Double Verifiable Lossless Secret Sharing Based on Hyper-chaos Generated Random Grid

Hang Gao[1], Mengting Hu[1], Tiegang Gao[2], and Renhong Cheng[1]
(Corresponding author: Tiegang Gao)
College of Computer and Control Engineering, Nankai University[1]
Tianjin Haihe Education Park, No. 38, Tongyan Road, Tianjin, China
(Email: gaotiegang@nankai.edu.cn)
College of Software, Nankai University[2]
Tianjin Haihe Education Park, No. 38, Tongyan Road, Tianjin, China
(Received July 27, 2016; revised and accepted Nov. 15 & Dec. 25, 2016)

Abstract

A novel multi secret sharing scheme is proposed in this paper. In the scheme, n secret images are firstly encrypted by hyper-chaos whose initial values are hash of the image itself. Then the encrypted images are shuffled into n confused images; lastly, the confused images are used to generate n sharing images using the random grid method. The proposed scheme has the advantage such that it can lossless restore the original secret image, and have the double verification ability, that is to say, it can verify whether the anyone of the sharing is modified, and it can also verify the whether the anyone of the original secret image is completely reconstructed. The above advantages make it especially suitable for secret sharing of important images such as medical and military images. Experimental results and some comparison analysis are given to testify the effectiveness of the proposed scheme.

Keywords: Hash-256; Hyper-chaos; Lossless Restoration; Random Grid; Visual Secret Sharing

1 Introduction

In general, a secret is regarded as safer when it is given two or more participants than that it is kept by only one person. Based on this kind of idea, Blakley and Shamir proposed the concept of secret sharing, respectively, in 1979 [2, 24]. As a kind of secret sharing, Naor and Shamir (1995) proposed a new type of secret sharing called visual cryptography (VC) or visual secret sharing (VSS) for images [21]. VC can encrypt a secret image into numerous meaningless sharing images, and anyone of the shared images does not reveal any information about the secret, In a general threshold VC scheme, a secret image is encoded into n meaningless random shares. The n shares are distributed to n corresponding participants. In order to reconstruct the original secret image, at least k or more participants are needed to share their shares, but any k?1 or fewer shares have no means to give clue about the secret.

In the last two decades, VC algorithm has aroused many interests among the researchers. Various proposals of the VC schemes for different situation have been proposed [6, 7, 9, 13, 20, 26, 31]. For example, people have presented extended VC schemes to encode the secret image into natural-looking shares [26, 31]; the progressive VC schemes is that the restored secret image can be shown in a progressive perceptual quality [9, 13]; prevention of cheating in VC is the scheme that break the misleading secret by dishonest participants [6, 7].

Another interesting VSS scheme is random grid (RG) based algorithm which has the advantages such as no pixel expansion in sharing secret. RG is firstly proposed by Kafri and Keren in 1987 [15]. Since then, some new researches have been made to probe extensive application scene of the RG based VSS scheme. Among them, Shyu proposed a RG-based scheme for gray-level and color images, and Chen et al. proposed a Threshold RG-based scheme for color images [3, 25]. In the study of meaningful shares in RG-based VCC, Chen and Tsao proposed a novel RG-based VSS scheme by skillfully designing a procedure of distinguishing different light transmissions on shared images, and the visual quality between meaningful random-grids and superimposed results can be adjusted to be more friendly [4]. Abd El-Latif, et al. proposed a scheme combines the error diffusion technique, RG and chaotic encryption to encode a secret binary image into meaningful shadow images [1]. Yan et al. proposed a generalized RG-based VC with meaningful shares which can support (k,n) threshold and provide adaptive visual quality, at the cost of slightly decreasing visual quality of shared images [28], and Chen proposed a lossless RG-based VSS scheme [5].

In recent years, security of VSS in different scenarios has aroused people's attentions [12, 16]. As different ap-

plication situations have different requirements. In 2006, Horng et al. showed that cheating is possible in (k,n) VSS scheme, the cheating problem will happen when dishonest participants collude to cheat honest ones by enabling the latter to accept the wrong secret information generated by the former [12]. Lee demonstrated that the inside collusion attacks to the RG-based VSS schemes is possible, and gives some examples for verifying feasibility of cheating [16]. In order to cope with cheating in VSS, some people proposed schemes to verify the genuineness of shares through generating extra verification shares [8, 18, 19, 27], for example, Wu et al. proposed a cheating immune method to provided extra ability of cheat-preventing for RG-based VSS [27]. Lin et al. exploited the hybrid codebook to hide the additional verification images into the share images to prevent cheating [18]; in order not to resort to any additional dedicated verification share, Lin et al. proposed a VC-based scheme with the ability to prevent cheating, the scheme designed an authentication pattern stamping process to detect the faked shares provided by malicious participants [19].

In real application of image encryption, for medical, satellite and military image, it is demanded that the encryption algorithm is safe. Moreover, the decrypted image must be lossless restored and verifiable. Inspired by the above algorithm, we proposed a RG-based VSS scheme for grey image, the main advantages of scheme include mini pixel expansions in sharing secret, verification of sharing secret and verification of restored secret image. Besides, the proposed algorithm can lossless restore secret image, the above features of the proposed scheme make it especially suitable for application of important images such as medical, satellite and military.

The rest of the paper is organized as follows. Some preliminaries are introduced in Section 2. The proposed scheme is describe in Section 3 and experimental results and analysis are given in Section 4, in the last, some conclusions are described in Section 5.

2 Preliminaries

In this section, some relative technology such as Hash function, RG-based VSS and the hyper-chaotic system based encryption algorithm are firstly introduced.

2.1 Hash Function

In computer science, one-way functions is a function that is easy to compute on every input, but it is very difficult to compute their inverse functions. That is to say, for a given data x, it is easy to calculate one-way function of x, on the other hand, knowing the value of one-way of x, it is quite difficult to calculate the value of x.

Hash functions is one-way function that converts input messages of any length into output sequences of fixed length, the output sequence is often called a hash value. A hash function has the properties of sensitivity to initial conditions, diffusion and confusion, collision resistance. This means that it should be very difficult to find two different sequences that produce the same hash value. These characteristics make it can be used for verification of data integrity. It has played an important role in the field of information security [34].

At present, some typical hash function includes MD5, SHA-1 and SHA-2. Among them, and the most important hash functions is the SHA family, which shares the same functional structure with some variation in the internal operations, message size, message block size, word size, number of security bits and message hash size [22]. In this paper, SHA-1 with the 256 bits output is used for generation of verification information.

