cf7cbb8324f0d7882e5d0762fc5b7210eaf\ c2e9adac32ab7aac49693dfbf83724c2ec0736ee\ 31c80291.$$

$$q = c773218c737ec8ee993b4f2ded30f48edace915f.$$

In Figure 2 and Figure 3, we provide the simulation of SgExp and SmExp algorithm, which means that the fault can be found with probability close to 1 if the server misbehaves. It is obvious that the time cost for the outsourcer T is much smaller than that for directly computing single modular exponentiation and simultaneously modular exponentiations since that a number of computations have been delegated to the server. Therefore, the proposed SgExp and SmExp algorithm are the implementations of secure and verifiable outsourcing for single modular exponentiation and simultaneously modular exponentiations.

In addition, we provide the evaluation time of the outsourcer for single modular exponentiation and simultaneously modular exponentiations proposed in [5, 19] and our paper. We show the result in Table 3. From Table 3, we conclude that for the outsourcer, the proposed SgExp and SmExp algorithm are more efficient than GExp proposed in [19]. The proposed algorithms improve the checkability based on one server though more time is needed than Exp and SExp of [5].

6 Conclusions

In this paper, we first propose an outsourcing algorithm for single modular exponentiation based on one server, and then extend the algorithm to secure outsourcing of simultaneous modular exponentiations. In the proposed two algorithms, the outsourcer can verify the outsourcing result efficiently and detect the error with probability close to 1. Our algorithms are superior in checkability and more efficient than that of the previous ones based on one untrusted server.

Acknowledgments

The work described in this paper was supported by the Natural Science Foundation of China (Grant No. 61373151, 61472235, 61572309, U1536108), and the Innovation Program of Shanghai Municipal Education Commission (Grant No. 14YZ020).

References

[1] G. Ateniese, B. de Medeiros, "On the key exposure problem in chameleon hashes," in *Security in Communication Networks (SCN'05)*, LNCS 3352, pp. 165–179, Springer, 2005.

[2] G. Ateniese, B. de Medeiros, "Identity-based chameleon hash and applications," in *International Conference on Financial Cryptography (FC'04)*, LNCS 3110, pp. 164–180, Springer, 2004.

[3] G. Brassard, D. Chaum, C. Cepeau, "Minimum disclosure proofs of knowledge," *Journal of Computer and System Sciences*, vol. 37, no. 2, pp. 156–189, 1988.

[4] Z. Cao, L. Liu, "A note on two schemes for secure outsourcing of linear programming," *International Journal of Network Security*, vol. 19, no. 2, pp. 323–326, 2017.

[5] X. Chen, J. Li, J. Ma, et al., "New algorithms for secure outsourcing of modular exponentiations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2386–2396, 2014.

[6] X. Chen, F. Zhang, W. Susilo, et al., "Efficient generic on-line/off-line signatures without key exposure," in *Applied Cryptography and Network Security (ACNS'07)*, LNCS 4521, pp. 18–30, Springer, 2007.

[7] M. Van Dijk, D. Clarke, B. Gassend, et al., "Speeding up exponentiation using an untrusted computational resource," *Designs, Codes and Cryptography*, vol. 39, no. 2, pp. 253–273, 2006.

[8] X. Dong, "A multi-secret sharing scheme based on the CRT and RSA," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 47–51, 2015.

Table 3: Time comparison for modular exponentiation

T/ms	GExp[8]	Exp(SExp)[1]	Our scheme
Single	0.511	0.031	0.091
Simultaneous	0.536	0.056	0.169

[9] J. A. Garay, P. MacKenzie, K. Yang, "Strengthening zero-knowledge protocols using signatures," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 177–194, 2003.

[10] W. R. Ghanem, M. Shokir, and M. Dessoky, "Defense Against Selfish PUEA in Cognitive Radio Networks Based on Hash Message Authentication Code," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 12–21, 2016.

[11] S. Hohenberger, A. Lysyanskaya, "How to securely outsource cryptographic computations," in *Theory of Cryptography (TCC'05)*, LNCS 3378, pp. 264–282, Springer, 2005.

[12] K. Li and H. Ma, "Outsourcing decryption of multi-authority ABE ciphertexts," *International Journal of Network Security*, vol. 16, no. 4, pp. 286–294, 2014.

[13] X. Ma, J. Li, F. Zhang, "Outsourcing computation of modular exponentiations in cloud computing," *Cluster Computing*, vol. 16, no. 4, pp. 787–796, 2013.

[14] X. Ma, F. Zhang, J. Li, "Verifiable evaluation of private polynomials," in *Fourth IEEE International Conference on Emerging Intelligent Data and Web Technologies (EIDWT'13)*, pp. 451–458, 2013.

[15] P. Q. Nguyen, I. E. Shparlinski, J. Stern, "Distribution of modular sums and the security of the server aided exponentiation," in *Cryptography and Computational Number Theory*, pp. 331–342, 2001.

[16] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual International Cryptology Conference (CRYPTO'88)*, LNCS 576, pp. 129–140, Springer, 1991.

[17] K. Peng, "Critical survey of existing publicly verifiable secret sharing schemes," *IET Information Security*, vol. 6, no. 4, pp. 249–257, 2012.

[18] J. Singh, "Cyber-attacks in cloud computing: A case study," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 78–87, 2014.

[19] Y. Wang, Q. Wu, D. S. Wong, et al., "Securely outsourcing exponentiations with single untrusted program for cloud storage," in *19th European Symposium on Research in Computer Security (ESORICS'14)*, LNCS 8712, pp. 326–343, Springer, 2014.

[20] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[21] F. Zhang, S. Rehaneh, "Private outsourcing of polynomial evaluation and matrix multiplication using multilinear maps," in *Cryptology and Network Security*, LNCS 8257, pp. 329–348, Springer, 2013.

[22] M. Zareapoor, P. Shamsolmoali, and M. A. Alam, "Establishing safe cloud: Ensuring data security and performance evaluation," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 88–99, 2014.

Biography

Jianxing Cai is a master candidate of Shanghai University. His research interests include public key cryptography and verifiable outsourcing computation.

Yanli Ren is an associate professor in School of Communication and Information Engineering at Shanghai University. She was awarded a MS degree in applied mathematics in 2005 from Shanghai Normal University, China, and a PhD degree in computer science and technology in 2009 from Shanghai Jiao Tong University, China. Her research interests include applied cryptography, secure computing, and network security.

Chunshui Huang is a master of Shanghai University. His research interests include public key cryptography and verifiable outsourcing computation.

Application of Community Detection Algorithm with Link Clustering in Inhibition of Social Network Worms

Yibing Wang[1], Jie Fang[1,2], and Fuhu Wu[3]
(Corresponding author: Yibing Wang)

Center of Computer Teaching, Anhui University[1]
No.111 Jiulong Road, Hefei, Anhui 236061, China
School of Electronics and Information Engineering, Chinese Academy of Sciences[2]
No.350 Shushanhu Road, Hefei, Anhui 230031, China
Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Anhui University[3]
No.3 Feixi Road, Hefei, Anhui 230039, China
(Email: wyb@ahu.edu.cn)
(Received Dec. 15, 2015; revised and accepted Mar. 6 & Mar. 28, 2016)

Abstract

The community detection was performed from the perspective of links, and we proposed an inhibition method against social network worms. Firstly, a community detection algorithm was proposed, which based on link clustering, and we got related link incremental information through the network structure information at various time points. In order to obtain the link communities, we adopted an improved link partition density function to dispose the link incremental information. Next, we gave three selection strategies of key nodes in community and proposed corresponding worm inhibition method. Finally, on the basis of real web data sets, we applied community detection and worm inhibition experiments to prove validity of algorithm in this paper.

Keywords: Community detection, link clustering, partition density, worm inhibition

1 Introduction

Social network connects users in the virtual network space, extends the human communication, information sharing and the social activity space, which is becoming the most influential internet application. The typical applications include Facebook, QQ, Renren, Sina Weibo, BBS and other shared spaces, etc. [25, 30].

As an extension of real world in the virtual network world, "birds of a feather flock together, and people of one mind fall into the same group", community structure is an important structure of social networks, which is also a kind of important structure for mesoscopic observation and the network topology analysis [29]. It makes the community internal nodes closer, and the connection between communities looser [19]. The process of finding community structure in the complex network is the community detection, which has important theoretical basis and practical significance for the network structure analysis in real world.

In recent years, based on the different understanding of community structure, scholars have put forward a lot of community detection algorithms [2, 8, 13, 18, 24, 28, 33]. At present, some algorithms can correctly extract community structure from small-scale social networks, which can be roughly divided into three categories: the method based on graph theory, such as GN [8] and FastGN [18]; Algorithms based on matrix decomposition, such as SymNMF [13]; The method based on the optimization, such as N-Cut and A-Cut [24], etc. Among them, in 2001, Girvan and Newman proposed the GN algorithm, setting off a new wave of research. In recent years, it has become a standard algorithm of community structure analysis. GN makes up the inadequacy of some traditional algorithms, which does not have to rely on redundant information, and can directly analyze from the network topological structure. But the biggest drawback is that it is unable to determine when to terminate the operation, eventually making the results too granular. The time complexity of this algorithm is $O(n^3)$, where n is the number of nodes in network. In order to improve the time-consuming shortcoming of GN, Newman proposed the FastGN where each node was seen as a community, and the two communities combined with the maximum Q value in each iteration until the entire network integrated into one community. The whole process can be represented as a tree diagram, and choose the hierarchical division with maximum Q value

to get the final community structure. The overall time complexity of the algorithm is $O(m(m+n))$, where m is the number of edges in network, and n is the number of nodes. This method makes to lower the time complexity of GN greatly. Clauset et al. [5] used stack to calculate and update the network modularity, and proposed a new greedy algorithm — CNM. The algorithm further accelerates the FastGN, getting close to linear complexity.

The community structure of large-scale complex networks, however, often has overlapping characteristics, that is, a node belongs to different communities. Both GN and its improved algorithms have a problem, that is, a node only belongs to one community. But it is not the case; each node can have different identities in different circumstances. In order to solve this problem, Palla et al. [22] proposed a clique filtering algorithm to analyze the overlapping community structure, introduced the concept of k-clique community. Ahn et al. [1] put forward the new idea to detect the community structure with overlaps and hierarchy — the edge detection algorithm. But these methods failed to solve the problem that a node belonged to multiple communities.

In addition, in order to avoid the limitation of priori information, Raghavan et al. [23] proposed a fast Label Propagation Algorithm (LPA) based on the idea. The algorithm firstly assigned the only label for each node. In every iteration, each node updated its own label to the label most frequently appears in neighboring nodes. If there were many same labels, randomly selected one as an updated value, after several iterations, the densely connected nodes would converge to the same label. In the end, the node with the same label would come into one community. LPA is simple, rapid and effective, but lacks high accuracy.

The above algorithms are only effective in the community detection in the small-scale networks; when the network scale is increasing, the efficiency will decrease obviously, and the algorithm complexity also increases exponentially along with the growth of the network dimension. Researchers adopt different standards and policies when partitioning nodes, deriving a lot of different styles of the new algorithms [17], including module optimization algorithm, spectrum analytical method, information theory method, and label transmission method. However, it is hard for these methods to find a good balance point between time complexity and accuracy.

Although many achievements have been made about community detection of complex network at home and abroad, some problems exist in these methods, that is, the algorithms are usually designed for a specific network or certain features of network, which are not suitable for most networks. At present, through the relevant research and analysis, it can be found that mining overlapping community structure is of great significance from the angle of link [1]:

1) Compared to the independent nodes in the network, the link between nodes can express more information;
2) Abstract network into a large number of links, mine these links sets to directly get the overlapping community structure, which is an intuitionistic expression without other auxiliary measures. These suggest that: finding overlapping community structure in complex networks from the perspective of link is more convenient. Therefore, this paper proposes the Link Clustering based Community detection algorithm (LCC); first, obtain the related link incremental information [6] through the network structure at any time, and then handle the link incremental information based on the improved link partition density function, with the improved link module as the objective function, then the link communities are obtained.

With the continuous development and large-scale popularity of social networks, some security issues start to emerge. For example, real-time resource sharing and interactive services provided in social network have attracted a large number of users, while, in the meantime, the frequent interaction between users also provides an effective way for the rapid spread of Internet worm. Different from traditional worm viruses, social network worm is a kind of malicious program that does not rely on particular system vulnerabilities. It uses its own camouflage to deceive users to click and execute the program to get infected, then it spreads through social networks to infect the user's friends, a large number of clicks and sharing among friends accelerate the proliferation of worms. Social network worm is characterized by high concealment, long life cycle, difficult to eradicate, etc. It is difficult to effectively control its dissemination through technology of patches release, which increases its potential damage. At the same time, with the increasing number of Internet users and the rapid development of various forms of virtual social networks, the social network worm has become one of the major hidden dangers for network security.

Traditional worm model is based on mathematical model, considering the similarity in propagation between computer worm and biological virus, it introduces SIS, SIR and other models which are widely used in biological virus propagation modal into computer worm model, so as to analyze and predict the features and trends of worm propagation [14]. Researchers begin to realize such external factors as the network topology, bandwidth and user countermeasures impact on the spread of worm propagation. For example: Yang and others [32] took Rose mail worm as an example, by establishing the mathematical model, they researched worm propagation in different social occasions such as Print Service Office Internet Cafe Friendship Network, and these occasions have added immune factors; Considering the following two factors could affect velocity of worm propagation: first one, the network users' countermeasures to worm, second one, fast-moving worm leads to retardation because of router block, Zou and others [35] proposed Two-Factor model, worm infection rate , host immunity and some parameters

were expressed as a Function of time T, and adjusted its value according to the change of infected host quantity. The models mentioned above only describe nodes infected number in unit time, but they can't reflect worm's infection route in network topology. In Reference 18, based on social topology, Faghani and others proposed XXS worm model which used undirected network topology, whereas, it was not conform to aeolotropism in real social network topology.

In Reference 19, Nguyen and others presented primarily worm inhibition method which based upon community structure, this method made use of popular BGLL, it was not necessary to provide division quantity to detect reasonable community structure for users. What's more, Nguyen and others gave selection strategies on key nodes in community, which means the nodes which possess the most connections between the community and other communities could be defined as key nodes, and gave these nodes immunization or issued patches primarily. But in their paper, the authors can't prove the selection of key nodes in theories or experiments.

In view of all kinds of hazards caused by social network worms, the defensive measures put forward by researchers mainly include two aspects:

1) Social network worm detection [3, 16, 26];

2) Social network worm inhibition [20, 36].

Among them, the social network worm detection can be divided into client and server detection according to the location of the detection. The client detection method mainly uses constantly updated feature library to match and detect the spread of worm. But when there is a new type of worm, limited by the bandwidth of existing network, it is hard for it to distribute the new features to the network test system of all users, so the method has certain delay. Server detection mainly captures the number of malicious messages in the network through the website server, but this method cannot detect worms until the malicious message spreads to a certain degree. So this method also has the unavoidable delay.

Although the worm inhibition method in social network is unable to timely detect the spread of the worm, it can reduce the number of infected users to maximum extent. At present, researchers generally start from the community structure of complex networks, in other words, find the community in the network first, and then adopt relevant measures to select key nodes in the community, finally, conduct immune operation for these key nodes, thus ensuring to immunize other nodes at full speed. This paper, based on such idea, conducts worms inhibition in social network. According to the choice strategy of key nodes in the community, we define the nodes connected with most other communities as the key nodes, then immune these nodes, finally immune the neighboring nodes with the help of these key nodes, thus effectively restraining the rapid spread of network worms.

The remained paper is divided into four sections. Section 2 introduces the concrete realization of Link Clustering based Community detection (LCC). Section 3 introduces the selection of key nodes and worms inhibition method, and Section 4 describes the experimental results on real data sets, and Section 5 summarizes the full paper.

2 Link Clustering Based Community Detection Algorithm

On the basis of existing methods, this work proposes a worm inhibition method, which based on dynamic community mining, as shown in Figure1. This method can be divided into four stages: original data pre-processing, dynamic community detection, key nodes abstraction, worm inhibition. First, through network structure information at various points, we get related link increment, and then we adopt improved Link Partition Density Function to process the increment information, counting the improved link modularity value as objective function, so as to detect community to get community structure. Next, we propose three different strategies to choose key nodes, analyze and compare worm inhibition effect under different strategies by comparison experiments in the fourth section. At last, we give these key nodes immunization, with the help of these nodes, give immunization to the neighbor nodes, in order to achieve the desired inhibitory effect of worm rapid spreading.

