Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 17(4): 4295–4316.
DOI: 10.3934/mbe.2020237
Received: 22 April 2020
Accepted: 15 June 2020
Published: 18 June 2020

Research article

A compressed secret image sharing method with shadow image verification capability

Guozheng Yang[1,2,*], **Lintao Liu**[1,2] **and Xuehu Yan**[1,2]

[1] National University of Defense Technology, Heifei 230037, China

[2] Anhui Key Laboratory of Cyberspace Security Situation Awareness and Evaluation, Heifei 230037, China

* **Correspondence:** Email: yangguoz0218@163.com.

Abstract: Secret image sharing (SIS) is an important research direction in information hiding and data security transmission. Since the generated shadow images (shares) are always noise-like, it is difficult to distinguish the fake share from the unauthorized participant before recovery. Even more serious is that an attacker with a fake share can easily collect shares of other honest participants. As a result, it is significant to verify the shares, before being taken out for recovery. Based on two mainstream methods of SIS, such as polynomial-based SIS and visual secret sharing(VSS), this paper proposed a novel compressed SIS with the ability of shadow image verification. Considering that the randomness of the sharing phase of polynomial-based SIS can be utilized, one out of shares of $(2, 2)$-threshold random-grid VSS is embedded into all shares of polynomial-based SIS by a XOR operation as the verification information, while the other binary share is private for verification. Before recovery, each participant must extract the binary share from the grayscale share to perform XOR operation with the private share, and the original binary image can be recovered only with the true share. The proposed scheme also has the characteristics of shadow image verification, pixel compression, loss tolerance and lossless recovery. Through experiments and comparative analysis of related research results, the effectiveness and advantages of the method are verified.

Keywords: secret image sharing; shadow verification; pixel compression; loss tolerance; lossless recovery

1. Introduction

Since Shamir proposed (k, n)-threshold secret sharing technology [1] in 1979, this scheme has become the basis of other secret sharing schemes. In 1995, Naor and Shamir [2] extended secret sharing technology to image field and proposed the concept of visual cryptography: visual secret sharing

(VSS). In 2002, Thien and Lin [3] proposed a secret image sharing (SIS) scheme based on (k, n)-threshold, and formally introduced Shamir's polynomial-based secret sharing method into the field of image encryption. Due to the unique advantages of image in information hiding and data processing, SIS has gradually developed into a special branch in the field of secret sharing.

SIS is a multi-channel secret information distribution technology [4]. In the process of SIS with (k, n)-threshold $(2 < k < n)$, the secret image is generated into n shadow images (also known as shadows or shares) by using method which ensures that insufficient shadow images can not recover the original secret image. When recovery is needed, k or more shadow images must participate in the process of effective recovery. Therefore, a SIS always consists of two phases: shadow images generation (secret image sharing) and secret image recovery. The measurement characteristics mainly include pixel expansion or compression, recovery loss or lossless, calculation efficiency high or low, and so on. Compared with the traditional image encryption and image hiding methods, the SIS method has the advantages of higher information security, lower computational complexity and loss tolerance. Therefore, the research on SIS has become a very popular direction in recent years.

At present, according to the different strategies of secret image sharing and recovery, SIS is mainly divided into two categories: polynomial-based scheme [5] and VSS [6, 7].

(k, n)-threshold polynomial-based scheme depends on a $k - 1$ degree polynomial in an integer field, in which k coefficients are embedded with a secret integer and $k - 1$ random integers. n values are computed as shared values with n different non-zero independent variables. In the recovery phase, the coefficients of polynomial can be calculated by Lagrange interpolation using at least k shares, so as to recover the secret information. This kind of strategy is flexible in setting (k, n) threshold, and the visual quality of the recovered image is high, but it has higher computational complexity for recovery, $O(k \log^2 k)$.

The (k, n)-threshold VSS-based scheme can directly decrypt the secret information just by superposition or simple logic operation with enough shares. This kind of methods need low calculation for recovery, but may have some problems, such as pixel expansion, poor visual quality of recovered image and the limited (k, n) threshold. The random-grid VSS [8–10] scheme has been studied on these problems such as improving the contrast of restored image and no pixel expansion.

These two kinds of schemes are mainly studied from the mechanism of how to construct SIS. In application field, after obtaining shadow images from multiple channels, how to verify the authenticity and integrity of each shadow image is also a problem to be solved. A fake share is easily forged by the adversary, and has a destructive impact on the recovery of the secret image. Even more serious is that an adversary with a fake share can easily collect shares of other honest participants. As a result, the application of SIS scheme in data transmission is seriously restricted. In this paper, we mainly study the compressed SIS method with the ability of shadow image verification. Based on the current polynomial-based SIS scheme, we add VSS technique to achieve the ability of both secret image lossless recovery and shadow image high-efficiency verification.

The main structure of this paper is provided as follows. Section 2 introduces the related research status, while Section 3 analyzes the research basis. The design concept and implementation of this scheme are provided in Section 4. Section 5 discusses the experimental situation and comparative analysis, and finally the conclusion is drawn in Section 6.

2. Related works

In 1989, Tompa and woll [11] first proposed that there might be fake shadow images in the process of secret sharing, which would not only result in the failure of the reconstruction of the secret image, but also increase the risk of disclosure. In the field of SIS, the verification problem of shadow images has attracted more and more attention.