2.2 RG-based VSS

The RG-based VSS was firstly proposed by Kafri, in the scheme, for a given secret binary image, two random grids and were generated, and anyone of them will leak no information about the binary image individually, yet they reveal the secret binary image when the two grids were superimposed. One of the three algorithms is given in follows.

1) For a binary secret image B with the size of $N \times M$, generate a random grid R_1 which includes only 0 and 1, the size of R_1 is the same as that of secret image.

2) For every pixel value $B_{(i,j)}$, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, M$, if $B_{(i,j)}$ is 0, then the value of $R_2(i, j)$ is equal to $R_1(i, j)$, else $R_2(i, j)$ is equal to the complement of $R_1(i, j)$.

3) R_1 and R_2 are the random grid.

The original secret binary image can be restored by superimposing R_1 and R_2 together. For example, in Figure 1, Figure 1(a) is the original secret image, Figures 1(b) and (c) are random grids, and the Figure 1(d) is the restored image by Figures 1(b) and (c).

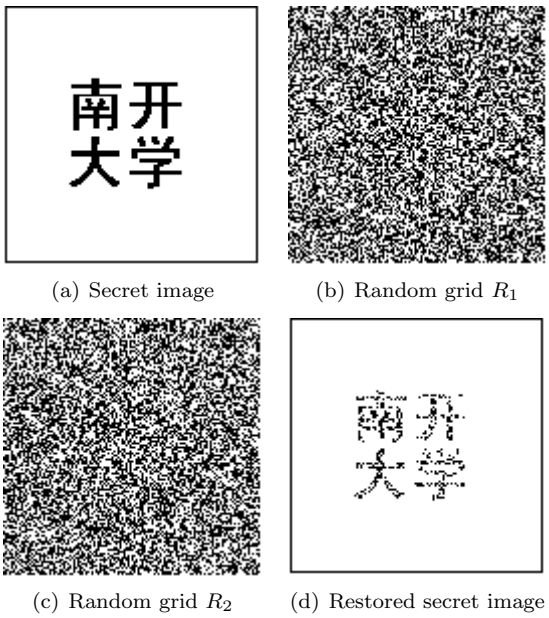
2.3 The Hyper-chaotic System

In the proposed scheme, a hyper-chaos system which is modeled by Equation (1) is used for generation of random grid.

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) \\ \dot{x}_2 = -x_1 x_3 + d x_1 + c x_2 - x_4 \\ \dot{x}_3 = -x_1 x_2 - b x_3 \\ \dot{x}_4 = x_1 + k \end{cases} \tag{1}$$

where a, b, c, d and k are parameters, when $a = 36, b = 3, c = 28, d = -16$ and $-0.7 \leq k \leq 0.7$, the system is hyper-chaotic, and its attractors are shown in Figure 2 with parameters $a = 36, b = 3, c = 28, d = -16$ and $k = 0.2$, its Lyapunov exponents are $\lambda_1 = 1.552, \lambda_2 = 0.023, \lambda_3 = 0, \lambda_4 = -12.573$.

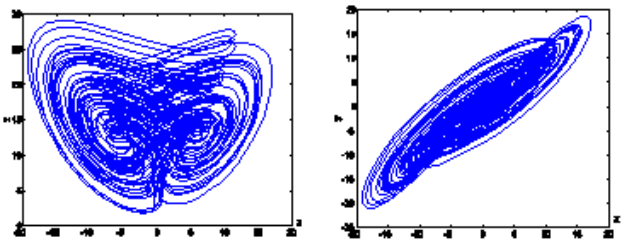
Because the hyper-chaos has two positive Lyapunov exponents, so the prediction time of a hyper-chaotic system



(a) Secret image (b) Random grid R_1

(c) Random grid R_2 (d) Restored secret image

Figure 1: Experimental results of random grid



(a) $x_1 - x_3$ plane (b) $x_1 - x_2$ plane

Figure 2: Hyper-chaos attractors of System (1) with $k = 0.2$

is shorter than that of a chaotic system [29], as a result, it is safer than chaos in security algorithm. For more detailed analysis of the complex dynamics of the system, please see relative reference [10].

As the hyper-chaos has the ergodicity, sensitive features on initial conditions and control parameters of chaotic maps and random-like behaviors, these features make it suitable for generating pseudo-random sequences and key sequences in cryptography [11, 23, 32, 33]. Among various image encryption algorithm based on hyper-chaos, it has been shown that some scheme can be effectively broken with known plaintext and chosen plaintext attacks, and people have given detailed both mathematical analysis and experimental results to testify the security weakness and potential risk of suffering statistical attacks [14, 17, 30].

In application of random grid based secret sharing, Ahmed A. proposed a novel secret image sharing scheme [1]. The scheme combines random grids (RG), error diffusion (ED) and chaotic permutation, it has the advantages of simple computation, alternative order of shadow images in recovery, avoids the design of complex codebook, and avoids the pixel expansion problem.

3 The Proposed Scheme

In this section, we give detailed description of the process of sharing secret generation and restoration of original secret image, and for simplicity,we only discuss the algorithm for grey image. The general flowchar of the scheme can be described in Figure 3.

3.1 The Generation of Sharing Secret

Step 1: For secret images $I_1, I_1, \cdots, I_m$, firstly, the message authentication code (MAC) with the size of 256 bit for every image is calculated, thus m message authentication codes (MAC) $H_1, H_2, \cdots, H_m$ are obtained.

Step 2: For every $H_i, i = 1, 2, \cdots, m$, assume its 256 bits are expressed by $h_1, h_2, \cdots, h_{256}$, then it is truncated into 64 bit by Equation (2).

$$\begin{cases} h_i^{'} = h_i \otimes h_{i+128}, i = 1, 2, \cdots, 128, \\ h_i^{''} = h_i^{'} \otimes h_{i+64}^{'}, i = 1, 2, \cdots, 64, \end{cases} \tag{2}$$

Step 3: For the produced 64 bit data, it is divided into 4 sections, every section includes 16 bits, apply Equation (3) to turn the 16 bits data into a integer which belongs to [0, 65535]. Thus, we can get 4 integer numbers.

$$\begin{cases} x_1 = Bin2dec(h_1^{''} h_2^{''}, \cdots, h_{16}^{''}) \\ x_2 = Bin2dec(h_{17}^{''} h_{18}^{''}, \cdots, h_{32}^{''}) \\ x_3 = Bin2dec(h_{33}^{''} h_{34}^{''}, \cdots, h_{48}^{''}) \\ x_4 = Bin2dec(h_{49}^{''} h_{50}^{''}, \cdots, h_{64}^{''}) \end{cases} \tag{3}$$

Step 4: Multiply the above generated 4 number by 10^{-5}, and give it to four initial values $x_1(0), x_2(0), x_3(0), x_4(0)$.

$$\begin{cases} x_1(0) = x_1 \times 10^{-5} \\ x_2(0) = x_2 \times 10^{-5} \\ x_3(0) = x_3 \times 10^{-5} \\ x_4(0) = x_4 \times 10^{-5} \end{cases} \tag{4}$$

Step 5: Implement encryption of secrets image $I_1, I_1, \cdots, I_m$ based on hyper-chaos using the method proposed by Gao [11], that is to say, for every image I_i, the following steps are done.