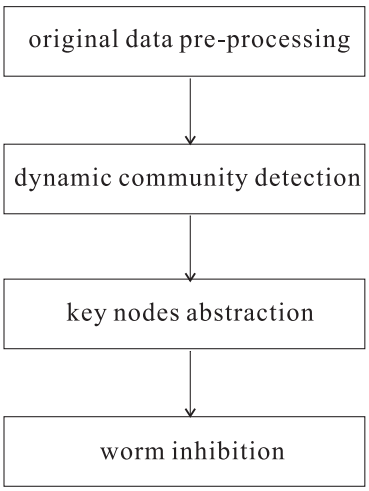


Figure 1: The diagram of worm inhibition

2.1 Link Partition Density

For a given network $G(V, E)$, where V is the node set in the network, and E is the edge set, $C = \{C_1, C_2, \cdots, C_k\}$ represent the community sets in the network.

Definition 1. *For the given network* G, *link graph* L_G *is the link aggregation formed by connection be-*

tween nodes in G, *as shown in Figure 2, where* $L_G = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\}$.

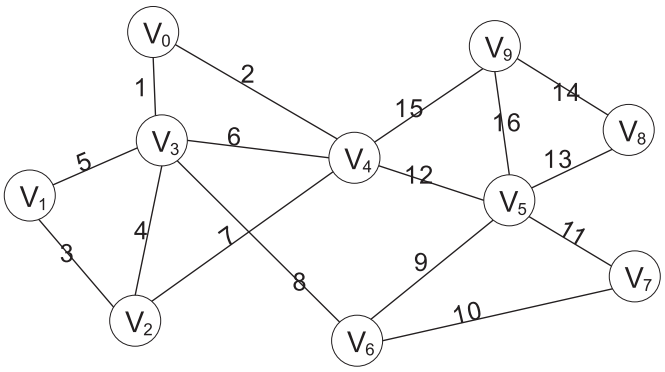


Figure 2: The link graph

Definition 2. *In the given link* $L_1(L_1 =< v_1, v_2 >)$ *and* $L_2(L_2 =< v_3, v_4 >)$, v_1, v_2, v_3, v_4 *are the nodes. If* $L_1 \cap L_2 =< v_1, v_2 > \cap < v_3, v_4 >\neq \emptyset$, *the link* L_1 *and* L_2 *are the neighboring links.*

In Figure 2, Link 1 and Link 2 share the node V_0, so Link 1 and Link 2 are the neighboring links.

Definition 3. *Given link communities* C_i, C_j *and link* $L =< v_1, v_2 >$ *in moment* t, *if* $\{L|L \notin C_i, L \notin C_j, v_1 \in C_i, v_2 \in C_j, i \neq j\}$, L *is the bridge link.*

Inspired by the literature [1], this paper proposes an improved link partition density function for dealing with incremental information. Assuming that a network has M links, and the network is divided into C link subsets by $\{P_1, P_2, \cdots, P_c\}$, the link partition density of community is:

$$D_c = \frac{m_c - (n_c - 1)}{\frac{n_c(n_c-1)}{2} - (n_c - 1)} - \frac{m_b - (n_b - 1)}{\frac{n_b(n_b-1)}{2} - (n_b - 1)}. \tag{1}$$

In the above formula, m_c and n_c respectively represent the link numbers and node numbers in the subset P_c, m_b represents the bridge link numbers between communities, n_b represents the node numbers between communities. And meet $m_c = |P_c|$, $n_c = |\cup_{e_{ij} \in P_c} \{i, j\}|$, $n_b = |\cup_{e_{ij}, i \in P_c, j \notin P_c} \{i, j\}|$. Then, the improved link partition density D_L is defined as:

$$\begin{aligned} D_L &= \sum_c \frac{m_c}{M} D_c \\ &= \frac{2}{M} \sum_c [m_c \frac{m_c - (n_c - 1)}{(n_c - 1)(n_c - 1)} - m_b \frac{m_b - (n_b - 1)}{(n_b - 2)(n_b - 1)}]. \end{aligned} \tag{2}$$

In the given network $G(V, E)$, $C = \{C_1, C_2, \cdots, C_k\}$, set the incremental information in the network as $\varepsilon = \{\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n\}$. In the network, the links can be divided into two categories: Intra-community Links (IL), that is, the two endpoints are within the community; Bridge-community Links (BL), that is, the two endpoints are located in different communities. For each community C in G, when adding IL or removing BL, the community will be closer and the network structure will be clearer. On the contrary, removing IL link or adding BL will make the network structure even vaguer. When there is no interference between the two communities, or the disturbance is small, adding or removing the link may form a new community structure. Therefore, in the update of community structure, the subtle changes of network structure will lead to the huge change of community. From the perspective of the link, with the passage of time, the change of network is in fact the link adding or removing. Thus, the change information in the network can be simply described as the adding of new link or the removing of existing link.

2.2 Adding Link Algorithm

Theorem 1. *If* C_i *is a community in network* G, *add any IL to* C_i, C_i *won't break down into smaller modules.*

Proof. Formula (1) shows the link partition density of C_i. Assuming that the incremental information ε_i represents adding an internal link e to the community C_i. Set $D_c^{'}$ as the link partition density while adding e into the community C_i, then

$$D_c^{'} = \frac{(m_c + 1) - (n_c - 1)}{\frac{n_c(n_c-1)}{2} - (n_c - 1)} - \frac{m_b - (n_b - 1)}{\frac{n_b(n_b-1)}{2} - (n_b - 1)}. \tag{3}$$

Obviously, $D_c^{'} > D_c$, therefore, when adding internal link e to community C_i, the community structure will be stronger. □

Theorem 2. *If the added link is between* C_i *and* C_j, *when the bridge link needs to be re-assigned, the community* C_i *and* C_j *are the first choice.*

Proof. Assuming that the added link e between C_i and C_j, because e's nodes are in the communities C_i and C_j, so when adding the link e to other communities, the value of D_L is not changed. For the community C_i, before adding link e, the link partition density is:

$$D_{L,i} = \frac{m_c - (n_c - 1)}{\frac{n_c(n_c-1)}{2} - (n_c - 1)} - \frac{m_b - (n_b - 1)}{\frac{n_b(n_b-1)}{2} - (n_b - 1)}. \tag{4}$$

□

When adding link e, the link partition density is:

$$D_{L,i}^{'} = \frac{m_c - (n_c - 1)}{\frac{n_c(n_c-1)}{2} - (n_c - 1)} - \frac{(m_b + 1) - n_b}{\frac{(n_b+1)n_b}{2} - n_b}. \tag{5}$$

Set $\Delta_1 = D'_{L,i} - D_{L,i}$, obviously $\Delta_1 > 0$, so after adding the link e into C_i, the link partition density of C_i will increase. Similarly, set $\Delta_2 = D'_{L,j} - D_{L,j}$, obviously $\Delta_2 > 0$, the link partition density of C_j also increases.

To sum up, if the added link is between C_i and C_j, the communities C_i and C_j are the first choice.

Deduction 1. *If the added bridge link* e *is between* C_i *and* C_j*, when meeting* $\Delta_d = D_{L,i}(E+e) - D_{L,j}(E+e) + D_{L,j}(E) - D_{L,i}(E) > 0$*, the bridge link* e *will be assigned to the community* C_i*; otherwise, the bridge link* e *will be assigned to* C_j*.*

Proof. Theorem 2 shows that if the added bridge link e is between C_i and C_j, the communities C_i and C_j are the first choice, then

$$\begin{aligned}\Delta_d &= \Delta_1 - \Delta_2 \\ &= (D'_{L,i} - D_{L,i}) - (D'_{L,j} - D_{L,j}) \\ &= D_{L,i}(E+e) - D_{L,j}(E+e) + D_{L,j}(E) - D_{L,i}(E).\end{aligned} \tag{6}$$

□

When $\Delta_d > 0$, the bridge link e should be assigned to C_i, while $\Delta_d < 0$, the bridge link e should be assigned to C_j.

When adding a new link e, there are two kinds of situations:

1) Link e is completely in community C_i;

2) Link e is between C_i and C_j, where $i \neq j$. For Case (1), according to Theorem 1, the community structure remains the same. For Case (2), based on Theorem 2, if the bridge link e is assigned to the new community, the community must be one of C_i and C_j. Deduction 1 shows the assigning criteria of bridge link e.

Therefore, the algorithm of adding link is described as Algorithm 1.

Algorithm 1 *adding_link*

1: Enter new link e and link community structure C_t in moment t.
2: Output the link community structure C_{t+1} in moment $t+1$.
3: If e is the internal link, then $C_{t+1} \equiv C_t$, otherwise $k = argmax(\Delta d_i, \Delta d_j)$, add e to C_k, and update C_{t+1}.

2.3 Removing Link Algorithm

Deduction 2. *If the link* e *is the bridge link between* C_i *and* C_j*, when removing the link, the structures of* C_i *and* C_j *will be more apparent, and the whole community structure remains the same.*

Proof. When the removed link e is the bridge link between C_i and C_j, the link relation among nodes within the community does not change, but when the link between communities is removed, the connection of community will become looser, and the community structure in the network will be stronger and more obvious. As a result, the overall community structure will not change. □

When link e is removed, it can be divided into two cases:

1) The bridge link e is between C_i and C_j, $(i \neq j)$;

2) The link e is fully inside the community C_i. According to the deduction 2, for case (1), when removing the bridge link, the community will not change. For case (2), when the removed link e is an IL, set $S(e)$ as the neighboring link set of e, $\forall l \in S(e)$. If $C_k = argmax(D_{L,k}(l))$, assign the link l to the community C_k, where N is total number of link communities at present, and $1 \leq k \leq N$.

Removing link algorithm is as Algorithm 2.

Algorithm 2 *removing_link*

1: Input the removed link e and link community structure C_t in moment t.
2: Output the link community structure C_{t+1} in moment $t+1$.
3: If e is the community external link, then $C_{t+1} \equiv C_t$, otherwise $C_k = argmax(D_{L,k}(l))$, $l \in S(e)$, $k \in (1, N)$, add the link l into C_k, and update C_{t+1}.

To sum up, the Link Clustering based Community detection algorithm is described as Algorithm 3.

Algorithm 3 The LCC algorithm

1: Input $G_0 = (V_0, E_0)$, incremental information $\varepsilon = \{\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n\}$.
2: Output the community structure C_t of network G_t in moment t.
3: Find the link community structure G_0 at the initial moment.
4: Start from the initial moment, if $e \in adding\_link(L(u,v))$, then $adding\_link(C_t, L(u,v))$, otherwise $removing\_link(C_t, L(u,v))$.
5: Map the link community structure C_t into the node community, and obtain the nodes community structure at each moment.

3 Worm Inhibition Method in Social Network

3.1 Selection of Key Nodes

When inhibiting the social network worms, in addition to the community detection algorithm, the selection strategy

of key nodes can also affect the inhibition effect of social network worms. The formalized definitions of the key nodes are given here first.

In the given network $G(V, E)$, where $C = \{C_1, C_2, \cdots, C_k\}$ represent the community sets in the network. V_j^i represent the nodes in C_i, we use $|V_j^i|_{in}$ to indicate the connection numbers of V_j^i with other nodes in the community, also known as the internal node degree, and $|V_j^i|_{out}$ to signify the connection numbers of V_j^i with nodes in other communities, also known as the external node degree, use $|V_j^i|$ represent the connection numbers of V_j^i with other nodes. Obviously, $|V_j^i| = |V_j^i|_{in} + |V_j^i|_{out}$.

Definition 4. *(The maximum internal degree nodes.) In C_i, V_{maxin}^i is called as the maximum internal degree node, if and only if meeting the following formula:*

$$\forall V_j^i \in C_i, |V_j^i|_{in} \leq |V_{max}^i|_{in}. \quad (7)$$

Definition 5. *(The maximum external degree node.) In C_i, V_{maxout}^i is called as the maximum external degree node, if and only if meeting the following formula:*

$$\forall V_j^i \in C_i, |V_j^i|_{out} \leq |V_{max}^i|_{out}. \quad (8)$$

Definition 6. *(The maximum degree node.) In C_i, V_{max}^i is called as the maximum degree node, if and only if meeting the following formula:*

$$\forall V_j^i \in C_i, |V_j^i| \leq |V_{max}^i|. \quad (9)$$

In the real community, although the maximum internal degree node, the maximum external degree node and the maximum degree node could be the same node in most cases, there are also different situations, so three different node selection strategies are needed here.

Maxin strategy. Once the social network worms outbreak, select the maximum internal degree node V_{maxin} in the community. The nodes have the most links with other nodes in the community; this selection strategy is mainly based on local thoughts, because the nodes can immune other nodes in the community at full speed, thus inhibiting the spread of the worm in the community.

Maxout strategy. Once the social network worms outbreak, select the maximum external degree node V_{maxout} in the community. The nodes have the most links with other outside communities, which can not only prevent the worm spreading from other communities to this community, but also inhibit the spreading of the worm from this community.

Max strategy. Once the social network worms outbreak, select the maximum degree node V_{max} in the community. The nodes have the most neighbors, and the neighboring nodes can be either in the same community, or in other communities. The selection strategy is mainly based on the greedy thought, that is, immune the node with the strongest local transmission capacity first.

3.2 Inhibition Algorithm for Social Network Worms

According to three selection strategies of key nodes, we give the worm inhibition algorithm in social network after using LCC to obtain community structure of the social network.

Algorithm 4 The worm inhibition

```
Input: edge sets E, community structures C = {C_i}, and selection strategy P
Output: the key nodes set R
for C_i in C
    if (P is the maxin strategy)
        v ⟵ get_maxin(C_i, E)
    elseif (P is the maxout strategy)
        v ⟵ get_maxout(C_i, E)
    else
        v ⟵ get_max(C_i, E)
    if (v= NULL )
        R.add(v)
for v in R
    Issue immune notice to v
    Being immune, v spreads immune notice to its neighbors
return R
```

4 Experiments

4.1 Data Sets and Evaluation Indexes

In order to prove the validity of algorithm proposed in this paper, our method was tested in real-world web data sets and compared with other classical community detection algorithms, followed by verifying the validity of social network worms inhibition. 5 typical real-world web data sets were adopted for experimental analysis as illustrated in Table 1.

Table 1: Data sets

Data Sets	Nodes Amount	Edges Amount
Zachary Karate	34	78
Dolphin	62	159
Book US politics	105	441
Amercian college football	115	613
LiveJournal Social Networking Dataset	4847571	68993773

To evaluate the quality of community partition, the first evaluation criterion adopted was Q Modularity proposed by Newman and Girvan and the second was Normalized Mutual Information (NMI) proposed by Danon. Their definitions respectively as follows.

Standard definition of Q Modularity:

$$Q = \sum_i (e_{ii} - a_i^2) = Tre - \|e^2\| \tag{10}$$

$\|x\|$ means the sum of all elements in the x-matrix. First of all, a symmetric matrix of $k \times k$ was defined as $e = (e_{ij})$, in which e_{ij} refers the proportion of the lines connecting two nodes of different communities on the network in all lines. The two nodes are in the *ith* community and the *jth* community respectively. Suppose the sum of all the elements in the diagonals of matrix is $Tre = \sum_i (e_{ii})$ which refers the proportion of the lines connecting every node in some community on network in total of all lines. Then define the sum of every element in each line or each column as $a_i = \sum_j e_{ij}$, which refers the proportion of the lines connecting nodes in the *ith* community in all the lines. The upper limit of Q is $Q = 1$, thus the more closer to the value Q is, the more obvious community structure will be.

Standard definition of NMI:

$$NMI = \frac{-2\Sigma_{i,j} N_{ij} log(\frac{N_{ij} N}{N_{i.} N_{.j}})}{\Sigma_i N_{i.} log(\frac{N_{i.}}{N} + \Sigma_j N_{.j} log \frac{N_{.j}}{N})} \tag{11}$$

N_{ij} is the number of public nodes in clustering X_i and Y_j, N_i is the sum of the line *ith*, N_j is the sum of the column *jth*. NMI's value is between 0 and 1. When NMI=0, it indicates the two consequences completely inconsistent; When NMI=1, it indicates the two consequences completely consistent.

4.2 Community Detection Results

In order to verify the validity of LCC, in this section it was compared with algorithms like GN, LPA and BGLL [20, 21]. The comparative results between the average Q modularity acquired from 10 runs of LCC and from other 3 algorithms were given in Table 2.