1) Iterate the hyper-chaotic system for N_0 times by Runge-Kutta algorithm to avoid the harmful effect of transient procedure.

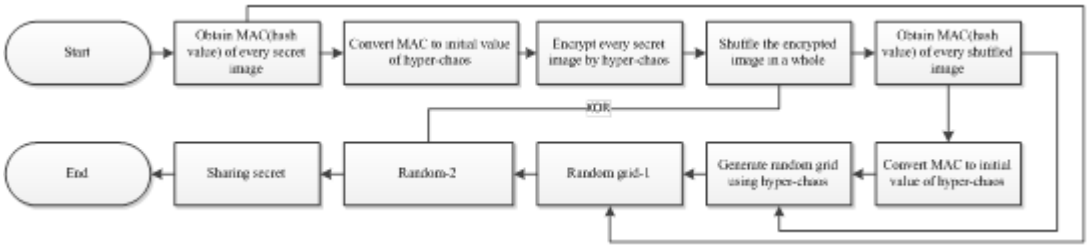


Figure 3: The general flowchart of the proposed scheme

2) The hyper-chaotic System (1) is iterated, and as a result, four decimal fractions x_1, x_2, x_3, x_4 will be generated. These decimal values are preprocessed firstly as follows

$$x_i = mod((Abs(x_i) - floor(Abs(x_i)) \times 10^{14}, 256) \tag{5}$$

where$Abs(x_i)$ returns the absolute value of x_i.$Floor(x)$ returns the value of x to the nearest integers less than or equal to x,$mod(x, y)$ returns the remainder after division.

3) Encrypt the image by formula according to Equation (6):

$$\begin{cases} C_{3\times(i-1)+1} = P_{3\times(i-1)+1} \otimes x_1 \\ C_{3\times(i-1)+2} = P_{3\times(i-1)+2} \otimes x_2 \\ C_{3\times(i-1)+3} = P_{3\times(i-1)+3} \otimes x_3 \\ C_{3\times(i-1)+4} = P_{3\times(i-1)+4} \otimes x_4 \end{cases} \tag{6}$$

where $i = 1, 2, \cdots$ represents the i^{th} iteration of the hyper-chaotic system. The symbol$\otimes$ represents the exclusive OR operation bit-by-bit.$P_i, i = 1, 2, \cdots, N \times M$ represents pixel values of the secret image, The process does not end until the set is all encrypted. Then the encrypted pixel set $C_i, i = 1, 2, \cdots, N \times M$ is written to the cipher-image.

When all the secret image are encrypted, m encrypted image $I_i^{'}, i = 1, 2, \cdots, m$ are gotten. Obviously, as the initial values of hyper-chaos are different, so it means that the encryption algorithm is One-Time Pad, which is information-theoretically secure in that the encrypted message provides no information about the original message to a cryptanalyst (except the length of the message).

Step 6: Put the images $I_i^{'}, i = 1, 2, \cdots, m$ in a row, so a matrix with the size M row and $N \times m$ column are presented. For convenience, it is assumed that every image is with the size of $M \times N$.

Step 7: Shuffle the encrypted image. Take one column from every image in turn until column are obtained, these columns are combined into a image $I_i^{''}$. Then in the same way, take the remainder columns in order from the m encrypted images, every column is combined into one image. Finally, m confused images $I_i^{''}, i = 1, 2, \cdots, m$ are generated.

Step 8: For every confused image of $I_i^{''}, i = 1, 2, \cdots, m$, repeat Step 1, then another m message authentication code (MAC) $H_i^{'}, i = 1, 2, \cdots, m$ which belongs to $I_i^{''}, i = 1, 2, \cdots, m$ are obtained.

Step 9: For every MAC $H_i^{'}, i = 1, 2, \cdots, m$, repeat Step 2 to Step 4, then four initial values of hyper-chaos are generated.

Step 10: For above four initial values $x_1(0)$, $x_2(0)$, $x_3(0)$, $x_4(0)$. Iterate the hyper-chaotic system by Runge-Kutta algorithm with initial parameters, the step of progression is 0.01, and the number of iterations is $M \times (N+1)$. Then we will get four decimal fractions in each iteration, and then preprocesses the four decimal fractions as follows

$$x_i^* = mod((Abs(x_i) - floor(Abs(x_i)) \times 10^{14}, 256) \tag{7}$$

where $Abs(x)$ returns the absolute value of x. $Floor(x)$ returns the value of x to the nearest integers less than or equal to x, $mod(x, y)$returns the remainder after division. As a result, x_1^*, x_2^*, x_3^* and x_4^*, which belong to will be used in generation of RG.

Step 11: Arrange the resulted x_1^* in M rows and $N+1$ columns. As a result, a random grid generated by hyper-chaos will be gotten.

Step 12: Repeat Steps 9 to 11 for every $I_i^{''}, i = 1, 2, \cdots, m$, m random grids will be obtained. They are called $R_i, i = 1, 2, \cdots, m$, with the size of $M \times (N+1)$.

Step 13: Generation of sharing secret. For confused image $I_1^{''}$. Firstly, convert the MAC H_1into 32 decimal values labeled by $a_i^1, i = 1, 2, \cdots, 32$. In the same way, $H_i^{'}, i = 1, 2, \cdots, m$ are also converted into

$b_i^1, i = 1, 2, \cdots, 32.$

$$\begin{cases} a_1^1 = Bin2dec(h_1^1 h_2^1, \cdots, h_8^1) \\ a_2^1 = Bin2dec(h_9^1 h_{10}^1, \cdots, h_{16}^1) \\ \cdots\cdots \\ a_{32}^1 = Bin2dec(h_{249}^1 h_{250}^1, \cdots, h_{256}^1) \\ b_1^1 = Bin2dec(h_1^{I'} h_2^{I'}, \cdots, h_8^{I'}) \\ b_2^1 = Bin2dec(h_9^{I'} h_{10}^{I'}, \cdots, h_{16}^{I'}) \\ \cdots\cdots \\ b_{32}^1 = Bin2dec(h_{249}^{I'} h_{250}^{I'}, \cdots, h_{56}^{I'}) \end{cases} \quad (8)$$

where as, $h_i^1, i = 1, 2, \cdots, 256$ represents 256 bits of H_1, and $h_i^{I'}, i = 1, 2, \cdots, 256$ is the 256 bits of H_1'. Obviously, $a_i^1, b_i^1 i = 1, 2, \cdots, 32$ all are in the scope of $[0, 255]$. Then, put them into the last position of the last column of R_1. Lastly, carry out the exclusive or operation between the ahead N columns of R_1 and I_1'', thus get the shared images E_1.

In the same way, for other confused images $I_i'', i = 2, 3, \cdots, m$, the sharing secret $E_i'', i = 2, 3, \cdots, m$ can be derived. all are with the size of $M \times (N + 1)$.

The detail flowchart of the generation of sharing secret is shown in Figure 4.

3.2 Process of Secret Image Restoration

For every shared image $E_i, i = 1, 2, \cdots, m$, the two groups MAC can be collected from the last column, then use the same way as that in the generation stage of sharing image, we can obtain random grid $R_i', i = 1, 2, \cdots, m$, then, the following steps are executed in order to restore the secret images.

Step 1: Implement the exclusive or operation between the ahead N columns of generated R_i' and $E_i, i = 1, 2, \cdots, m$, then, image $U_i'', i = 1, 2, \cdots, m$ with the size of $M \times N$ are produced.

Step 2: Transform $U_i'', i = 1, 2, \cdots, m$ into the $U_i', i = 1, 2, \cdots, m$ in the reverse order with the Step 7 of generation stage of sharing.

Step 3: Use the MAC to execute Steps 2-4 in the generation stage of sharing secret to perform decryption for every $U_i', i = 1, 2, \cdots, m$, then the original secret images $I_i, i = 1, 2, \cdots, m$ are restored.

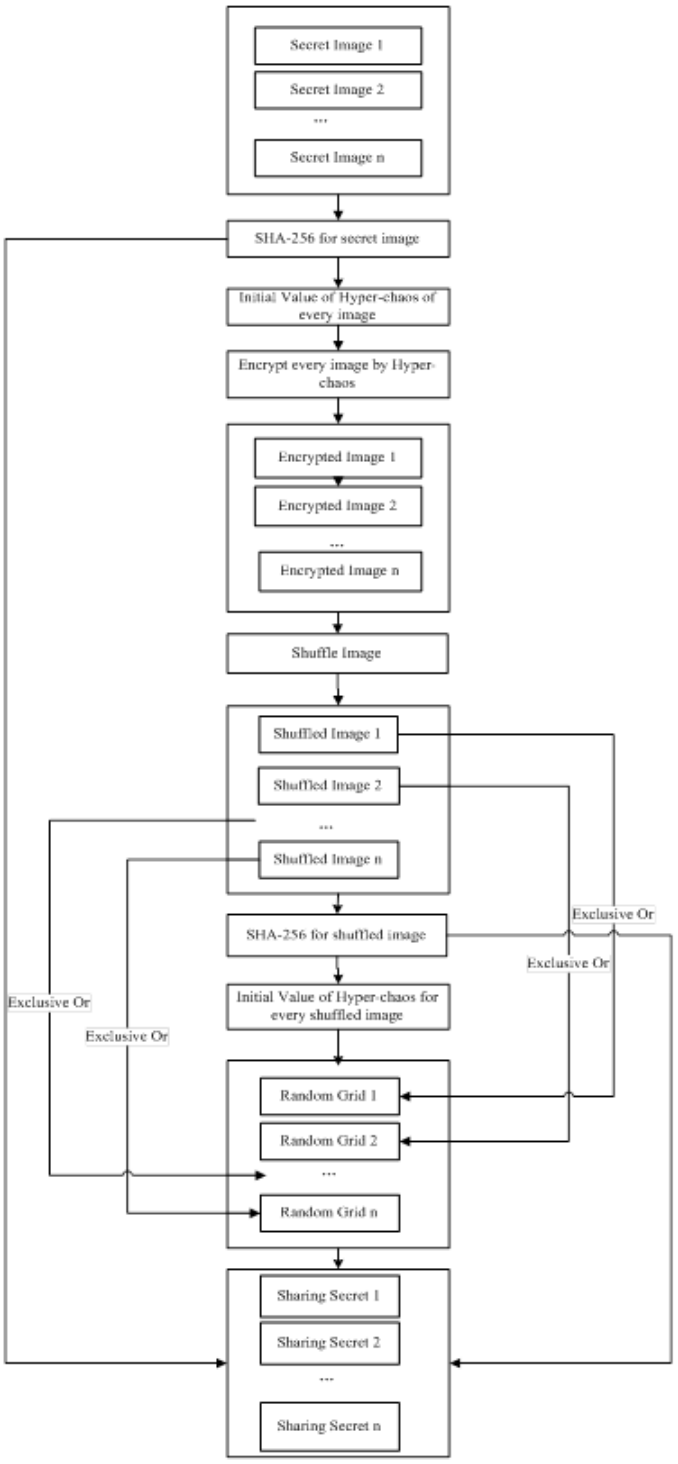


Figure 4: The flowchart of the generation of sharing secret

4 Experimental Results and Discussions

The experiment was done by Mathworks MATLAB version 12b. Some grey images such as "Lenna, Bird, Aerial and Camera" with the size of 256×256 are used for secret image; they are shown in Figure 5.

4.1 Experimental Results

Firstly, the secret images are calculated by MAC, and then MAC generated by the secret images is given in Table 1, respectively.

The initial values of hyper-chaos converted from the

Table 1: The MAC of original secret image

Secret Image	**MAC**
Lenna	CD77462213222E005C31767595E33417FB1C78DB8570837C769E8A7AB4E754F8
Bird	A0C7B83460A68AE1DB97FFD9F98E0CB993FFAAACAB595012DA784833365D65F7
Aerial	05B44DC1F9BEE57B656651A62E2A7389D8774AF93FF133A7E9C11D0AF8F67814
Camera	CB629BACC15E52D202567C47421D0368B92FE25B475FFABFFE9FBEB1C9415A17

above MAC are as follows:

$$\begin{cases} x_1(0) = 0.07364 \\ x_2(0) = 0.49910 \\ x_3(0) = 0.46934 \\ x_4(0) = 0.52627 \end{cases}$$

$$\begin{cases} x_1(0) = 0.13105 \\ x_2(0) = 0.42354 \\ x_3(0) = 0.01068 \\ x_4(0) = 0.46013 \end{cases}$$

$$\begin{cases} x_1(0) = 0.14164 \\ x_2(0) = 0.15432 \\ x_3(0) = 0.01257 \\ x_4(0) = 0.49200 \end{cases}$$

$$\begin{cases} x_1(0) = 0.36484 \\ x_2(0) = 0.47873 \\ x_3(0) = 0.03421 \\ x_4(0) = 0.61714 \end{cases} \quad (9)$$

So, the encrypted image and the final sharing secret image can be got and are shown in Figure 6. Here the size of encrypted image is with 256×256, and the sharing secret image is with the size of 256×257.