Table 2: Comparison of Q modularity between our method and other algorithms

Data Sets	GN	LPA	BGLL	LCC
Zachary Karate	0.401	0.407	0.419	0.435
Dolphin	0.519	0.511	0.516	0.517
Book US politics	0.517	0.516	0.498	0.523
Amercian college football	0.599	0.598	0.602	0.611

Known from Table 2, as for Dolphin, the Q value acquired from LCC was slightly lower than that of GN while for other 3 data sets, the Q modularity value of LCC was the highest. Thus it was clear that LCC was able to perform community detection against large-scale complex networks.

The LCC and other algorithms such as GN, LPA, NFA, BGLL were acted on four known community structures (Zachary Karate, Dolphins, Book US politics, American college Football) and then the comparative results in NMI accuracy among such algorithms were given in Table 3.

Table 3: Comparison among different algorithms in term of NMI on real-world networks

Data Sets	GN	LPA	NFA	BGLL	LCC
Zachary Karate	0.58	0.84	0.69	0.59	1.0
Dolphin	0.55	0.59	0.57	0.52	0.63
Book US politics	0.56	0.51	0.52	0.57	0.66
Amercian college football	0.88	0.90	0.79	0.90	0.91

From Table 3 it was indicated that (1) The NMI value of optimum community partition by LCC for Zachary Karate was 1 and the community structure partitioned was shown in Figure 3. It could be indicated from Figure 3 that the community structure partitioned by LCC had a completely consistent structure with the real-world community structure. (2) For Dolphin, the NMI value of community partition by LCC was 0.63 and the community structure partitioned was shown in Figure 4. Seen from Figure 4, it was partitioned into 4 communities by LCC, in which the part represented by purple circle was corresponding with real-world community structure in Dolphin dataset, and by LCC the other part of real-world community structure was further partitioned into 3 tighter communities, which were represented as red square, green diamond, and blue triangle, respectively. (3) As for Book US polities and American college football sets, the NMI value acquired by LCC was higher than that by any other algorithm. Thus it was indicated that the community structure detected by LCC had a high accuracy.

4.3 Worm Inhibition Results

Related experiments were performed in order to verify the validity of algorithm proposed in this paper using for the inhibition of social network worms. Since the iterative method was used to analyze the propagation process of worms, here the end condition of iterative process was required for discussion. "Newly infected nodes" and "most nodes infected" were taken as two judgment conditions, by satisfying either of which the iteration might be terminated. In this experiment, the LiveJournal data set in Table 1 was selected to conduct four experiments. Then no inhibition means were adopted and the worm inhibition effects under 3 key node selection strategies stated in

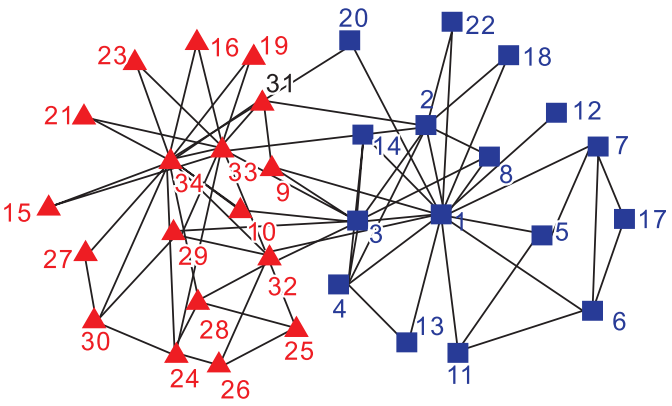


Figure 3: Detection results of LCC against Zachary Karate

Section 3.1 were used for comparison. The experimental results were shown in Figure 5.

During the experiments, the worms inhibition course started when the worm infection rate was over 2%. Seen from Figure 5, when adopting key node inhibition strategy, it was indicated that during the worms' propagation process, after 600 time units, the infected user amount basically remained at about 30%, showing that the inhibition scheme of key nodes in social network had better defense against worms. Meanwhile, seen from the inhibition effect under three key node selection strategies listed in Figure 5, the max strategy had the best worm inhibition effect.

The worm inhibition effect of maxout strategy and other worm inhibition algorithms were compared. In consideration of larger community amount and key nodes amount in large-scale network, in this experiment different proportions of immunization nodes were selected for such comparison. The experiment results were shown in Figure 6.

Seen from Figure 5, only a certain proportion of key nodes were required for immunization to acquire a better worm inhibition effect. For instance, the adoption of max strategy only required about 40% of nodes to guarantee the final proportion of nodes infected were not more than 20%. The final proportion of infected nodes would not be more than 25% even when only 20% of key nodes were selected. Therefore, the validity of algorithm in this paper was proved.

According to the experiments above, we can find that the strategy used in the article has less suppression effective than Livshit's method, the main cause is that Q Modularity value of community structure found by LCC is slightly lower, which indicates it is very important to improve accuracy for Community Detection, in order to get better worm inhibition results, and this part will be our key research in the next step.

5 Conclusions

At first, adopting improved Link Partition Density Function, this paper makes Community Detection. Then we propose three different strategies to choose key nodes, and give these key nodes immunization to get better worm inhibition results. Finally, to verify the validity of mentioned algorithm, we perform it on a lot of real network data sets.

Acknowledgment

This study was supported by the National Science Foundation of China under Contracts 51477001. The authors would like to thank the Associate Editor and the Reviewers for their valuable comments and suggestions.

References

[1] Y. Y. Ahn, J. P. Bagrow, S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[2] P. Brodka, T. Filipowski, P. Kazienko, "An introduction to community detection in multi-layered social network," *Informatio Systems, E-learning, and Knowledge Management Research*, pp. 185–190, 2013.

[3] Y. Cao, V. Yegneswaran, P. Porras, Y. Chen, "PathCutter: Severing the self-propagation path of XSS JavaScript worms in web social networks," in *Proceedings of the 19th Network and Distributed System Security Symposium (NDSS'12)*, San Diego, USA, 2012.

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

[5] A. Clauset, "Finding local community structure in networks," *Physical Review E*, vol. 72, no. 2, 026132, 2005.

[6] Z. Dong, P. Yi, "A community detection algorithm for dynamic networks using link clustering," *Journal of Xian Jiaotong University*, vol. 48, no. 8, pp. 73–79, 2014.

[7] M. R. Faghani, H. Sandi, "Social networks' XSS worms," in *Proceedings of the International Conference on Computational Science and Engineering*, pp. 1137–1141, 2009.

[8] M. Girvan, M. E. J. Newman, "Community structure in social and biological networks," *National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[9] L. He, D. G. Feng, P. R. Su, et al., "Parallel community detection based worm containment in online social network," *Chinese Journal of Computers*, vol. 38, no. 4, pp. 846–857, 2015.

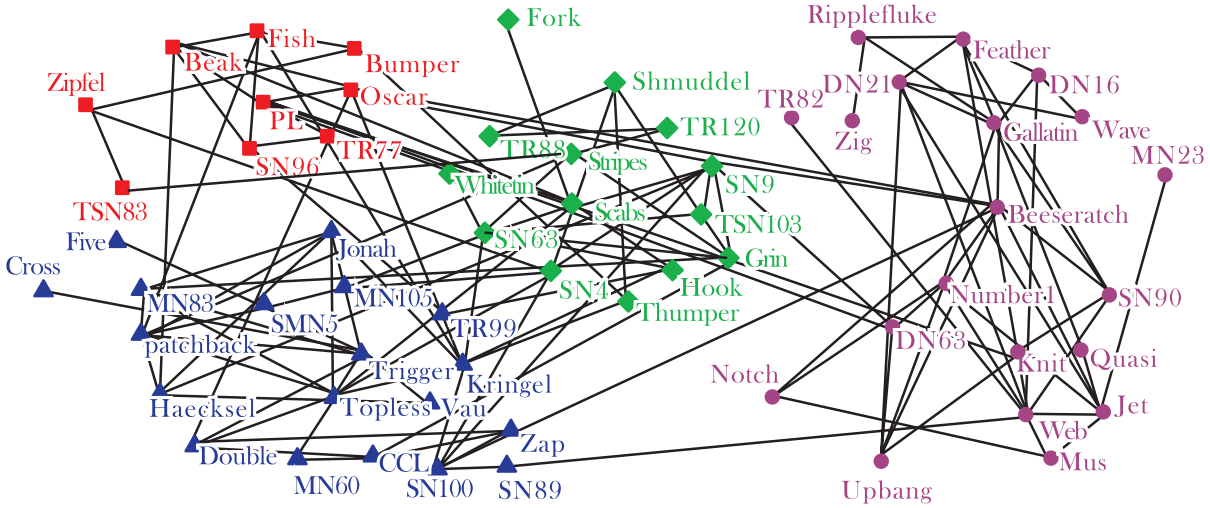


Figure 4: Detection results of LCC against Dolphin

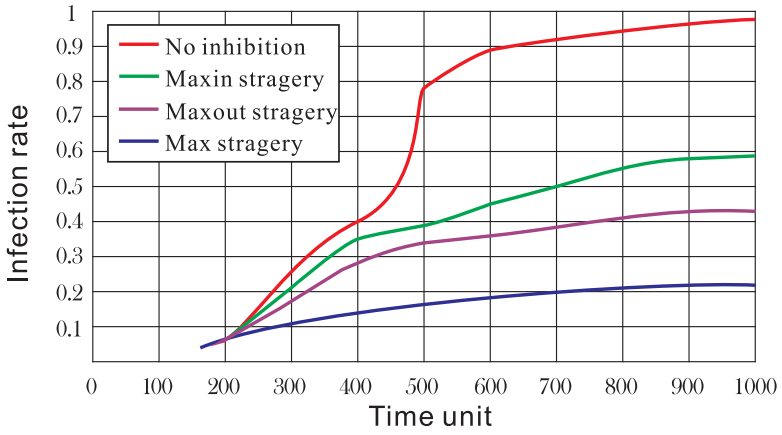


Figure 5: The results of worms inhibition by different strategies

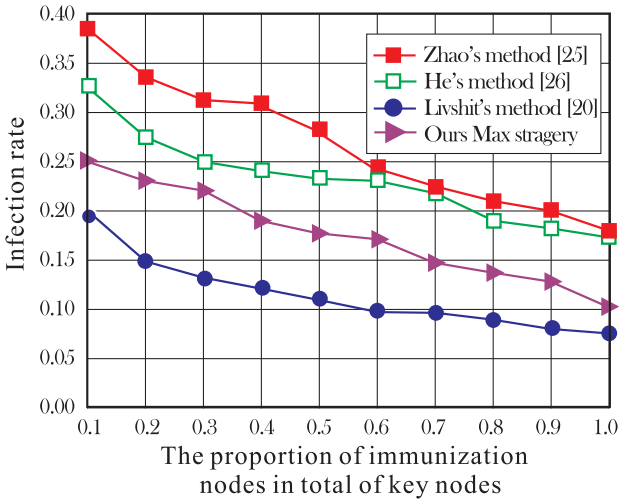


Figure 6: Comparison results of worms inhibition between our method and other algorithms

[10] M. S. Hwang, C. C. Lee, S. K. Chong, J. W. Lo, "A key management for wireless communications," *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 8, pp. 2045–2056, 2008.

[11] 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.

[12] M. S. Hwang, C. C. Yang, S. F. Tzeng, "Improved digital signature scheme based on factoring and discrete logarithms," *Journal of Discrete Mathematical Sciences & Cryptography*, vol. 5, no. 2, pp. 151–155, Aug. 2002.

[13] D. Kuang, C. Ding, H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proceedings of 2012 SIAM International Conference on Data Mining*, pp. 106–117, 2012.

[14] M. Kumar, K. Dutta, I. Chopra, "Impact of wormhole attack on data aggregation in hieracrchical WSN," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 70–77, 2014.

[15] W. T. Li, T. H. Feng, M. S. Hwang, "An intrusion detection technique based on continuous binary communication channels," *International Journal of Network Security*, vol. 16, no. 5, pp. 323–330, 2014.

[16] B. Livshit, W. Cui, "Spectator: Detection and containment of JavaScript worms," in *Proceedings of the USENIX Annual Technical Conference on Annual Technica1 Conference*, pp. 335–348, 2008.

[17] Z. G. Luo, X. Z. Jiang, "New progress on community detection in complex networks," *Journal of National University of Defense Technology)*, vol. 33, no. 1, pp. 47–52, 2011.

[18] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *American Physical Society*, vol. 69, no. 6, pp. 187–206, 2004.

[19] M. E. J. Newman, M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, 2004.

[20] N. P. Nguyen, T. N. Dinh, S. Tokala, "Overlapping communities in dynamic networks: Their detection and mobile applications," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, pp. 85–96, 2011.

[21] N. P. Nguyen, Y. Xuan, M. T. Thai, "A novel method for worm containment on dynamic social networks," in *Proceedings of the 2010 Military Communications Conference (MILCOM'10)*, pp. 2810–2815, 2010.

[22] G. Palla, I. Derenyi, I. Farkas, et a1., "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[23] U. Raghavan, R. Albert, S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, 036106-1, 2007.

[24] M. Shiga, I. Takigawa, H. Mamitsuka, "A spectral clustering approach to optimally combining numerical vectors with a modular network," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 647–656, 2007.

[25] J. Singh, "Cloud based technique for Blog search optimization," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 32–39, 2016.

[26] F. Sun, L. Xu, Z. Su, "Client-side detection of XSS worms by monitoring payload propagation," in *Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS'09)*, pp. 539–554, 2009.

[27] X. Sun, Y. Y. Liu, J. Q. Zhu, et al., "Research on simulation and modeling of social network worm propagation," *Chinese Journal of Computers*, vol. 34, no. 7, pp. 1252–1260, 2011.

[28] A. L. Traud, P. J. Mucha, M. A. Porter, "Social structure of Facebook networks," *Physical A: Statistical Mechanics and Its Applications*, vol. 391, no. 16, pp. 4165–4180, 2012.

[29] L. Wang, X. Q. Cheng, "Dynamic community in online social networks," *Chinese Journal of Computers*, vol. 38, no. 2, pp. 219–237, 2015.

[30] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[31] W. Xu, F. Zhang, S. Zhu, "Toward worm detection in online social networks," in *Proceedings of 26th Annual Computer Security Applications Conference*, PP.11–20, 2010.

[32] S. Yang, H. Jin, X. Liao, et al., "Modeling modern social-network-based epidemics: A case study of rose," in *International Conference on Autonomic and Trusted Computing*, PP.302–315, 2008.

[33] X. Z. Zhang, Y. Y. Pu, L. Yang, B. Wang, "Community discovery of large-scale web service network," *Journal of Chinese Computer Systems*, vol. 36, no. 5, pp. 1017–1020, 2015.

[34] Y. Zhao, P. K. Yi, "A dynamic worm propagation model based on social network," *Computer Engineering and Science*, vol. 35, no. 12, pp. 34–38, 2013.

[35] C. C. Zhou, W. Gong, D. Towsley, "Code red worm propagation modeling and analysis," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 138–147, 2002.

[36] Z. Zhu, G. Cao, S. Zhu, et a1., "A social network based patching scheme for worm containment in cellular networks," in *Proceedings of the IEEE INFOCOM*, pp. 1476–1484, 2009.

Biography

Yibing Wang is currently a lecturer at Anhui University. She chaired or participated the national, provincial and

municipal scientific research projects over 20 itemsand published more than 10 journal papers. what is moreas the first inventor, she obtained 2 items national patent. Her research interests include machine learning, pattern recognition and network security.

Jie Fang received her Doctor Degree in Computer Science from Chinese Academy of Sciences. She has published more than 30 papers in international conferences and journals (SCI or EI journals). She is currently a faculty member in the college of Computer and Information Engineering at Anhui University. Her research interests include network security and Artificial intelligence.

Fuhu Wu is currently a Doctoral student at Anhui University. Her research interests include network protocols and security, enterprise systems, etc.

An Improved Biometric Based Authentication Scheme with User Anonymity Using Elliptic Curve Cryptosystem

Lidong Han, Qi Xie, and Wenhao Liu

(Corresponding author: Lidong Han)

Key Laboratory of Cryptography and Network Security, Hangzhou Normal University
No.58, Haishu Rd, Yuhang District, Hangzhou 311121, China
(Email: ldhan@hznu.edu.cn)

(Received Oct. 27, 2015; revised and accepted Feb. 21 & Apr. 6, 2016)

Abstract

Telecare medical information systems (TMIS) provides convenient health care services for patients in order to save the patients' time and expense. The protection of user's privacy and data security is significant over public communication. Recently, Lu et al. presented a three-factor based authentication protocol using elliptic curve cryptography. In this paper, we analyze the security of Lu et al.'s scheme. We demonstrate that Lu et al's scheme can't protect user anonymity and insecure against impersonation attack. To remedy the mentioned security weakness, we propose a new authentication scheme to improve on Lu et al.'s scheme. In comparison with recent schemes, our scheme can provide stronger security and more efficiency in implementation.