4.2 Security of the Proposed Scheme

In the proposed scheme, the security of algorithm lies in two aspects. One is that the hash values of original secret images are used to generate the initial values of the hyper-chaos system and initial values decides the security of encryption. It can be obviously seen from the generation processing of sharing secret, the initial values of hyper-chaos (secret keys) are strongly related to secret images. In the third stage of the scheme, the MAC of shuffled image is used to generate another group initial value of hyper-chaos, which affects the generation of final sharing secret. This is a combination of two one-time pads from the point of encryption, and the two MAC codes all have 512 bits, this makes the secret space reaches, which ensures the security of the scheme.

(a) Lenna (b) Bird

(c) Aerial (d) Camera

Figure 5: The image for test

Another aspects lies in the generation of sharing secret based on hyper-chaos diffusion. One has shown that image encryption algorithm based on hyper-chaotic system with only one round diffusion process has some kind of weak security [32]. The proposed scheme achieves the security through two rounds of diffusion process and one position confusion.

From the description of the proposed scheme, it can be seen that, the sharing secret can be used to restore the original secret image, but the scheme need all the sharing secret images to take part in. From the point of security, once anyone of the sharing secret images is modified, the original image will not be restored. The data position in the sharing secret image is shown in Figure 7.

Obviously, the some original secret images will not be completely restored if the "Sharing image data" is tampered with from the scheme. Even if only one bit are modified, some original secret images will not restored completely.

For example, the value of the 30^{th} position in Sharing image data in sharing secret 1 is 45. If it is modified to be 46, other data in all sharing secrets are all kept intact,

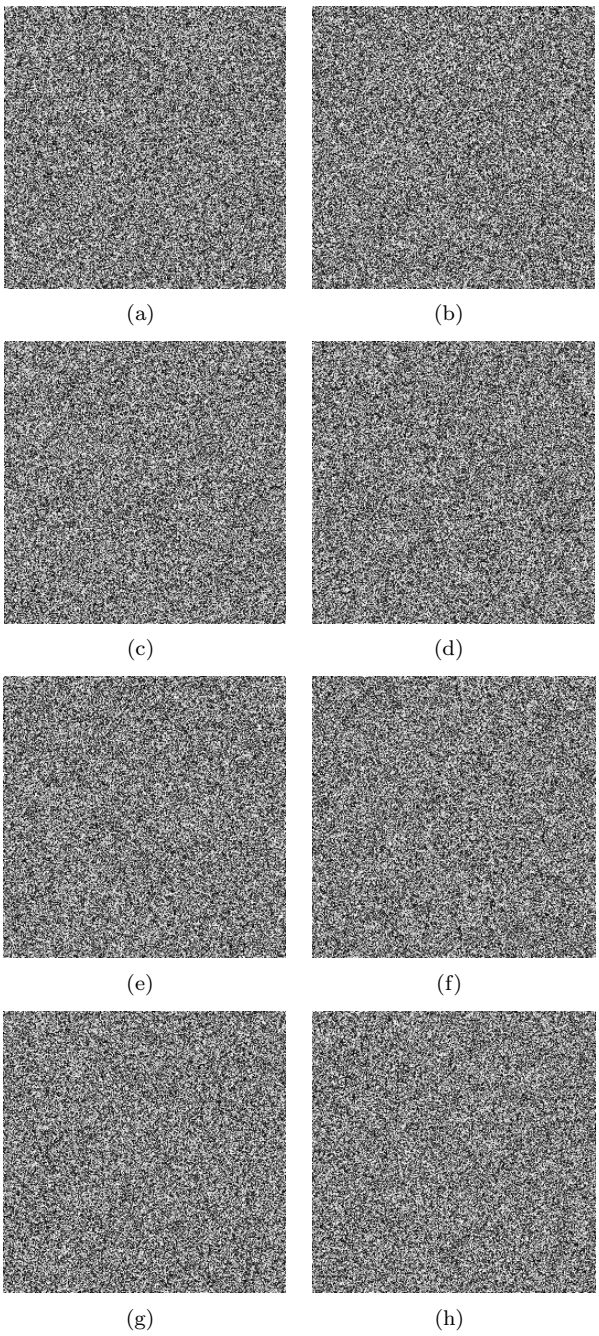


Figure 6: The encrypted and sharing secret images, (a)-(d) the encrypted image (e)-(h) the sharing secret

then it is found that the image "Aerial" is not lossless restored when we restore the original secret image, as the MAC of restored "Aerial" is

"1C3132BE45A60A4E544F39E29803B03A477C6
C4B2D9F0557AD217C6168BD3DA0"

It is totally different from that in table 1. Although the image has no distinct changes from the visual effect, it is indeed altered from the computation, so it is not lossless restore for the algorithm in real sense.

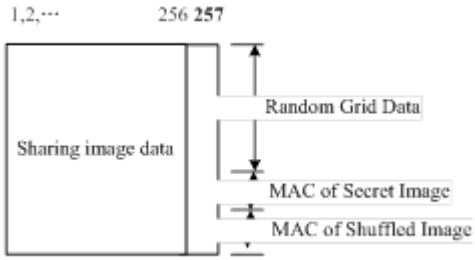


Figure 7: Data position information in sharing secret

"MAC of Shuffled Image" affects the generation of random grid, if it is falsified, the sharing secret will become useless, because if it is modified, the initial values of hyper-chaos will be altered, thus the random grid generated by hyper-chaos will be different from the original one, this will result in the mistakenly restoration of original secret image. In the same sense, "MAC of Secret Image" affects the decryption of original secret image, if it is falsified, the original secret image will not be correctly decrypted. "Random Grid Data" in the sharing secret is redundant; it can be replaced by some valuables, which will be studied in future works.