Keywords: Authentication, password, user anonymity

1 Introduction

With the rapid development of information and network technologies, connected health care can be applied in many fields, such as telecare medicine information system (TMIS). TMIS provides a convenient communication via public channels between patients (doctors) at home and medical servers. The merit of TMIS is that it enables patients accessing and updating patient's medical information in TIMS server, and it provides health-care services directly into the patient at home using internet, which can save patients much time and expenses. In order to protect patients' privacy and security, it is very important to achieve secure mutual authentication between patients and the medical server before diagnosis. which result in data security and user's privacy issues. The sensitive healthcare information should be protected and user's personal private information should not be leaked to the malicious users or adversaries. A secure and efficient authentication and key agreement scheme can provide various aspects of security for health data and user privacy. Recently, many authentication schemes using smart card have been presented to ensure secure and authorized access of data [4, 8, 9, 13, 14, 21, 22, 24, 27].

In 2009, Wu et al. [24] proposed an efficient authentication scheme using smart card for TMIS with pre-computation. However, He et al. [8] found that Wu et al.'s scheme is not secure against impersonation attacks and insider attacks. To address these problems, He et al. gave an improved authentication scheme. Later, Wei et al. [22] showed both Wu et al.'scheme and He et al.'s scheme are not resistant to off-line password guessing attacks and cannot achieve two-factor authentication. They also presented an improved authentication scheme for TMIS and claimed that the improved scheme can achieve two-factor authentication. Unfortunately, Zhu [29] pointed out that Wei et al.'s scheme is also vulnerable to off-line password guessing attack using stolen smart card. Zhu designed a new RSA based authentication scheme and claimed that the new scheme is secure against various attacks.

However, in all password based remote user authentication schemes mentioned above, an adversary can obtain user's identity since the identity was transmitted in plaintext in authentication process. In 2004, Das et al. [7] proposed a dynamic ID-based password authentication scheme to solve this security weakness. Since then, many dynamic ID-based authentication schemes have been designed. Chen et al. [6] showed Khan et al.'s scheme [11] can not protect the user's anonymity and presented a dynamic ID-based password authentication scheme. However, Xie et al. [25] showed that Chen et al.'s scheme does not provide user privacy protection and perfect forward secrecy, and proposed an improved scheme. Until now, many researchers have analyzed the security of password-based authentication schemes. Also other researchers proposed their authentication and key agreement schemes.

All above mentioned authentication schemes are based

on two factors password and smart cards [1, 19]. Lately, researchers focused on three factor based authentication and key agreement scheme employing biometric, which has stronger security than two factor based schemes [28]. In 2013, Tan [20] proposed a biometric based remote user authentication scheme for telecare medical information system to achieve mutual authentication and session key establishment. Awasthi and Srivastava [3] presented an efficient biometric based authentication scheme, which only uses the Xor operation and hash function to lower computational cost for smart cards. Tan showed that Awasthi-Srivastava's three-factor scheme is vulnerable to the reflection attacks and it fails to provide user anonymity and three-factor security. Recently, Lu et al. [15] put forward the security weakness of of Arshad et al.'s scheme [2], and proposed an biometric-based authentication schemes for TMIS using elliptic curve cryptosystem.

In this paper, we demonstrate that Lu et al.'s scheme fails to protect patient's anonymity. Additionally, we show that a legal user can impersonate any user of the system to communicate with the server, and disguise as a legitimate server to deceive a user. Furthermore, we put forward an improved biometric based authentication scheme to deal with the weakness of Lu et al.'s scheme. Our proposed scheme also employs lower computational operations such as ECC and hash function to lower its computational cost.

The remainder of this paper is organized as follows: In first section, we introduce some notations and definitions used in this paper. Section 3 will review the biometric -based authentication scheme by Lu et al. Section 4 analyzes the security problems of Lu et al.'s protocol. We present a new biometric -based authentication scheme based on ECC in Section 5. Section 6 will elaborate the security and efficiency of our new scheme briefly, and gives a comparison of several previous biometric based authentication schemes. And A comparison with some previous authentication schemes in the aspect of security and efficiency is given in Section 7. Finally, we give a conclusion in the last section.

2 Preliminaries

This section lists the notations and definitions used in this paper, and briefly reviews the basic concepts of biohashing, ECC cryptosystem along with some hardness problems are introduced.

2.1 Notations

Table 1 lists the notations that will be used in this paper.

In Table 1, one-way hash function $h(\cdot)$ maps a string of arbitrary length to a string of fixed length which is called hashed value. It can be represented as $h : \{0,1\}^* \rightarrow \{0,1\}^n$. Such hash function is easy to compute on every input, but hard to invert given the image of a random input.

Table 1: Notations

Symbol	Description
U	the user/patient
S	The telecare server
PW, ID, B	Password, Identity, Biometric of user
x	Private key of server
$h_1(\cdot)$	Hash function h_1: $\{0,1\}^* \rightarrow \{0,1\}^l$
$h_2(\cdot)$	Hash function h_2: $\{0,1\}^* \rightarrow Z_p^*$
$H(\cdot)$	Biometric Hash function
SK	Session key between U and S
$\|\|$	String concatenation operation
$\oplus$	Exclusive-or operation
$E_x(\cdot)$	Symmetric encryption with x
$D_x(\cdot)$	Symmetric decryption with x

It is noted that, when the elliptic curve point $P = (x, y)$ as input in hash operation and $\oplus$ operation, P is represented as a value by $x||y$.

2.2 Bio-hashing

The biometrics is of great importance to provide genuine user authentication in any authentication scheme. In general, imprint biometric characteristics (face, fingerprint, palmprint) may not be exactly same at each time. Therefore, high false rejection of valid users resulting low false acceptation, is often occurs in the verification of biometric systems. In order to resolve the high false rejection rate, Jin et al. [10] proposed a two-factor authenticator on iterated inner products between tokenised pseudo-random number and the user specific fingerprint features, which produces a set of user specific compact code that coined as Bio-Hashing. Later, Lumini and Nanni [16] proposed the improvement of Bio-Hashing. As specified in [5], Bio-Hashing maps a user/patients biometric feature onto user specific random vectors in order to generate a code, called biocode and then discretizes the projection coefficients into zero and one. Biocode is also as secure as a hashed password.

3 Review of Lu et al.'s Scheme

In 2015, Lu et al. proposed an biometric-based authentication and key agreement scheme based on elliptic curve cryptosystem [15], which is based on Arshad et al.'s scheme [2]. It consists of four phases: registration, login, authentication, password change. In this section, we will briefly review these phases of Lu et al.'s scheme.

Registration phase.

In this phase, a new user U_i registers to the server S and achieves the personalized smart card via the following steps:

- The user U_i selects his identity ID_i, password PW_i and inputs his biometric B_i. He computes

$MP_i = PW \oplus H(B_i)$, and sends $\{ID_i, MP_i\}$ to the server S through a secure channel.

- Upon receiving the registration request, S calculates $AID_i = ID_i \oplus h_2(x)$ using the private key x, $V_i = h_1(ID_i||MP_i)$, and stores $\{AID_i, V_i, h_1(\cdot), h_2(\cdot), H(\cdot)\}$ into a smart card SC_i. S issues SC_i to the patient U_i.

Login and Authentication phase.

The user U_i and the server S execute the following steps in order to achieve authentication and session key agreement.

- U_i first inserts the smart card SC_i, and inputs his identity ID_i, password PW_i and biometric B_i. Then, SC_i checks whether $h_1(ID_i||PW_i \oplus H(B_i)) = V_i$ holds or not. If it holds, go to the next step.
- The smart card SC_i generates a random number d_u, and computes $K = h_1(ID_i||ID_i \oplus AID_i)$, $M_1 = K \oplus d_uP$ and $M_2 = h_1(ID_i||d_uP||T_1)$. SC_i sends the login request $\{M_1, M_2, AID_i, T_1\}$ to S.
- After receiving $\{M_1, M_2, AID_i, T_1\}$, S first examines whether $|T_c - T_1| < \Delta T$, where T_c is the current time stamp. If true, S computes $AID_i \oplus h_2(x)$ using his private key x to extract ID_i, then he calculates $d_uP = h_1(ID_i||h_2(x)) \oplus M_1$ and verifies whether $M_2 = h_1(ID_i||d_uP||T_1)$ holds. If correct, S chooses a number d_s randomly, and computes $M_3 = K \oplus d_sP$, $SK = d_s(d_uP)$, $M_4 = h_1(K||d_uP||SK||T_2)$, where T_2 is the current time. Then, S transmits $\{M_3, M_4, T_2\}$ to U_i.
- Upon receiving $\{M_3, M_4, T_2\}$, SC_i checks the validity of T_2. Then, U extracts d_sP from computing $M_3 \oplus K$, and computes $SK = d_u(d_sP)$, $M_4' = h_1(K||d_uP||SK||T_2)$. Then, checks whether $M_4' = M_4$ holds. If correct, the smart card SC_i computes $M_5 = h_1(K||d_sP||SK||T_3)$ and then sends the message $\{M_5, T_3\}$ to S.
- S checks the freshness of T_3, and then verifies $h_1(K||d_sP||SK||T_3) \stackrel{?}{=} M_5$. If both are correct, S authenticates U_i and accepts SK as the session key.

Password change phase.

If the patient U_i wants to change his old password PW_i, U_i inserts the smart card into the device and inputs the ID_i, PW_i, and B_i. Then SC_i verifies $h_1(ID_i||PW_i \oplus H(B_i)) \stackrel{?}{=} V_i$. If holds, U_i keys a new password PW_i^{new}, SC_i computes $V_i^{new} = h_1(ID_i||PW_i^{new} \oplus H(B_i))$. Finally, it replaces V_i by V_i^{new}.

4 Security Weakness of Lu et al.'s Scheme

In this section, we demonstrate that Lu et al.'s scheme fails to achieve their claimed security goals. In the attack model, it can be assumed that an adversary could get the values which are stored into a user U_i's smart card by monitoring the power consumption [12, 17]. Also an adversary has the ability of controlling over the communication totally. That means that he can extract and modify the transmitting messages between U_i and S. In the following, we will analyze the security of Lu et al.'s scheme in detail.

Linkability.

The linkability is that an adversary can determinate whether two login messages are sent by the same patient. Since the login request message $m_1 = \{AID_i, M_1, M_2, T_1\}$ contains the fixed value $AID_i = ID_i \oplus h_2(x)$ where ID_i is the user's identity and $h_2(x)$ is a hash function with input x. Therefore, when an adversary intercepts two login messages m_1 and m_1', he only decides whether the first part of m_1 and m_1' are equal. If it's correct, we determine that two login messages must be from the same patient.

Fails to protect user anonymity.

In this subsection, we will show that Lu et al.'s scheme does not protect patient's anonymity for the insider users.

In Lu et al.'s scheme, the patient's identity is obscured by the form $AID_i = ID_i \oplus h_2(x)$, which is part of the transmitted message by public channel in login phase. For outside attackers, it's not efficient to retrieve the patient's identity without knowledge of the secret value $h_2(x)$. However, for a legal but malicious patient U_j, he can extracts $h_2(x)$ using his own identity ID_j and the value AID_j stored in smart card. Then, U_j can easily compute any other patient's identity by computing $ID = AID \oplus h(x)$ where AID can be intercepted in initiating login phase.

Server impersonation attack.

This subsection describes that a legitimate user in Lu et al.'s scheme can impersonate as a legal sever. Denote U_j be a legal patient, who wants to simulate as legal TMIS server. U_j will perform the following steps to impersonate as a legal server.

1) U_j extracts the secret information $\{V_j, AID_j, h_1(\cdot), h_2(\cdot), H(\cdot)\}$ stored into his smart card by monitoring the power consumption or analyzing the leaked information. U_j then computes $AID_j \oplus ID_j$ using his password PW_j to obtain $h_2(x)$.
2) When a patient U_i executes the login and authentication process and sends $\{M_1, M_2, AID_i, T_1\}$ to S, U_j intercepts the login message.

3) U_j computes $AID_i \oplus h_2(x)$ using the value $h_2(x)$ to retrieve the identity of U_i. Then U_j chooses a random number $d'_s \in Z_p^*$, and computes $M'_3 = h_1(ID_i||h_2(x)) \oplus d'_s P$, $SK' = d'_s(d_u P)$, $M'_4 = h_1(K||d_u P||SK'||T_2)$, where T_2 is the current time stamp. U_j sends $\{M'_3, M'_4, T_2\}$ to U_i
4) U_i check the validity of T_2. Then computes $K \oplus M'_3 = d'_s P$, $SK = d_u(d'_s P)$,$M_4^* = h_1(K||d_u P||SK||T_2) \stackrel{?}{=} M'_4$. U_i accepts the session key SK since the verification is correct and regards U_j as a legitimate sever.

Therefore, a legal patient can simulate as a legitimate sever to all other users.

User impersonation attack.

Lu et al. claimed their scheme could withstand various attack. Now, we demonstrate that a legal but malicious patient U_j can impersonate a patient to the server. The details of impersonation attack are presented in the following.

1) U_j can get $h_2(x)$ by computing $AID_j \oplus ID_j$ as similar as step 1 in server impersonation, where AID_j is retrieved in his his smart card.
2) When another patient U_i initiates the login process and transmits the request $\{M_1, M_2, AID_i, T_1\}$ to S. U_j extracts AID_i from the request message and computes $ID_i = AID_i \oplus h_2(x)$. The adversary U_j terminates this session.
3) U_j selects a random nonce $d'_u \in Z_p^*$, current time stamp T_1, calculates $K = h_1(ID_i||h_2(x))$, $M'_1 = K \oplus d'_u P$ and $M'_2 = h_1(ID_i||T_1||d'_u P)$. Then U_j sends the login message $\{M'_1, M'_2, AID_i, T_1\}$ as the login message of U_i to S.
4) After receiving the login message, S verifies whether $|T_1 - T_s| \leq \Delta$. If not true, S aborts the session. Otherwise, S computes $ID_i = AID_i \oplus h_2(x)$. Then S chooses a random number $d_s \in Z_p^*$, and computes $M_3 = h_1(ID_i||h_2(x)) \oplus d_s P$, $SK = d_s(d_u P)$, $M_4 = h_1(K|T_2||SK'||d_u P|)$, where T_2 is the current time stamp. U_j sends $\{M_3, M_4, T_2\}$ to U_i.
5) U_j computes $K \oplus M_3 = d_s P$, $SK = d_u(d_s P)$. Then U_j checks whether $M_4^* = h_1(K||d_u P||SK||T_2) \stackrel{?}{=} M'_4$. U_j computes $M_5 = h_1(K||d_s P||SK||T_3)$ and then sends the message $\{M_3, T_3\}$ to S.
6) S checks the freshness of T_3 from the received message, and verifies $M'_5 = h_1(K||d_s P||SK||T_3) \stackrel{?}{=} M_5$. S authenticates U_j as U_i and accepts SK as the session key.

Hence, a legal patient can impersonate himself as any other patients to sever S. Therefore, it is shown that Lu et al.'s scheme is vulnerable to user impersonation attack.

5 Proposed Scheme

From previous section, it is observed that the important weakness of Lu et al.'s scheme is the form of AID_i, which leaks some information of server's private key. That means any legal user can obtain $h_2(x)$ that can be used in attacking Lu et al.'s scheme. This section proposes an improved three-factor authentication scheme based on Lu et al.'s scheme. In the proposed scheme, in order to resist the impersonation attack, we employ a hash function with inputs the patient's identity and private key, which is related to the communicating patient. The four phases of our proposed scheme are described as follows.

Registration phase.

A new user U_i chooses identity and password and then registers his identity to the server S. Server registers the user and provides the valid smart card in return.

- The patient U_i generates a random number r, and chooses his identity ID_i, password PW_i and his biometric B_i. He computes $MP_i = PW_i \oplus H(B_i) \oplus r$, and sends $\{ID_i, MP_i\}$ to the server S through a secure channel.
- The sever S computes $AID_i = h(ID_i||x)$, $K_i = h(AID_i)$, $V_i = AID_i \oplus MP_i$. Then, S generates a number a randomly and computes $CID_i = E_x(ID_i||a)$. The server issues a smart-card SC_i to the patient U_i which is stored by $\{K_i, V_i, CID_i, h(\cdot), H(\cdot)\}$.
- Upon receiving the smart card, U_i computes $R_i = r \oplus h(ID_i||PW_i||H(B_i))$, and stores R_i into SC_i.

Login and authentication phase.

A legal user with valid smart card can establish a secure and authorized session with the server. In this phase, user and server first authenticate each other and then agree on a session key that can be used for the secure transmission of data.

- U_i first inserts SC_i into the card reader, and enters his identity ID_i, password PW_i and biometric B_i. Then, smart card SC_i computes $r = R_i \oplus h(ID_i||PW_i||H(B))$, $MP_i = PW_i \oplus H(B_i) \oplus r$, and $AID_i = V_i \oplus MP_i$. The card checks whether $h(AID_i) \stackrel{?}{=} K_i$. If holds, go to next step.
- SC_i generates a random nonce $d_u \in Z_p$, and computes $d_u P$, $M_1 = AID_i \oplus D$ and $M_2 = h(AID_i||d_u P||T_1)$. SC_i transmits $\{M_1, M_2, CID_i, T_1\}$ to the server.
- After receiving the login request $\{M_1, M_2, CID_i, T_1\}$, S first checks the freshness of T_1 by verifing whether $|T_c - T_1| < \Delta T$, where T_c is the current time. If true, S retrieves ID_i by decrypting CID_i, and

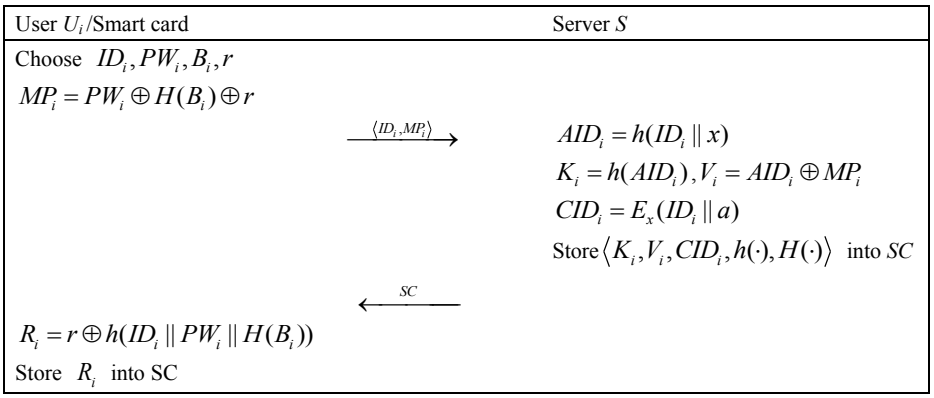


Figure 1: Registration phase of proposed scheme

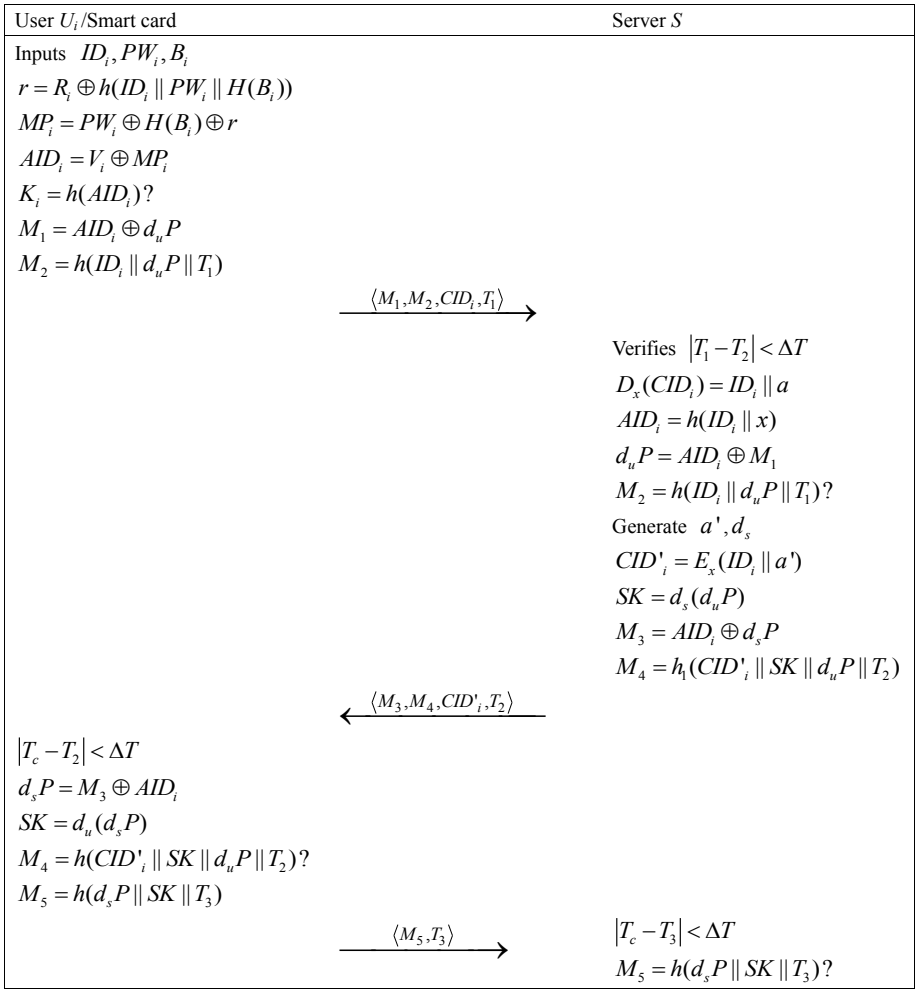


Figure 2: Login and authentication phase of proposed scheme

computes $AID_i = h(ID_i||x)$. Then he calculates $d_uP = AID_i \oplus M_1$ and verifies whether $M_2 = h(AID_i||d_uP||T_1)$ holds. If correct, The sever generates $d_s \in Z_p$ and a' randomly, and computes $E = d_sP$, $CID_i' = E_x(ID_i, a')$, $M_3 = AID_i \oplus E$, $SK = h(AID_i||d_s(d_uP)||CID_i)$, $M_4 = h(CID_i'||SK||d_sP||T_2)$, where T_2 is the current time. Then, S sends $\{M_3, M_4, CID_i', T_2\}$ to U.

- Upon receiving $\{M_3, M_4, CID_i', T_2\}$, SC_i checks the freshness of T_2. Then, U extracts d_sP from computing $M_3 \oplus AID_i$, and computes $SK = h(AID_i||d_u(d_sP)||CID_i)$, $M_4' = h(CID_i'||SK||d_sP||T_2)$. Then, check whether $M_4' = M_4$ holds. If correct, SC_i replaces CID_i with CID_i', and computes $M_5 = h(d_sP||SK||T_3)$ and then sends the message $\{M_5, T_3\}$ to S.
- S checks the validity of T_3, and verifies $h(d_sP||SK||T_3) \stackrel{?}{=} M_5$. If both are correct, S authenticates U and accepts SK as the session key.

Password change phase.

A valid user with smart card can change the password of the smart card as follows:

- U_i inserts the smart card into the device and inputs the ID_i, PW_i and B_i.
- SC_i computes $r = R_i \oplus h(ID_i||PW_i||H(B_i))$, $MP_i = PW_i \oplus H(B_i) \oplus r$, $AID_i = V_i \oplus MP_i$ and checks $h(AID_i) \stackrel{?}{=} K_i$. If it holds, U_i inputs a new password PW_i^{new}, biometric B_i^{new} and a new random number r^{new}.
- SC_i computes $MP_i^{new} = PW_i^{new} \oplus H(B_i^{new}) \oplus r^{new}$, $V_i^{new} = AID_i \oplus MP_i^{new}$, $R_i^{new} = r^{new} \oplus h(ID_i||PW_i^{new}||H(B_i^{new}))$. Finally, it replaces R_i, V_i by R_i^{new}, V_i^{new} respectively.

6 Security Analysis

In this section, we demonstrate that our scheme can resist a number of possible attack types.

User anonymity.

Suppose an adversary eavesdrops the login request $\{M_1, M_2, CID_i, T_1\}$ during the login phase, and the authentication message $\{M_3, M_4, CID_i', T_2\}$ during the authentication and key agreement phase, where $M_1 = AID_i \oplus D, M_2 = h(AID_i||D||T_1)$, $CID_i = E_x(ID_i||a), M_3 = AID_i \oplus d_sP, M_4 = h(CID_i'||SK||E||T_2)$, $CID_i' = E_x(ID_i||a')$. Note that CID_i, CID_i' are encrypted ciphertexts of ID_i by x, and nobody other than the server has the private key. And remaining parts M_1, M_2, M_3, M_4 are the form of output hash function of with a user's identity. Due to one-way property of collision-resistant hash function, it is hard to compute the ID from these eavesdropped message. Hence, our proposed scheme provide patient's anonymity.

Replay attack.

In proposed scheme, the current timestamp is included in the login message $\{M_1, M_2, CID_i, T_1\}$ and the response message $\{M_3, M_4, CID_i', T_2\}$, where $M_1 = AID_i \oplus D$, $M_2 = h(AID_i||D||T_1)$, $M_3 = AID_i \oplus E$, $M_4 = h(CID_i'||SK||E||T_2)$. If the attacker wants to send the login message or authentication message only altering time stamp, the patient and the server could detect the replay attack by checking the validity of T_1 and T_2 respectively. If the attacker generates M_1, M_2 or M_3, M_4 by himself, M_1 and M_3 are computed from value AID_i, which needs the knowledge of PW_i, B_i(or x), and M_2 and M_4 are protected by a hash function. Hence, our scheme can present replay attack.

User impersonation attack.

If the adversary wants to impersonate the legitimate user to the server, he has to generate a valid login request message $\{M_1, M_2, CID_i, T_1\}$, where $M_1 = AID_i \oplus D, M_2 = h(AID_i||D||T_1)$,. It is clear that the adversary can generate a random element in Z_p^* and guess the patient's identity to compute M_2, and the third part CID_i can be retrieved from the eavesdropped message $\{M_3, M_4, CID_i', T_2\}$ in authentication phase. But, it is very difficult to calculate the valid number M_1 without the knowledge of AID_i. AID_i can be computed by the pair (ID_i, PW_i, B_i) and V_i. Then the server could detect the attack by checking the correctness of M_1 and M_2. Therefore, the proposed scheme can prevent the user impersonation attack.

Server spoofing attack.

To masquerade as the legal server, an attacker aims to generate the forged response message $\{M_3, M_4, CID_i, T_1\}$, where $M_3 = AID_i \oplus D$, and $M_4 = h(CID_i'||SK||E||T_2)$. It is easy to obtain the part CID_i' by monitoring the communicating channel. However, M_3 cannot made without the value AID_i, and M_4 is a one-way hash function with private parameters SK. Computing both valid M_3, M_4 are hard for the attacker without server's secret key x. Therefore, the proposed scheme can withstand the server impersonation attack.

Off-line password guessing attack.

For this attack model, an adversary is assumed that he is able to extract all the secret information stored in the memory of smart card by power analysis attack. Thus, he obtain the parameters $\{K_i, V_i, CID_i, R_i\}$, where only V_i and R_i are related with the patient's password. In the following, we show that the adversary can not extract successfully the patient's password by off line password guessing attack.

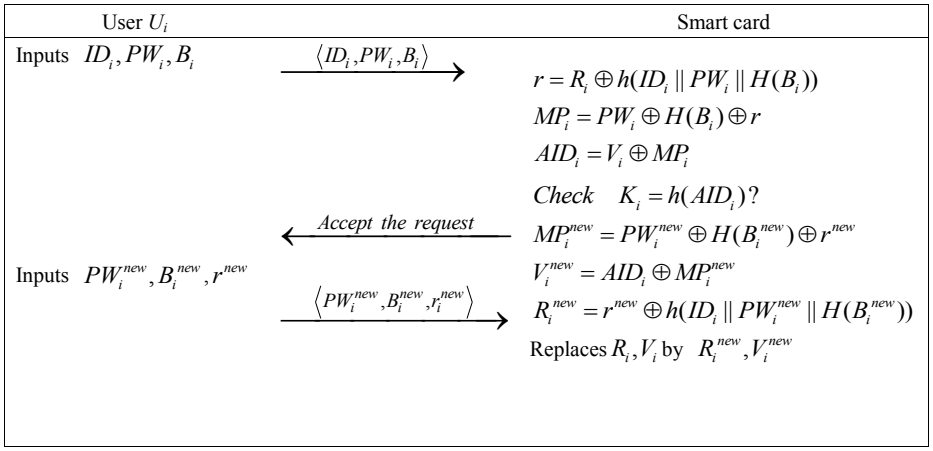


Figure 3: Password change phase of proposed scheme

- We know that $V_i = AID_i \oplus PW_i \oplus H(B_i) \oplus r$, where AID_i is computed by ID_i and server's private key x, r is a random number. Given ID_i, PW_i, it is computationally hard to get x, user's biometric B_i and r. Hence, the attacker can not check whether the equation of V_i holds by guessing a patient's identity and password.
- For R_i, we have $R_i = r \oplus h(ID_i||PW_i||H(B_i))$. In order to utilize the above equality, the attacker not only needs the parameters ID_i, PW_i but also have to know both private biometric B_i and r.
- Combining V_i and R_i, r can be represented by $R_i \oplus h(ID_i||PW_i||H(B_i))$. That is to say, $V_i = h(ID_i||x) \oplus PW_i \oplus H(B_i) \oplus R_i \oplus h(ID_i||PW_i||H(B_i))$. It only works efficiently to know (ID_i, PW_i, B_i, x) for the adversary by executing off-line password guessing attack.

From above analysis, the adversary has to check the validity of pair $(ID_i,\ PW_i, B_i, x)$ by combining V_i and R_i. It is infeasible for doing an exhausted search for all possible (ID_i, PW_i, B_i, x) pairs. Hence, the proposed scheme could resist off-line password guessing attack.

Perfect forward secrecy.
An authentication and key agreement protocol can provide perfect forward secrecy if an adversary knowing both user's password PW_i and server's secret key x, but still can not compute previous session keys. For a session key $SK = h(AID_i||d_s(D)||CID_i)$ in our proposed scheme, U_i selects a random d_u and S chooses a random d_s for each session, the attacker needs to know d_u and d_s in order to get the session key. However, he only get d_uP and d_sP from PW_i ,x and the eavesdropped messages. That means an adversary need to compute d_sd_uP from d_uP and d_sP. It's a Diffie-Hellman problem in elliptic curve, which has no efficient polynomials algorithm solving it. Therefore, our authentication scheme posses perfect forward secrecy.

Mutual authentication.
In our scheme, an user U validates the message $\{M_3, M_4, CID_i', T_2\}$ by checking both the timestamp T_2 and the condition $M_4' = M_4$ are valid or not. S validates the message $\{M_5, T_3\}$ sent by patient using checking whether both the timestamp T_3 and the condition $M_5' = M_5$ hold. Also, an user and the server agree with a session key which is known with themselves.

Efficient login and password change.
In the login and password change phase of our proposed scheme, the smart card must verify $K_i \stackrel{?}{=} (V_i \oplus MP_i)$, where $MP_i = PW_i \oplus H(B_i) \oplus R_i \oplus h(ID_i||PW_i||H(B))$, which includes the patient's identity, password, and biometric. If it's not true, the smart card rejects the user's login and password changing request. The quick detection of incorrect identity, password and biometric make the proposed scheme efficient. Also, this verification can present denial of service attack well.

7 Discussion

This section give a comparison of security and performance of recent biometric-based authentication and key agreement schemes for TMIS [2, 3, 15, 18, 23, 26]. Table 2 describes that the flaws of security and efficiency for biometric based authentication schemes for TMIS.

In Table 2, we represent $\surd$ as the scheme which prevents attack or satisfies the attribute and $\times$ as the scheme which fails to prevent attack or does not satisfy the attribute. From Table 2, it is clear to see that most of previous biometric authentication schemes do not satisfy desirable security attributes. However, Yan et al.'s

Table 2: Security attributes comparison of biometric based authentication schemes

Security attributes \ Schemes	[15]	[26]	[3]	[23]	[2]	[18]	Ours
User anonymity	×	×	×	×	√	√	√
Off-line password guessing attack	√	×	×	×	×	√	√
Stolen smart card attack	√	√	√	√	√	√	√
Impersonation attack	×	√	√	√	√	×	√
Replay attack	√	√	×	√	√	√	√
Denial of service attack	√	√	×	√	√	√	√
Strong forward secrecy	√	√	√	×	×	×	√
Session key verification	√	√	×	√	√	√	√
Efficient password change	√	×	×	√	√	√	√

scheme [26] and Awasthi-Srivastava's scheme [3] with hash function computation have less computation overhead as compare to [2, 15] which have elliptic curve point multiplication costs, which is also shown in Table 3. Lu et al.'s scheme [15], Yan et al.'s scheme [26], Awasthi-Srivastava and Wen's schemes [3, 23] have the failure of protecting user anonymity. However, user anonymity during message exchange ensures consumer's privacy by preventing an attacker from acquiring consumer's sensitive personal information. Thus, an ID-based authentication scheme should ensure anonymity and unlinkability. The schemes [2, 3, 23, 26] can't resist against the off-line password guessing attack, which means an attacker is able to find user's correct password using an off-line exhaustive search for all possible passwords. Hence, a password-based authentication scheme should resist on-line and off-line password guessing attacks.