For example, for sharing secret 1, the data of "MAC of Secret Image" is "205, 119, 70, 34, 19, 34, 46, 0, 92, 49, 118, 117, 149, 227, 52, 23, 251, 28, 120, 219, 133, 112, 131, 124, 118, 158, 138, 122, 180, 231, 84, 248", if the first number "205" is modified to be "206", other data in sharing secret 1 and other sharing secret are all not modified. In this situation, the four shuffled images are all correctly restored, and when restore the original secret image, the initial values of hyper-chaos for the first group become:

$$\begin{cases} x_1(0) = 0.08132 \\ x_2(0) = 0.49910 \\ x_3(0) = 0.46934 \\ x_4(0) = 0.52627 \end{cases} \tag{10}$$

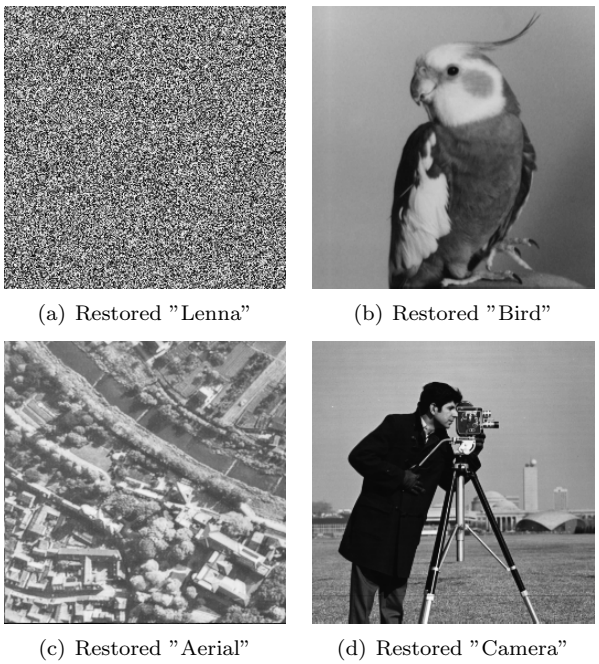
It is obviously different from the original initial value in Equation (7), so we get the restored secret image, which are shown in Figure 8.

4.3 Double Verification

It can be seen from the generation of sharing secret, the proposed scheme has verification of two stages. One is the integrity verification of sharing secret image; the other is the integrity verification of restored original secret image.

For anyone of sharing images, it is with the size of 256×257, in order to verify the integrity of sharing secret image; we can perform the following steps:

1) Convert the "MAC of Shuffled Image" into initial value of hyper-chaos, and generate the random grid R.



(a) Restored "Lenna" (b) Restored "Bird"

(c) Restored "Aerial" (d) Restored "Camera"

Figure 8: The recovered secret images when sharing secret is modified

2) Carry out the exclusive or operation between R and the sharing secret image, and the resulting image is represented by $I^{''}$.

3) Calculate the MAC of $I^{''}$, and compared it with "MAC of Shuffled Image", if they are identical, it proves the "MAC of Shuffled Image" is intact; else we can think the "MAC of Shuffled Image" is modified.

Of course, if the "Sharing image data" are modified, then the MAC of $I^{''}$ and that of "MAC of Shuffled Image" must be different from the properties of MAC. Similarly, we can also judge whether the Sharing secret is tampered with.

For the second verification stage, assume that the restored original image is I^s, then, calculate the MAC of the restored image, and compare it with that of the "MAC of Secret Image", thus we can also judge the integrity of the restored secret image.

The necessity of the second verification can be explained by the following example.

Assume the four sharing secret image are $I_i^r, i = 1, 2, 3, 4$, and the 20th column data in is tampered with, and other any data in are all not modified.

From the processing of sharing image generation and decryption, it can be concluded that

1) The first sharing secret image is modified through the verification of "MAC of Shuffled Image", and the other three sharing secret images are intact.

2) For the four shuffled images, only the first shuffled image is modified.

3) Convert to original secret image from the shuffled image, only the fourth image is altered, and the other three are lossless.

To clearly explain the verification process, two examples are given in the following. The example1 illustrates the verification of sharing secret, another one is for verification of restored secret image.

Example 1. *For sharing image 2, the 32 decimal values corresponding to "MAC of Shuffled Image" are "231, 30, 203, 231, 35, 111, 191, 127, 186, 197, 47, 240, 131, 114, 102, 156, 1, 136, 108, 248, 15, 215, 160, 183, 15, 170, 183, 44, 127, 237, 110, 231". If the last two numbers are modified into "111, 230" from "110 230", then, the original initial value of hyper-chaos and that of modified one will become the following, respectively:*

$$\begin{cases} x_1(0) = 0.21497 \\ x_2(0) = 0.16323 \\ x_3(0) = 0.53287 \\ x_4(0) = 0.06067 \end{cases}$$

$$\begin{cases} x_1(0) = 0.21497 \\ x_2(0) = 0.16323 \\ x_3(0) = 0.53287 \\ x_4(0) = 0.05810 \end{cases} \tag{11}$$

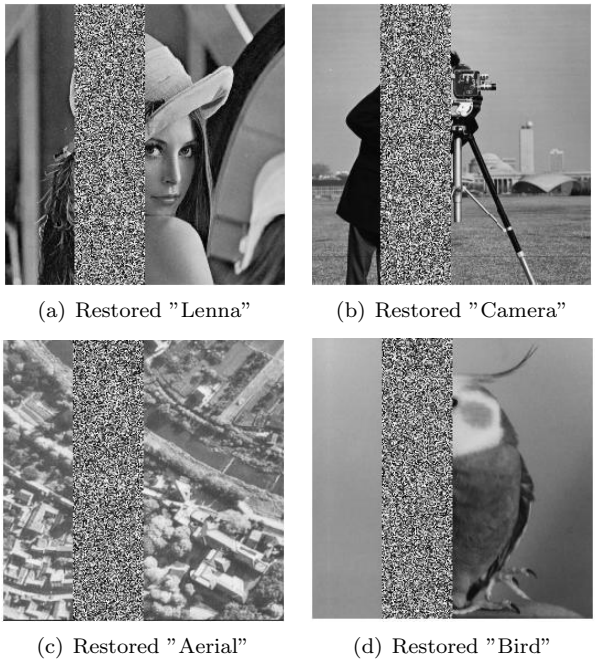
Thus, the random grid generated by hyper-chaos with the initial values of modified one is different from that generated by original initial values. In this case, the restored secret images are shown in Figure 9. it can also be verified that the restored images are damaging.