Lu et al.'s scheme [15] and Mishra et al.'s scheme [18] is vulnerable to impersonation attack, which means that an adversary could impersonate as a legal user to access any services. Awasthi-Srivastava's scheme [3] does not resist replay attack. In general, their schemes can remedy this security flaw by adding time stamp or a counter. Therefore, an secure authentication scheme should be secure against replay attack.

In the above discussed schemes of Table 2, the smart card cannot correctly identify the correctness of input which causes extra computation and communication overhead. The scheme [3] has flaws in password change phase and the schemes [3, 26] have inefficient password change phase. It is clear from the study that inefficient password change can cause DOS attack in case of incorrect input in password change phase, i.e., onetime mistake in password change phase, a valid user no longer login to the server using the same smart card. The authentication schemes could detect incorrect input quickly so that denial of service attack, and extra communication and computation overhead can be avoided.

Table 3 discusses the computation overhead of these schemes in login and authentication phase, where $T_{sym}, T_h, T_H, T_{ME}$ and T_{ECC} denote the time complexity of symmetric encryption/decryption, hash function, biometric hash function, modular exponentiation and elliptic curve point multiplication, respectively. It is noted that, $T_{ECC} > T_{ME} \gg T_{sym} \gg T_H \gg T_h$. Since the login and authentication phases are executed for each session while the registration and password change phases occur once, we only discuss the computational cost of the login and authentication phases.

8 Conclusions

We have analyzed the security of Lu et al.'s biometric based authentication schemes for TMIS. It is shown that their scheme is vulnerable to protect user anonymity, and an adversary could determine whether two messages are transmitted from the same user. The scheme is also insecure against impersonation attack which leads to an adversary could impersonate as a legal user to access any services provided by telecare server, and cheat a honest user as a legal server. Moreover, we employ biometric hash function and elliptic curve Diffie C Hellman problem to improve the security and efficiency of Lu et al.'s scheme. It is noted that the enhanced scheme does not provide all security attributes of three-factor authentication schemes.

Acknowledgments

This research is supported by National Basic Research Program of China (Grant No. 2013CB834205), Natural Science Foundation of Zhejiang Province (Grant No. LZ12F02005) and Opening project of Key Laboratory of Public Security Information Application Based on Big-data Architecture, Ministry of Public Security (Grant No. 2014DSJSY004).

References

[1] N. Anwar, I. Riadi, A. Luthfi, "Forensic SIM card cloning using authentication algorithm," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 71–81, 2016.

Table 3: Performance evaluation of biometric based authentication schemes

Schemes	User computation	Server computation
Lu et al.'s scheme [15]	$2T_{ECC} + T_H + 5T_h$	$T_{ECC} + 5T_h$
Yan et al.'s scheme [26]	$6T_h$	$5T_h$
Awasthi-Srivastava's scheme [3]	$4T_h + T_v b$	$3T_h$
Wen's scheme [23]	$2T_{sym} + 9T_h$	$2T_{sym} + 6T_h$
Arshad et al.'s scheme [2]	$2T_{ECC} + T_m + 8T_h$	$2T_{EEC} + 2T_m + 7T_h$
Mishra et al.'s scheme [18]	$T_H + 6T_h$	$2T_{sym} + 7T_h$
Our scheme	$2T_{ECC} + T_H + 5T_h$	$2T_{ECC} + 4T_h + T_{sym}$

[2] H. Arshad and M. Nikooghadam, "Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 12, pp. 136–147, 2014.

[3] A. K. Awasthi and K. Srivastava, "An enhanced biometric authentication scheme for telecare medicine information systems with nonce," *Journal of Medical Systems*, vol. 37, no. 12, pp. 9964–9976, 2013.

[4] C. C. Chang, W. Y. Hsueh, and T. F. Cheng, "An advanced anonymous and biometrics-based multi-server authentication scheme using smart cards," *International Journal of Network Security*, vol. 18, no. 6, pp. 1010–1021, 2016.

[5] Y. F. Chang, S. H. Yu, and D. R. Shiao, "An uniqueness and anonymity-preserving remote user authentication scheme for connected health care," *Journal of Medical Systems*, vol. 37, no. 12, pp. 9902–9910, 2013.

[6] H. M. Chen, J. W. Lo, and C. K. Yeh, "An efficient and secure dynamic id-based authentication scheme for telecare medical information systems," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3907–3915, 2012.

[7] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic id-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 629–631, 2004.

[8] D. B. He, J. H. Chen, and R. Zhang, "A more secure authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 2, pp. 1989–1995, 2012.

[9] S. H. Islam and G. P. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography," *Mathematical and Computer Modelling*, vol. 57, no. 12, pp. 2703–2717, 2013.

[10] A. T. Jin, D. Ling, and A. Goh, "Biohashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 37, no. 11, pp. 2245–2255, 2004.

[11] M. K. Khan, K. S. Kim, and K. Alghathbar, "Cryptanalysis and security enhancement of a more efficient secure dynamic idbased remote user authentication scheme," *Computer Communications*, vol. 34, no. 3, pp. 305–309, 2010.

[12] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of 19th Annual International Cryptology Conference (CRYPTO'99)*, pp. 388–397, Santa Barbara, California, USA, Aug. 1999.

[13] H. D. Le, N. T. Nguyen, and C. C. Chang, "Provably secure and efficient three-factor authenticated key agreement scheme with untraceability," *International Journal of Network Security*, vol. 18, no. 2, pp. 335–344, 2016.

[14] C. T. Li, "A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card," *IET Inforamtion Security*, vol. 7, no. 1, pp. 3–10, 2012.

[15] Y. Lu, L. Li, H. Peng, and Y. Yang, "An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem," *Journal of Medical Systems*, vol. 39, no. 32, pp. 1–9, 2015.

[16] A. Lumini and L. Nanni, "Biohashing: Two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 40, no. 3, pp. 1057–1065, 2007.

[17] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smartcard security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[18] D. Mishra, S. Mukhopadhyay, A. Chaturvedi, S. Kumari, and M. K. Khan, "Cryptanalysis and improvement of yan et al.'s biometric-based authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 24, pp. 1–12, 2014.

[19] E. O. Osei, J. B. Hayfron-Acquah, "Cloud computing login authentication redesign," *International Journal of Electronics and Information Engineering*, vol. 1, no. 1, pp. 1–8, 2014.

[20] Z. Tan, "An efficient biometrics-based authentication scheme for telecare medicine information systems," *Network*, vol. 2, no. 3, pp. 200–204, 2013.

[21] R. Wang amd W. Juang and C. Lei, "Robust authtication and key agreement scheme preserving the privacy of secret key," *Computer Communications*, vol. 34, no. 3, pp. 274–280, 2011.

[22] J. Wei, X. Hu, and W. Liu, "An improved authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3597–3604, 2012.

[23] F. T. Wen, "A robust uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care," *Journal of Medical Systems*, vol. 37, no. 6, pp. 1–9, 2013.

[24] Z.Y. Wu, Y.C. Lee, F. Lee, H.C. Lee, and Y. Chung, "A secure authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1529–1535, 2012.

[25] Q. Xie, J. Zhang, and N. Dong, "Robust anonymous authentication scheme for telecare medical information systems," *Journal of Medical Systems*, vol. 37, no. 2, pp. 1–8, 2013.

[26] X. Yan, W. Li, P. Li, J. Wang, X. Hao, and P. Gong, "A secure biometrics-based authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 37, no. 5, pp. 9972–9977, 2013.

[27] H. Zhu, "Secure chaotic maps-based group key agreement scheme with privacy preserving," *International Journal of Network Security*, vol. 18, no. 6, pp. 1001–1009, 2016.

[28] H. Zhu, Y. Zhang, H. Li, and L. Lin, "A novel biometrics-based one-time commitment authenticated key agreement scheme with privacy protection for mobile network," *International Journal of Network Security*, vol. 18, no. 2, pp. 209–216, 2016.

[29] Z. Zhu, "An efficient authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3833–3838, 2012.

Biography

Lidong Han received his Ph.D. degree from School of Mathematics, Shandong University, China, in 2010. Now, he work at Hangzhou Normal University. His research interests include Cryptography and cloud computing.

Qi Xie is a professor in Hangzhou Normal University of China. And he received his Ph.D. degree from Department of Mathematics(School of Mathematical Sciences), Zhejiang University, China, in 2010. His research interests include authentication and key exchange.

Wenhao Liu received his Ph.D. degree from University of Electronic Science and Technology of China in 2010. Now, he work at Key Laboratory of Cryptography and Network Security of Hangzhou Normal University. His research interests include signature and data security in cloud.

A Pseudo Random Number Generator Based on Chaotic Billiards

Khalid Charif, Ahmed Drissi, Zine El Abidine Guennoun
(Corresponding author: Khalid Charif)

Department of Mathematics, Mohamed V University in Rabat
No. 4, Avenue Ibn Battouta B. P. 1014 RP, Rabat, Morocco
(Email: Khalidcharif@gmail.com)
(Received Mar. 12, 2016; revised and accepted June 14 & July 17, 2016)

Abstract

The systems of chaotic billiards revealed a well developed chaotic behavior. Despite its good characteristics, these systems have not yet been applied to the cryptography; among the reasons is the difficulty of expressing the equation of particle motion in an explicit form. In this work, we took advantage the properties offered by the random walks and unpredictability of two particles moving in a chaotic billiard (Sinai Billiard) for the designing a new pseudo-random number generator. The results are subjected to an experimental study to test the randomness and the chaotic behavior of the generator. the key stream passed all the NIST statistical tests and the generator is highly sensitive for a bit change in the keys.

Keywords: Chaos, pseudo-random number generator, Sinai billiard

1 Introduction

The use of chaotic systems in cryptography has been well studied [17, 18, 19]. In fact there are similarities between the needs of cryptography and properties offered by the chaos. The algorithms based on chaos showed good performance for data encryption such as images, videos or audio data [1, 16, 20, 25]. Characterized by speed, reproducibility and simplicity of implementation, the PRNGs based on chaos took more attention. The first PRNG was proposed by Oishi and Inoue [21] in 1982, using the chaotic first order nonlinear differencial equations. After this article, several GNPAs were suggested. Generators have been proposed based on the logistic system in [2, 15, 23]. In [28], a generator based on the generalized Henon map. Using the Lorenz system, a new generator for the voice data encryption is designed in [1]. Chaotic standard system was applied in the conception of the generator in [22].

Our work focuses on an alternative approach based on the implementation of a system more concrete that has interesting chaotic properties, those are the systems of chaotic billiards in two dimensions [7]. They are among the classes of simple systems, which are still exploring chaos. The mathematical theory of billiards was introduced by Sinai in 1970 [26]. It is developed and evolved with remarkable speed to become a well grounded within the theory of dynamical systems theory and statistical mechanics. Several studies were devoted specifically to the chaotic billiards. In billiards where a particle moves with constant velocity and reflects off the border in accordance with the law: "the incidence angle is equal to the reflection angle". The angles and positions taken by the particle can be treated as random variables, which encouraged us to use it in the construction of a new PRNG. Sinai billiards is the first class of chaotic billiards, it is also called the dispersion system. A circular disc inside the billiard causes divergent trajectories.

This work is organized as follows. In Section 2, we present the Sinai billiard, its geometric shape, the direction of a particle travelling in the billiard table and its chaotic properties. In Section 3 we give a detailed description of our PRNG. A validation of the PRNG by test batteries and a study chaotic behavior of the sequences generated by our generator are reported in Section 4. In the final section, we draw a conclusion.

2 Presentation of the Sinai Billiard

The Sinai Billiard (Figure 1) is a planar area, consisting of a square of side $2a$ and a circular barrier with radius $r < a$ is placed at the center. A free billiard two-dimensional ($2D$) witch was proposed by Sinai in 1970 [26]. Billiards emerged to simplify the study of the behavior of two discs (gas molecule) bouncing by mutual collisions in a square. The dynamics of two interacting disks reduces to that of Sinai billiard. The billiard is sometimes called the Lorentz gas. The notations of the paper is listed in Table 1.

Table 1: Notations

$PRNG$	Pseudo-random number generator
S	the random sequence $S = S_1\, S_2\, \ldots$
Pw	Password
L	Password lenght
$[\,]$	The integer part
$\overline{p}$	Invert bits of p
A^i	The collision point at the i^{th} step
$D(O, D_n)$	The distance from O to D_n
Δ_n	Discriminant
f	Transition function
$\bigoplus$	Bitwise exclusive OR operator
HD	Hamming distance
$\|$	Concatenation operation

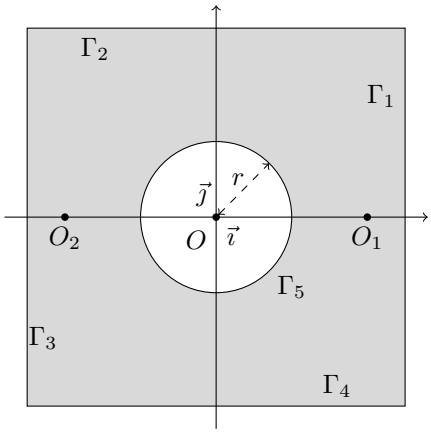


Figure 1: Sinai billard

2.1 The Billiard Geometric and Particle Direction Description

In an orthonormal $(O, \vec{\imath}, \vec{\jmath})$, we are interested in the billiard whose border $\Gamma = \cup_{i=1}^{5}\Gamma_i$ (Figure 1), is constituted by a square with side $2a$ and a circular hole of center O and raduis $r < \frac{3}{2}a$. A closed domain limited by portions in Euclidean space in two dimensions $\mathbb{R}^2$. The set $s = \cup_{i\neq j}^{4}\Gamma_i \cap \Gamma_j$ is the singular part of the border, it is composed of 4 points and $\Gamma \setminus s$ is the set of regular points of the border. In any regular point there is an internal normal vector $\overrightarrow{N}$. A point particle travels the billiard at a constant velocity v. When it reaches the border, undergoes elastic collision with specular reflection according to the law of reflection, the angle of incidence is equal to the angle of reflection with respect to $\overrightarrow{N}$ the normal vector at the border collision point. Between two collisions, the particle follows a straight path.