Example 2. *In order to testify the necessity of integrity verification of the restored secret image, it is assumed that the first sharing image is inserted by two white lines, such as shown in Figure 10. Then, the decryption process is used to retrieve the secret images; the restored secret images are shown in Figure 10. It can be concluded that from the verification process that, the image "Lenna" and "Camera" are lossless restored, because the MAC of the two restored image and that in the sharing images are the same, but the MAC of restored "Bird" and "Aerial " are:*

"$E3D53BB3C4324FA6E023F4CDEE4A7F4CD956$
$CBB36E9523403B83C436049552D4$"

"$F8F7944EFE837387F9C56C0CB8EEA19077337$
$09115D0FDAA23C6505B8D31ADC0$"

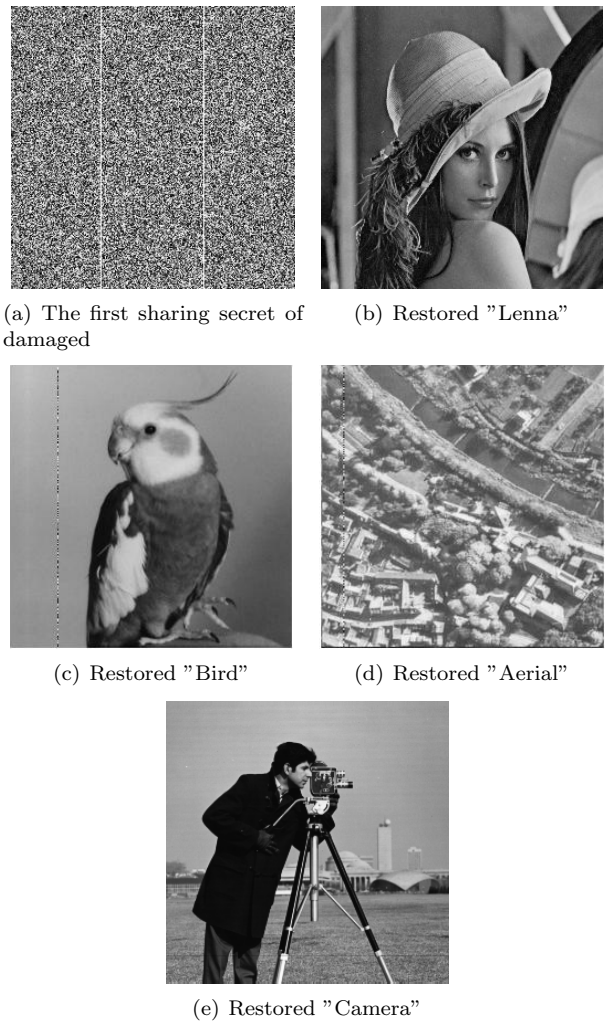
Obviously, they are different from the MAC of secret images (Table 1), so, it is regarded that the image "Bird" and "Aerial" are damaging restored.



(a) Restored "Lenna" (b) Restored "Camera"

(c) Restored "Aerial" (d) Restored "Bird"

Figure 9: Restored secret image when the second sharing secret is modified



(a) The first sharing secret of damaged (b) Restored "Lenna"

(c) Restored "Bird" (d) Restored "Aerial"

(e) Restored "Camera"

Figure 10: Restored secret image when the first sharing secret is modified

4.4 Comparison with Existing Algorithms

It can be seen from the description of the scheme, the final sharing secret is with the size of $M \times (N+1)$, the pixel expansion is $1+\frac{1}{NM}$. Obviously, the pixel expansion is mini with the larger M and N.

Table 2 gives some comparisons with existing algorithms in application type of images, size of sharing secret, property of restored secret image and verification of the scheme.

Apparently, the proposed scheme has the following advantages when it is compared with some published literatures.

1) The generated sharing secrets have only mini expansions than original secret image, if the original secret image is with the size of $M \times N$, then the expansion is only $\frac{1}{NM}$ compared with the original secret image.

2) The sharing secret is generated by One-Time Pad, which is information-theoretically secure in that the encrypted message provides no information about the original message to a cryptanalyst (except the length of the message).

3) The proposed algorithm can not only verify the integrity of the sharing secret, it can also verify the integrity of the restored secret image. If the sharing image is not modified, the restored stored image is completely the same as the original one, parts of the sharing images are modified may not means that the original secret images are lossless restored or not, it can be testified through the verification process.

4) Every right sharing secret can help to restore one part of original secret image, the more sharing secret takes part in, the better that original secret image restore, if all the sharing secret are used, the secret image can be restored completely. So the proposed scheme can progressively recover the original secret image.

5 Conclusions

A novel multi secret sharing scheme is proposed in this paper. Generation of sharing secret includes three stages in the scheme. The first stage is encryption of secret image based on hyper-chaos, the encryption algorithm rely on the secret image self. The second stage is shuffling of encryption. The third stage is generation of sharing secret. In this stage, MAC of shuffled encryption image are used for initial values of hyper-chaos, then the hyper-chaos is used to generate random grid, finally, the sharing secret is generated through XOR operation between random grid

Table 2: Some comparisons with existing scheme

The scheme	Type of secret image	Pixel expansion	Recovered image	Type of VSS	Verification
Shyu's method (2009)	B, G, C	N	Lossy	RG-based	No
Chen's method (2011)	B	N	Lossy	RG-based	No
Wu's method (2012)	B	N	Lossy	RG-based	No
Chen's method (2013)	G	N	Lossless	RG-based	No
Lin's method (2015)	B, G, C	N	Lossy	RG-based	No
Ours method	G	Mini	Lossless	RG-based	Yes

and confused image. Large numbers of experiments show that the proposed scheme can lossless restore the original secret image, and have the double verification ability for sharing secret and restored secret image. The above advantages make it especially suitable for secret sharing of important images such as medical and military images.

Acknowledgments

The work was partly supported by the Program of National Science Fund of Tianjin, China (Grant NO. 16JCYBJC15700). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] A. A. AbdEl-Latif, X. H. Yan, L. Li, et al., "A new meaningful secret sharing scheme based on random grids, error diffusion and chaotic encryption," *Optics and Laser Technology*, vol. 54, pp. 389–400, 2013.

[2] G. R. Blakley, "Safe guarding cryptographic keys," in *Proceedings of the National Computer Conference*, pp. 313–317, 1979.

[3] T. H. Chen, K. H. Tsao, "Threshold visual secret sharing by random grids," *Journal of Systems and Software*, vol. 84, pp. 1197–1208, 2011.

[4] T. H. Chen, K. H. Tsao, "User-friendly random-grid-based visual secret sharing," *IEEE Transactions on Circuit Syst. VideoTech*, vol. 21, no. 11, pp. 1693–1703, 2011.

[5] W. K. Chen, "Image sharing method for gray-level images," *The Journal of Systems and Software*, vol. 86, pp. 581–585, 2013.

[6] Y. C. Chen, D. S. Tsai, R. B. Horng, "Visual secret sharing with cheating prevention revisited," *Digital Signal Processing*, vol. 23, pp. 1496–1504, 2013.

[7] Y. C. Chen, D. S. Tsai, R. B. Horng, "A new authentication based cheating prevention scheme in Naor-Shamir's visual cryptography," *Journal of Visual Communication and Image Representation*, vol. 23, no. 8, pp. 1225–1233, 2012.

[8] Y. C. Chen, D. S. Tsai, R. B. Horng, "Visual secret sharing with cheating prevention revisited," *Digital Signal Process*, vol. 23, pp. 1496–1504, 2013.

[9] W. P. Fang, J. C. Lin, "Progressive viewing and sharing of sensitive images," *Pattern Recognition and Image Analysis*, vol. 16, no. 4, pp. 632–636, 2006.

[10] T. Gao, Z. Chen, et al. "A hyper-chaos generated from Chen's system," *International Journal of Modern Physics C*, vol. 17, pp. 471–478, 2006.

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

[12] G. Horng, T. H. Chen, D. S. Tsai, "Cheating in visual cryptography," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 219–236, 2006.

[13] Y. C. Hou, Z. Y. Quan, C. F. Tsai, et al., "Block-based progressive visual secret sharing," *Information Sciences*, vol. 233, no. 1, pp. 290–304, 2013.

[14] F. J. Jeng, W. L. Huang, T. H. Chen, "Cryptanalysis and improvement of two hyper-chaos-based image encryption schemes," *Signal Processing: Image Communication*, vol. 34, pp. 45–51, 2015.

[15] O. Kafri, E. Keren, "Encryption of pictures and shapes by random grids," *Optics Letters*, vol. 12, no. 6, pp. 377–379, 1987.

[16] Y. S. Lee, T. H. Chen, "Insight into collusion attacks in random-grid-based visual secret sharing," *Signal Processing*, vol. 92, pp. 727–736, 2013.

[17] C. Q. Li, Y. S. Liu, T. Xie, et al., "Breaking a novel image encryption scheme based on improved hyperchaotic sequences," *Nonlinear Dynamics*, vol. 73, pp. 2083–2089, 2013.

[18] C. H. Lin, T. H. Chen, Y. T. Wu, et al, "Multi-factor cheating prevention in visual secret sharing by hybrid codebooks," *Journal of Visual Communication and Image Representation*, vol. 25, pp. 1453–1557, 2014.

[19] P. Y. Lin, R. Z. Wang, Y. J. Chang, et al., "Prevention of cheating in visual cryptography by using coherent patterns," *Information Sciences*, vol. 301, pp. 61–74, 2015.

[20] Y. J. Liu, C. C. Chang, "An integratable verifiable secret sharing mechanism," *International Journal of Network Security*, vol. 18, no. 4, pp. 617–624, 2016.

[21] M. Naor, A. Shamir, "Visual cryptography," in *Advances in cryptography (Eurocrypt'95)*, pp. 1–12, 1995.
[22] NIST, *Announcing The Secure Hash Standard*, Federal Information Processing Standards Publication 180-2, U.S. DoC/NIST, Aug. 2002.
[23] B. Norouzi, S. Mirzakuchaki, S. M. Seyedzadeh, et al., "A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1469–1497, 2014.
[24] A. Shamir, "How to share a secret," *Communications of the ACM* , pp. 612–613, 1979.
[25] S. J. Shyu, "Image encryption by multiple random grids," *Pattern Recognition*, vol. 42, pp. 1582–1596, 2009.
[26] D. S. Tsai, T. H. Chen, G. Horng, "On generating meaningful shares in visual secret sharing scheme," *Image Science Journal*, vol. 56, pp. 49–55, 2008.
[27] X. T. Wu, W. Sun, "Random grid-based visual secret sharing for general access structures with cheat-preventing ability," *The Journal of Systems and Software*, vol. 85, pp. 1119–1134, 2012.
[28] X. Yan, S. Wang, X. Niu, et al., "Generalized random grids-based threshold visual cryptography with meaningful shares," *Signal Processing*, vol. 109, pp. 317–333, 2015.
[29] S. Yanchuk, T. Kapitaniak, "Symmetry-increasing bifurcation as a predictor of a chaos-hyperchaos transition in coupled systems," *Physical Review E*, vol. 64, pp. 056235, 2001.
[30] Y. S. Zhang, D. Xiao, W. Y. Wen, et al., "Breaking an image encryption algorithm based on hyper-chaotic system with only one round diffusion process," *Nonlinear Dynamics*, vol. 76, no. 3, pp. 1645–1650, 2014.
[31] Z. Zhou, G. R. Arce, G. D. Crescenzo, "Halftone visual cryptography," *IEEE Transactions on Image Process*, vol. 15, no. 8, pp. 2441–2453, 2006.
[32] C. Zhu, "A novel image encryption scheme based on improved hyperchaotic sequences," *Optical Commun*, vol. 285, no. 1, pp. 29–37, 2012.
[33] H. Zhu, Y. Zhang, Y. Zhang, "A provably password authenticated key exchange scheme based on chaotic maps in different realm," *International Journal of Network Security*, vol. 18, no. 4, pp. 688–698, 2016.
[34] X. Zhuang, C. C. Chang, Z. H. Wang, et al., "Simple password authentication scheme based on geometric hashing function," *International Journal of Network Security*, vol. 16, no. 4, pp. 271–277, 2014.

Biography

Hang Gao was born in Tianjin City, China, in 1992. He received the B. S. degree in Software Engineering from University of Electronics Science and Technology of China, Chengdu, China, in 2015. He is currently working toward the M. S. degree in Software Engineering from Nankai University, Tianjin, China. His research interests include information security and cloud computing.

Mengting Hu was born in Shanxi Province, China, in 1993. She received the B. S. degree in Software Engineering from Tongji University, Shanghai, China, in 2015. She is currently working toward the M. S. degree in Software Engineering from Nankai University, Tianjin, China. Her research interests include cloud computing and Information Retrieval.

Tiegang Gao received Ph. D degree from Nankai University, Tianjin, China in 2005. He is a professor in college of software, Nankai University, China since 2006. His research interests include cloud computing and information security, he has published or co-authored more than 100 papers in related field.

Renhong Cheng received Ph. D degree from Nankai University, Tianjin, China. Now He is a professor in college of computer and control engineering, Nankai University, His research interests include database technique and Information Retrieval.