Initially, the particle oriented at an angle $\theta_0 = (\vec{\imath}, \overrightarrow{v_0})$ where $\vec{\imath}$ is the unit vector of x-axis. We have:

$$\overrightarrow{v_0} = \cos(\theta_0)\vec{\imath} + \sin(\theta_0)\vec{\jmath}.$$

After collision, we have Equation (1):

$$(\vec{\imath}, \overrightarrow{v_{new}}) = (\vec{\imath}, \overrightarrow{v_{old}}) + (\overrightarrow{v_{old}}, \overrightarrow{N}) + (\overrightarrow{N}, \overrightarrow{v_{new}}) \mod 2\pi. \quad (1)$$

After collision rule, we have:

$$(\overrightarrow{N}, \overrightarrow{v_{new}}) = (\overrightarrow{-v_{old}}, \overrightarrow{N}).$$

We have also:

$$(\overrightarrow{-v_{old}}, \overrightarrow{N}) = (\overrightarrow{-v_{old}}, \overrightarrow{v_{old}}) + (\overrightarrow{v_{old}}, \overrightarrow{N}) \mod 2\pi$$

$$= \pi + (\overrightarrow{v_{old}}, \overrightarrow{N}) \mod 2\pi.$$

Therefore Equation (1) becomes:

$$(\vec{\imath}, \overrightarrow{v_{new}}) = (\vec{\imath}, \overrightarrow{v_{old}}) + 2(\overrightarrow{v_{old}}, \overrightarrow{N}) + \pi \mod 2\pi. \quad (2)$$

At the $(n+1)^{th}$ collision, where $n \geq 0$, we put $\theta_n = (\vec{\imath}, \overrightarrow{v_{old}})$ and we obtain $\theta_{n+1} = (\vec{\imath}, \overrightarrow{v_{new}})$ and therefore Equation (2) becomes:

$$\theta_{n+1} = \theta_n + 2(\overrightarrow{v_n}, \overrightarrow{N_{n+1}}) + \pi \mod 2\pi \quad (3)$$

such as:

$$\overrightarrow{v_n} = \cos(\theta_n)\vec{\imath} + \sin(\theta_n)\vec{\jmath}$$

where $\overrightarrow{N_{n+1}}$ the unit normal vector at the border to the $(n+1)^{th}$ collision. We have after the geometric shape of the billiard:

$$\overrightarrow{N_{n+1}} = \begin{cases} -\dfrac{x_{n+1}}{|x_{n+1}|} & \text{if } |x_{n+1}| = a \\ -\dfrac{y_{n+1}}{|y_{n+1}|} & \text{if } |y_{n+1}| = a \\ \dfrac{x_{n+1}\vec{\imath} + y_{n+1}\vec{\jmath}}{\sqrt{x_{n+1}^2 + y_{n+1}^2}} & \text{otherwise} \end{cases}$$

We consider $A_{n+1}(x_{n+1}, y_{n+1})$ the point of $(n+1)^{th}$ collision, therefore A_{n+1} belongs to the intersection of the particles trajectory with billiard border Γ.
We define f, the transition function from (A_n, θ_n) to (A_{n+1}, θ_{n+1}) such as:

$$(x_{n+1}, y_{n+1}, \theta_{n+1}) = f(x_n, y_n, \theta_n).$$

2.2 The Transition Function f Description

We give the algorithm description of the transition function f between two collisions A_n and A_{n+1}:

$$f\colon [-a\,;a]^2 \times [0\,;2\pi[\to [-a;a]^2 \times [0\,;2\pi]$$

$$(x_n, y_n, \theta_n) \mapsto (x_{n+1}, y_{n+1}, \theta_{n+1}).$$

The equation of motion particle between two collisions is:

$$(D_n)\colon \sin(\theta_n)x - \cos(\theta_n)y - \sin(\theta_n)x_n + \cos(\theta_n)y_n = 0. \quad (4)$$

a) For $\theta_n \notin \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$.
From Equation (4), we have $y = \tan(\theta_n)(x - x_n) + y_n$,

we put $g_n(x) = \tan(\theta_n)(x - x_n) + y_n$ and we have $x = \cot(\theta_n)(y-y_n)+x_n$, we put $h_n(y) = \cot(\theta_n)(y-y_n)+x_n$.

The distance from O to D_n is:

$$d(O, D_n) = |-x_n \sin(\theta_n) + y_n \cos(\theta_n)|\,.$$

i) In case where $A_{n+1} \in (D_n) \cap \Gamma_5$.
$A_{n+1} \in (D_n) \cap \Gamma_5$ then $d(O, D_n) \leq r$ and $A_n \notin \Gamma_5$, therefore:

$$\begin{cases} (x_{n+1})^2 + (y_{n+1})^2 = r^2 \\ \sin(\theta_n)x_{n+1} - \cos(\theta_n)y_{n+1} - \sin(\theta_n)x_n + \cos(\theta_n)y_n = 0 \end{cases}$$

$$\Updownarrow$$

$$\begin{cases} (x_{n+1})^2 + (y_{n+1})^2 = r^2 \\ y = \tan(\theta_n)(x - x_n) + y_n = g_n(x) \end{cases}$$

We find

$$\begin{cases} (x_{n+1})^2 + (y_{n+1})^2 = r^2 \\ (5) \begin{cases} \dfrac{1}{\cos^2(\theta_n)}(x_{n+1})^2 + 2(y_n - \tan(\theta_n)x_n)\tan(\theta_n)x_{n+1} \\ +(y_n - \tan(\theta_n)^2 - r^2 = 0 \end{cases} \end{cases}$$

$\triangle_n$ discriminant of (5) is defined as:

$$\triangle_n = 4\Big(\frac{r^2}{\cos^2(\theta_n)} - (y_n - \tan(\theta_n)^2\Big)$$

Two possible solutions are:

$$x_{n+1} = \cos^2(\theta_n)\Big(- \big(y_n - \tan(\theta_n)\big)\tan(\theta_n) - \frac{\sqrt{\triangle_n}}{2}\Big)$$

or

$$x_{n+1} = \cos^2(\theta_n)\Big(- \big(y_n - \tan(\theta_n)\big)\tan(\theta_n) + \frac{\sqrt{\triangle_n}}{2}\Big)$$

and $y_{n+1} = g(x_{n+1})$. For $\theta_n \in \Big[0, \frac{\pi}{2}\Big] \cup \Big[\frac{3\pi}{2}, \pi\Big]$, we have:

$$x_{n+1} = \cos^2(\theta_n)\Big(- \big(y_n - \tan(\theta_n)\big)\tan(\theta_n) - \frac{\sqrt{\triangle_n}}{2}\Big)$$

and for $\theta_n \in \Big[\frac{\pi}{2}, \pi\Big] \cup \Big[\pi, \frac{3\pi}{2}\Big]$ we have:

$$x_{n+1} = \cos^2(\theta_n)\Big(- \big(y_n - \tan(\theta_n)\big)\tan(\theta_n) + \frac{\sqrt{\triangle_n}}{2}\Big)$$

ii) In case where $A_{n+1} \in (D_n) \cap (\cup_{i=1}^4 \Gamma_i)$.
$A_{n+1} \in (D_n) \cap (\cup_{i=1}^4 \Gamma_i)$ then $d(O, D_n) > r$ or $A_n \in \Gamma_5$ for $\theta_n \in \Big[0, \frac{\pi}{2}\Big] \cup \Big[\frac{3\pi}{2}, 2\pi\Big]$, we have:

$$(x_{n+1}, y_{n+1}) = \begin{cases} \big(a, g_n(a)\big) & \text{if } -a \leq g_n(a) \leq a \\ \big(h_n(a), a\big) & \text{if } g_n(a) > a \\ \big(h_n(-a), -a\big) & \text{otherwise} \end{cases}$$

for $\theta_n \in \Big[\frac{\pi}{2}, \pi\Big] \cup \Big[\pi, \frac{3\pi}{2}\Big]$, we have:

$$(x_{n+1}, y_{n+1}) = \begin{cases} \big(-a, g_n(-a)\big) & \text{if } -a \leq g_n(-a) \leq a \\ \big(h_n(a), a\big) & \text{if } g_n(-a) > a \\ \big(h_n(-a), -a\big) & \text{otherwise} \end{cases}$$

b) For $\theta_n \in \Big\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\Big\}$.

i) In case where $\theta_n \in \Big\{\frac{\pi}{2}, \frac{3\pi}{2}\Big\}$.

$$(x_{n+1}, y_{n+1}) =$$

$$\begin{cases} (x_n, a) & \text{if } A_n \in \Gamma_5 \text{ and } y_n \geq 0 \\ (x_n, -a) & \text{if } A_n \in \Gamma_5 \text{ and } y_n < 0 \\ (x_n, \sqrt{r^2 - x_n^2}) & \text{if } A_n \notin \Gamma_5 \text{ and } -r < x_n < r \text{ and } y_n \geq 0 \\ (x_n, -\sqrt{r^2 - x_n^2}) & \text{if } A_n \notin \Gamma_5 \text{ and } -r < x_n < r \text{ and } y_n < 0 \\ (x_n, -y_n) & \text{otherwise} \end{cases}$$

ii) In case where $\theta_n \in \{0, \pi\}$.

$$(x_{n+1}, y_{n+1}) =$$

$$\begin{cases} (a, y_n) & \text{if } A_n \in \Gamma_5 \text{ and } x_n \geq 0 \\ (-a, y_n) & \text{if } A_n \in \Gamma_5 \text{ and } x_n < 0 \\ (\sqrt{r^2 - y_n^2}, y_n) & \text{if } A_n \notin \Gamma_5 \text{ and } -r < y_n < r \text{ and } x_n \geq 0 \\ (-\sqrt{r^2 - y_n^2}, y_n) & \text{if } A_n \notin \Gamma_5 \text{ and } -r < y_n < r \text{ and } x_n < 0 \\ (-x_n, y_n) & \text{otherwise} \end{cases}$$

2.3 The Sinai Billiards Chaotic and Ergodic Properties

After the publication of the Article [26] in 1970, Sinai billiard has become popular, it has undergone many subsequent studies by many mathematicians and physicists authors [3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 26, 27]. The system shows a completely chaotic behavior [12, 26].

In general, the geometric shape of its boundaries determines dynamic billiards properties. It may consist of a convex curve, concave or linear. Sinai has shown in [26] that all billiards with the outwardly convex borders are always strongly chaotic. In [12], Per Dahlqvist has calculated an explicit expression of the Lyapunov exponent λ of Sinai billiard. λ measure quantitatively the chaos of dynamical systems, being positive means that there is a dependence on initial conditions (i.e the chaos existence). λ is positive for all values of the radius r of the disc.

Sinai has developed a method to prove that all dispersing billiard (Sinai billiard) is ergodic, mixing and it has a stronger property, the K-mixing. Moreover Gallavotti and Ornstein proved in [14] that the Sinai billiard is a Bernoulli system. The Bernoulli property is the strongest among the ergodic properties. It involves K-mixing, mixing and ergodicity.

3 Designing a PRNG Based on the Sinai Billiard

It is a deterministic pseudo-random numbers generator initialized by a password Pw with arbitrary sized, the output is a cryptographically secure binary sequence. we consider two point particles that move in the Sinai billiard with a constant velocity $\|\overrightarrow{v_1}\| = \|\overrightarrow{v_2}\| = 1$, without interaction between it. The departure point for the first particle (resp. second particle) is O_1 (resp. O_2) such as $\overrightarrow{OO_1} = \frac{3a}{4}\vec{i}$ (resp. $\overrightarrow{OO_2} = -\frac{3a}{4}\vec{i}$). Initially, it is oriented by $\overrightarrow{v_{0,1}}$ (resp. $\overrightarrow{v_{0,2}}$) such as $\theta_{0,1} = (\vec{i}, \overrightarrow{v_{0,1}})$ (resp. $\theta_{0,2} = (\vec{i}, \overrightarrow{v_{0,2}})$) where:

$$0 \leq \theta_{0,1}, \theta_{0,2} < 2\pi.$$

The angles $\theta_{0,1}$ and $\theta_{0,2}$ are calculated from the Pw using a technique based on a pointer, it positions on the Pw bits. The pointer moves from a position to other according to a linear congruential throughout the ASCII representation of Pw.

After initialization, we performed a predetermined number of collisions for the two particles, then start generating individuals necessary for the construction of the final sequence $S = S_1\, S_2 \ldots S_i \ldots$ with $S_i = I_{i,1} \bigoplus I_{i,2}$, $I_{i,1}$ and $I_{i,2}$ are two individuals generated in the i^{th} step. At each step i two individuals $I_{i,1}$ and $I_{i,2}$ of 32 bits will be generated based on the coordinates of the collision point of two balls with the square border of the billiard.

3.1 The Initial Values $\theta_{0,1}$ and $\theta_{0,2}$ Calculation

From a password $Pw = (p_{L-1} \ldots p_2 p_1 p_0)_2$, a binary string of any length L, we calculate the initialization angles $\theta_{0,1}$ and $\theta_{0,2}$. For each angle, we need to extract 64 bits from Pw. We consider a pointer pt that takes values indicating the bit positions in the Pw. The positions suite is defined as following:

$$\begin{cases} pt(0) & = 1 \\ pt(i+1) & = \left(\left(\left[\frac{L}{2}\right] + 1\right) \times pt(i) + 1\right) \bmod (L) \text{for } i \geq 0 \end{cases}$$

The pointer moves on the Pw, every time it positions on a new bit p_i and reads the information 0 or 1 necessary to calculate $\theta_{0,1}$ and $\theta_{0,2}$. We find $I_{0,1}$ and $I_{0,2}$ ($0 \leq I_{0,1}, I_{0,2} < 2^{64}$) as following:

$$\begin{aligned} I_{0,1} &= (\overline{p}_{pt(63)}\, p_{pt(62)} \cdots p_{pt(2)}\, \overline{p}_{pt(1)}\, p_1)_2 \\ &= p_1 + \sum_{i=0}^{31} \overline{p}_{pt(2\times i+1)} \times 2^i + \sum_{i=1}^{31} p_{pt(2\times i)} \times 2^i \end{aligned}$$

and

$$\begin{aligned} I_{0,2} &= (p_{L-1-pt(63)}\, \overline{p}_{L-1-pt(62)} \cdots p_{L-1-pt(1)}\, \overline{p}_{L-2})_2 \\ &= \overline{p}_{L-2} + \sum_{i=0}^{31} p_{pt(2\times i+1)} \times 2^i + \sum_{i=1}^{31} \overline{p}_{pt(2\times i)} \times 2^i \end{aligned}$$

finally

$$\theta_{0,1} = \frac{2\pi \times I_{0,1}}{2^{64}} \quad \text{and} \quad \theta_{0,2} = \frac{2\pi \times I_{0,2}}{2^{64}}$$

We call Initialize the initial values $\theta_{0,1}$ and $\theta_{0,2}$ calculation algorithm (Algorithm 1).

Algorithm 1 Calculation of $\theta_{0,1}$ and $\theta_{0,2}$

1: Begin
2: On taking password $Pw = (p_{L-1} \ldots p_2 p_1 p_0)_2$ a binary string of any length L .
3: $pt \leftarrow 1$
4: $I_{0,1} \leftarrow p_1$
5: $I_{0,2} \leftarrow p_{L-2}$
6: **for** $i = 1$ to 63 **do**
7: $\quad pt \leftarrow \left(\left(\left[\frac{L}{2}\right] + 1\right) \times pt + 1\right) \bmod (L)$
8: $\quad$ **if** i is even **then**
9: $\quad\quad I_{0,1} \leftarrow I_{0,1} + \overline{p}_{pt} \times 2^i$
10: $\quad\quad I_{0,2} \leftarrow I_{0,2} + p_{L-1-pt} \times 2^i$
11: $\quad$ **else** {i is odd}
12: $\quad\quad I_{0,1} \leftarrow I_{0,1} + p_{pt} \times 2^i$
13: $\quad\quad I_{0,2} \leftarrow I_{0,2} + \overline{p}_{L-1-pt} \times 2^i$
14: $\quad$ **end if**
15: **end for**
16: $\theta_{0,1} \leftarrow \frac{2\pi \times I_{0,1}}{2^{64}}$
17: $\theta_{0,2} \leftarrow \frac{2\pi \times I_{0,2}}{2^{64}}$
18: End

3.2 Generating the Pseudo-random Sequence

After calculating the initialization's angles $\theta_{0,1}$ and $\theta_{0,2}$, the two particles are ready to travel the billiard. Before starting to generate the individuals, we let the particles circulate and hit the billiard's wall until the e^{th} collision, we get $(x_k^0, y_k^0, \theta_k^0) = f^e(x_{0,k}, y_{0,k}, \theta_{0,k})$ for $k = 1, 2$ where e ($0 \leq e \leq 255$) is determined from the last 8 bits of the password $Pw = (p_{L-1} \ldots p_2 p_1 p_0)_2$ such as:

$$e = \sum_{i=0}^{7} p_i \times 2^i$$

At every step $i (i \geq 1)$, we carry out $(n_i + 1)$ collisions for both particles with $n_i (0 \leq n_i \leq 3)$ is determined by 2 bits taken directly from the password Pw as follows:

$$n_i = 2 \times p_{j+1} + p_j$$

where

$$j = 2 \times i \bmod (L - 1).$$

After $(n_i + 1)$ collisions, New coordinates are obtained $(x_k^i, y_k^i, \theta_k^i) = f^{n_i+1}(x_k^{i-1}, y_k^{i-1}, \theta_k^{i-1})$. We are interested in the collision coordinates with the square border of the billiard (i.e $|x_k^i| = a$ and $|y_k^i| = a$) ignoring the collisions

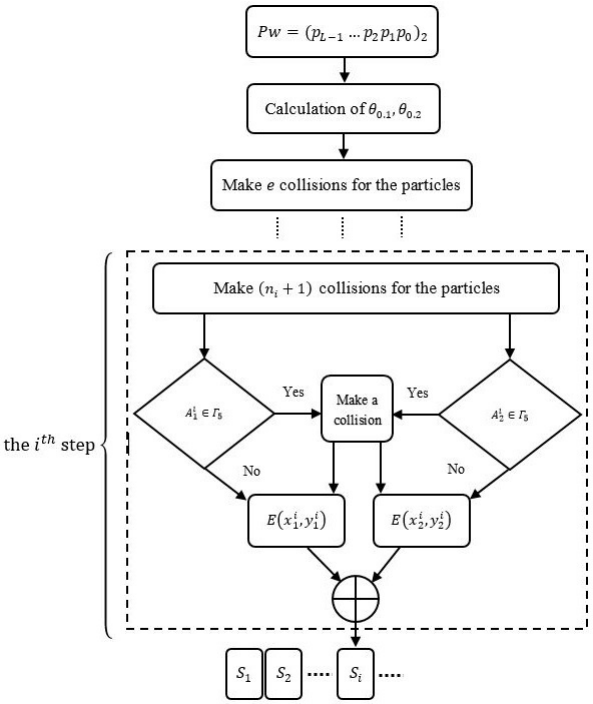


Figure 2: an operation step of our PRNG

with circle. If $A_k^i \in \Gamma_5$ (i.e $(x_k^i)^2 + (y_k^i)^2 = r^2$), we go to the next collision point for k^{th} particle $(x_k^i, y_k^i, {}^i_k) \longleftarrow f^{n_i+2}(x_k^{i-1}, y_k^{i-1}, \theta_k^{i-1})$, and then two individuals are generated values:

$$E(x_k^i, y_k^i) = \begin{cases} \left[2^{32}\frac{x_k^i}{a}\right] & \text{if } x_k^i \geq 0 \text{ and } \left|y_k^i\right| = a \\ \left[2^{32}\left(1 + \frac{x_k^i}{a}\right)\right] & \text{if } x_k^i < 0 \text{ and } \left|y_k^i\right| = a \\ \left[2^{32}\frac{y_k^i}{a}\right] & \text{if } y_k^i \geq 0 \text{ and } \left|x_k^i\right| = a \\ \left[2^{32}\left(1 + \frac{y_k^i}{a}\right)\right] & \text{if } y_k^i < 0 \text{ and } \left|x_k^i\right| = a \end{cases}$$

$$= I_{i,k} = (b_{31}^{i,k} b_{30}^{i,k} \dots b_1^{i,k} b_0^{i,k})_2$$

The output S of the PRNG is the concatenation of the sub-sequences $S_1, S_2, \dots, S_i \dots$ then:

$$S = S_1 S_2 \dots S_i \dots,$$

with

$$S_i = I_{i,1} \bigoplus I_{i,2},$$

where $I_{i,1}$ and $I_{i,2}$ two individuals are generated at the i^{th} step.

The algorithm have two input parameters, a password Pw and an integer N which indicates the length of the requested binary sequence as shown in (Algorithm 2).

Algorithm 2 Generation the random suit RS

```
Begin
θ1, θ2 ← Initialize(Pw)
(x1, y1) ← (0, 3/2 a)
(x2, y2) ← (0, −3/2 a)
e ← p0
for i = 1 to 7 do
    e ← e + p_i × 2^i
end for
(x1, y1, θ1) ← f^e(x1, y1, θ1)
(x2, y2, θ2) ← f^e(x2, y2, θ2)
I_{0,1} ← p1
I_{0,2} ← p_{L−2}
i ← 1
j ← i mod (L − 1)
n ← 2 × p_{j+1} + p_j
l ← 0
while l < ⌈N/32⌉ do
    (x1, y1, θ1) ← f^{n+1}(x1, y1, θ1)
    (x2, y2, θ2) ← f^{n+1}(x2, y2, θ2)
    if (x1)^2 + (y1)^2 = r^2 then
        (x1, y1, θ1) ← f(x1, y1, θ1)
    end if
    if (x2)^2 + (y2)^2 = r^2 then
        (x2, y2, θ2) ← f(x2, y2, θ2)
    end if
    I1 ← E(x1, y1)
    I2 ← E(x2, y2)
    RS ← RS||(I1 ⊕ I2)
    l ← l + 1
    i ← i + 1
    j ← 2 × i mod (L − 1)
end while
End
```

4 Security Analysis

A PRNG should verify security properties to resist the attacks. The security analysis must be done with care to assess the quality of the sequences. We study in the following paragraphs, the key space size, sensitivity to initial conditions and the level of randomness of the sequences. In the following study we fixed r at $\frac{a}{2}$.

4.1 The Key Space

The size of the key space is among the criteria by which a crypto-systems to be robust, a large size makes brute force attacks infeasible. Our algorithm has as an initialization key, a binary string of any size as mentioned above. The two particles billiards need exactly 128 bits to calculate its initial orientations. These 128 bits are extracted via a pointer that traverses the password Pw. This leads us to say that the size of the key space is large enough to be attacked exhaustively.

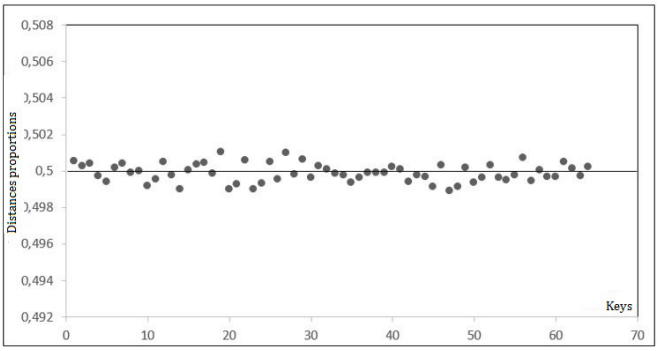


Figure 3: The proportion $\frac{DH(S^0, S^i)}{N}$

4.2 Sensitivity to Key

The sensitivity to a small change in the key is one of the essential properties for a PRNG. In other words, a small difference in the seeds of the system should cause a big change in the pseudo-random sequences. This property makes the generator highly secured against statistics and differential attacks, and so the sequence can not be broken even if there is a small difference between the keys. In our case, the generator is based on two dynamic systems of a purely chaotic billiard [12, 26]. In fact, to analyze the chaotic behavior of our generator, we place several k_i keys in the input of the generator with a bit of difference between it. A pseudo-random sequences S^i of size $N = 10^6$ are generated. The Hamming distance between two binary sequences $S^i = x_{1,i}\, x_{2,i} \ldots x_{N,i}$ and $S^j = x_{1,j}\, x_{2,j} \ldots x_{N,j}$ of equal length N is the number $DH(S^i, S^j) = card\{d/\, x_{d,i} \neq x_{d,j}\}$. Thus, for the two binary sequences S_i and S_j, the Hamming distance is given by:

$$DH(S^i, S^j) = \sum_{t=1}^{N} x_{t,i} \bigoplus y_{t,i}$$

In the case where the generator is chaotic, this distance is generally ranges around $\frac{N}{2}$, witch gives $\frac{DH(S^i, S^j)}{N}$ is approximately 0.5 for each pair of sequences produced.

We generate a group of pseudo-random sequences $\{S^i\}_{0 \leq i \leq 64}$ using the keys $\{k_i\}_{0 \leq i \leq 64}$. the key k_0 = "$GUENNOUN$", its binary representation in ASCII code is k_0 = (01000111 01000101 010011100100111 01001111 01010101 01001110)$_2$. The other 64 keys $\{k_i\}_{1 \leq i \leq 64}$ are derived from k_0, by changing the i^{th} bit among the 64 bits of k_0 to find k_i. The value $\frac{DH(S^0, S^i)}{N}$ between the sequences is shown in the graph 3.

From the results obtained, the differences in proportions between the sequences are approximately 0.5, indicating that the proposed generator is highly sensitive to initial conditions. Sensitivity to a small perturbation in the key for our generator is due to two reasons:

1) The generator is based in its construction on a system of a chaotic billiard, so the generated sequences inherit the chaos and unpredictability of the billiard. A number additional of iterations extracted directly from the password allows the generator to benefit maximally of the chaos offered by the billiard;

2) The initialization angles are taken from the Pw, using a pointer that points to different positions until the its total cover. Indeed, a difference in a bit between two keys may cause a different orientation to the particles and thus to the generated sequences.

Sinai billiard is chaotic for all values of the radius r, but there is a difference in the chaos level for each value of r as shown in [12] where the Lyapunov exponent is expressed in terms of r. Therefore, the user can control the level of the chaos generator by an input parameter at the algorithm (r, where $0 < r < \frac{3}{2}a$).

In the next section, we examine the randomness of the generator by statistical tests NIST (National Institute of Standards and Technology), which are considered the most valued.

4.3 Statistical Tests

The NIST Statistical Test Suite [24] is a statistical package, the result of collaboration between the statistical Engineering Division (SED) at NIST and the Computer Security Division. This suite consists of 16 tests, developed to quantify and assess the degree of random binary sequences produced by cryptographic generators. For each statistical test, a P_{value} is calculated from the bit sequence. This P_{value} is compared to a predefined threshold α, which is also called significance level. If P_{value} is greater than α, then the sequence is considered to be random with $1 - \alpha$ confidence level, and it proceeds the statistical test successfully, otherwise the sequence does not appear random. Generally, as suggested by NIST, α is set to its default value of 0.01, it indicates that one would expect 1 sequence in 100 sequences to be rejected.

To test our PRNG and as recommended by the NIST, we generated 1000 sequences, the length of each sequence is 1000^6 from a randomly selected keys. The test results on the sequences are presented in Table 2.

The minimum pass rate for the test Random Excursions (Variant) is approximately 609 for a sample of 625 binary sequences. The minimum pass rate for other tests is approximately 980 for a sample of 1000 binary sequences. We can see that the number of sequences that have managed to pass each test is greater than the minimum rate. Therefore, the proposed generator passed all NIST statistical tests. We can conclude that the numbers generated by the PRNG are random.

Table 2: Results of testing our generator on NIST test suite

Test Name	The P_{value}	The proportion	Result
Frequency	0.695200	992/1000	Success
Block-Frequency	0.861264	990/1000	Success
Cumulative Sums (1)	0.169981	995/1000	Success
Cumulative Sums (2)	0.978072	991/1000	Success
Runs	0.542228	985/1000	Success
Longest Run	0.709558	985/1000	Success
Rank	0.169981	995/1000	Success
FFT	0.080027	984/1000	Success
Non-Overlapping	0.505854	987/1000	Success
Overlapping	0.041169	991/1000	Success
Universal	0.334538	991/1000	Success
Approximate Entropy	0.851383	989/1000	Success
Random Excursions	0.478175	616/625	Success
Random Excursions Variant	0.470796	616/625	Success
Serial (1)	0.919131	986/1000	Success
Serial (2)	0.334538	980/1000	Success
Linear Complexity	0.948298	992/1000	Success

5 Conclusion

The PRNG proposed after a rigorous analysis, showed encouraging results, it is sensitive to a small change in the key and passed the NIST statistical test suite. Our generator has inherited the Sinai billiard unpredictability. It can be used for critical cryptographic applications. Furthermore, the systems of the chaotic billiards are good candidates to get into new cryptographic system design.

References

[1] M. Ahmad, B. Alam, and O. Farooq, "Chaos based mixed keystream generation for voice data encryption," *arXiv preprint arXiv: 1403.4782*, 2014.

[2] M. Andrecut, "Logistic map as a random number generator," *International Journal of Modern Physics B*, vol. 12, no. 9, pp. 921–930, 1998.

[3] M. V. Berry, "Quantizing a classically ergodic system: Sinai's billiard and the KKR method," *Annals of Physics*, vol. 131, no. 1, pp. 163–216, 1981.

[4] L. A. Bunimovich, "On billiards close to dispersing," *Matematicheskii Sbornik*, vol. 136, no. 1, pp. 49–73, 1974.

[5] L. A. Bunimovich, "On ergodic properties of certain billiards," *Functional Analysis and Its Applications*, vol. 8, no. 3, pp. 254–255, 1974.

[6] L. A. Bunimovich, Y. G. Sinai, and N. I. Chernov, "Statistical properties of two-dimensional hyperbolic billiards," *Russian Mathematical Surveys*, vol. 46, no. 4, pp. 47–106, 1991.

[7] N. Chernov and R. Markarian, "Chaotic billiards," *Mathematical Surveys and Monographs*, vol. 127, 2006.

[8] N. I. Chernov, "Sinai billiards under small external forces," *Annales Henri Poincaré*, vol. 2, pp. 197–236, Springer, 2001.

[9] N. I. Chernov and C. Haskell, "Nonuniformly hyperbolic k-systems are bernoulli," *Ergodic Theory and Dynamical Systems*, vol. 16, no. 1, pp. 19–44, 1996.

[10] N. Chernov, "Decay of correlations and dispersing billiards," *Journal of Statistical Physics*, vol. 94, no. 3-4, pp. 513–556, 1999.

[11] N. Chernov and L. S. Young, "Decay of correlations for lorentz gases and hard balls," *Hard Ball Systems and the Lorentz Gas*, pp. 89–120, Springer, 2000.

[12] P. Dahlqvist, "The lyapunov exponent in the sinai billiard in the small scatterer limit," *Nonlinearity*, vol. 10, no. 1, pp. 159, 1997.

[13] P. Dahlqvist and R. Artuso, "On the decay of correlations in sinai billiards with infinite horizon," *Physics Letters A*, vol. 219, no. 3, pp. 212–216, 1996.

[14] G. Gallavotti and D. S. Ornstein, "Billiards and bernoulli schemes," *Communications in Mathematical Physics*, vol. 38, no. 2, pp. 83–101, 1974.

[15] C. Guyeux, Q. Wang, and J. M. Bahi, "A pseudo random numbers generator based on chaotic iterations: Application to watermarking," in *Web Information Systems and Mining*, pp. 202–211, Springer, 2010.

[16] A. Jolfaei and A. Mirghadri, "Image encryption using chaos and block cipher," *Computer and Information Science*, vol. 4, no. 1, pp. 172, 2010.

[17] L. Kocarev, G. Jakimoski, T. Stojanovski, and U. Parlitz, "From chaotic maps to encryption schemes," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'98)*, vol. 4, pp. 514–517, 1998.

[18] A. Lasota and M. C. Mackey, *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*, vol. 97, Springer Science & Business Media, 1998.

[19] S. Li, Q. Li, W. Li, X. Mou, and Y. Cai, "Statistical properties of digital piecewise linear chaotic maps and their roles in cryptography and pseudo-random coding," in *Cryptography and Coding*, pp. 205–221, Springer, 2001.

[20] S. Lian, J. Sun, J. Wang, and Z. Wang, "A chaotic stream cipher and the usage in video protection," *Chaos, Solitons & Fractals*, vol. 34, no. 3, pp. 851–859, 2007.

[21] S. Oishi and H. Inoue, "Pseudo-random number generators and chaos," *IEICE Transactions*, vol. 65, no. 9, pp. 534–541, 1982.

[22] V. Patidar and K. K. Sud, "A novel pseudo random bit generator based on chaotic standard map and its testing," *Electronic Journal of Theoretical Physics*, vol. 6, no. 20, pp. 327–344, 2009.

[23] V. Patidar, K. K. Sud, and N. K. Pareek, "A pseudo random bit generator based on chaotic logistic map and its statistical testing," *Informatica*, vol. 33, no. 4, 2009.

[24] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, Technical Report, DTIC Document, 2001.

[25] F. Shang, K. Sun, and Y. Cai, "An efficient mpeg video encryption scheme based on chaotic cipher," in *IEEE Congress on Image and Signal Processing (CISP'08)*, vol. 3, pp. 12–16, 2008.

[26] Y. G. Sinai, "Dynamical systems with elastic reflections," *Russian Mathematical Surveys*, vol. 25, no. 2, pp. 137–189, 1970.

[27] L. S. Young, "Statistical properties of dynamical systems with some hyperbolicity," *Annals of Mathematics*, vol. 147, no. 3, pp. 585–650, 1998.

[28] F. Zheng, X. J. Tian, J. Y. Song, and X. Y. Li, "Pseudo-random sequence generator based on the generalized henon map," *The Journal of China Universities of Posts and Telecommunications*, vol. 15, no. 3, pp. 64–68, 2008.

Khalid Charif is a Ph.D student at the Faculty of Science, Mohamed V University in Rabat. He obtained his master's degree in mathematics and statistics, option cryptography and information security at the same university in 2013. His research interests include information security and cryptography.

Ahmed Drissi received his Ph.D degree in cryptology from the Faculty of Science, University Ibn Zohr Agadir, Morocco in 2014. His research interests include Code theory and the Cryptology. Currently he is associate member of the Laboratory for Analysis, Algebra and decision aid (LA3D). Faculty of Sciences Rabat, Morocco.

Zine El Abidine Guennoun is a professor of Department of Mathematics at the Faculty of Science, Mohamed V University in Rabat, Morocco. He received his Ph.D. (1989). His research interests include non linear analysis, fixed point theory, differential equation, financial mathematics and cryptography.

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.