The traditional method to check the integrity of shadow image is to use hash function [12] and information hiding [13], calculate the hash value of the generated shadow image and hide this value by using information hiding technologies. Before the recovery of the secret image, each participant must compute the hash value of shares from others, and then compare the result with the hidden hash value to confirm whether the shadow image has been tampered or forged. Liu and Chang [14] proposed a VSS scheme with reversibility and verification by setting up turtle shell reference matrix. Liu et al. [15] studied the shadow image forgery problem by introducing the results of two polynomial of random value association, based on Thien and Lin's polynomial-based scheme. These schemes have more time cost on the verification phase rather than the sharing and recovery phases. Moreover, the share privately hold by each participant should be handed out for verification. Therefore, the private share is still disclosed, even though the verification fails. In addition, these scheme cannot achieve precise recovery.

Yan et al. [16] proposed an SIS with separate shadow verification ability, which is a combination of polynomial-based SIS and random-grid VSS. In their scheme, a binary secret image used for verification is shared by $(2, 2)$-threshold random-grid VSS. Then, the highest bit of pixel of each grayscale shadow images is required to be the same as 1-bit pixel value of one of binary shares. Before recovery, the binary share can be extracted from the grayscale shares and then stacked with the other binary share privately hold by a trusted third party: if the original binary secret image is reconstructed, the grayscale share is valid for recovery, and vice versa. Since 1-bit verification information is fixedly embedded into the highest bit of the pixel of each grayscale shares, n shares have the identical highest bit plane. However, if information of the highest bit plane is leaked, the adversary can easily forge a authenticated grayscale share.

Inspired by Yan et al.'s scheme, we proposed a novel compressed SIS with shadow image verification capability, which is also a combination of polynomial-based SIS and random-grid VSS. In the proposed scheme, the size of each grayscale shadow image is reduced to half of the original secret image; and the verification information is hidden into two lowest bits of the grayscale pixel. As a result, the special design improves the efficiency and security. In addition, an improved version of polynomial-based SIS is utilized to achieve precisely recovery.

The specific idea of the proposed scheme is stated as follows. The framework of polynomial-based scheme is selected, which is built on the preciseness of mathematical calculation with the main characteristics of (k, n) threshold and lossless recovery; then a VSS scheme with simple stacking-to-see decryption is used for the verification of shadow images; finally, the above two schemes are fused with experimental verification to achieve the goal of this paper.

3. Preliminary

3.1. Polynomial-based SIS

In a polynomial-based SIS scheme for (k, n) threshold, $k-1$ degree polynomial over a prime field P is usually used to share and recover secret information. The form of the polynomial is in formula 3.1.

$$f(x) = (a_0 + a_1 x + a_2 x^2 + ... + a_{k-1} x^{k-1}) \bmod P \tag{3.1}$$

In Shamir's scheme [1], one secret pixel is assigned to a_0 of the polynomial each time, and other coefficients of the polynomial are obtained by random values in $[0, P-1]$. Because the value of P is 251, but the maximum pixel value of grayscale image is 255, when the pixel value of image exceeds 251, this part of pixel cannot be recovered correctly and lossless recovery cannot be realized. On the basis of Shamir's scheme, in order to reduce the size of shadow image, Thien and Lin's scheme [3] embeds k secret pixel values into the coefficients a_0 to a_{k-1}, and the specific sharing and recovery methods are given in algorithms.

Thien and Lin's sharing algorithm
Input: grayscale secret image S_1, with size of (H, W), threshold (k, n)
Output: n shadow images $\{S_1C_i, 1 \le i \le n\}$, with size of $(H, W/k)$
Algorithm: 1. Set the prime number $P = 251$; 2. For $s_1(h, w) \in \{S_1(h, w) \mid 1 \le h \le H, 1 \le w \le W\}$,if $s_1(h, w) \ge 251$, set $s_1(h, w) = 250$ 3. Encrypt S_1 with key to get S_1' 4. Select k pixels in S_1' in order, denoted by $b_0, b_1, ..., b_{k-1}$ 5. Construct $k-1$ degree polynomial: $f(x) = (b_0 + b_1 x + ... + b_{k-1} x^{k-1}) \bmod P$ 6. Calculate $s_1c_1 = f(1), s_1c_2 = f(2), ..., s_1c_n = f(n)$ 7. Assign $sc_1, sc_2, ..., sc_n$ to $S_1C_1(h, w), S_1C_2(h, w), ..., S_1C_n(h, w)$, 8. If there are still pixels in S_1' , go to 4 9. Generate n shadow images $S_1C_1, S_1C_2, ..., S_1C_n$

The original Thien and Lin's scheme [3] inherits the lossy recovery issue of Shamir's scheme. At the same time, because k secret pixels are embedded each time, although the size of the shadow image will be reduced to $1/k$ times of the original image, the randomness is missing. If no additional information encryption method is used, the more serious image information disclosure problem will be caused. Figure 1 shows an example of building (3,4) threshold image sharing without additional encryption in Thien and Lin's scheme.

3.2. Random-grid VSS

Random-grid VSS [17] scheme is a VSS scheme based on probability, mainly for binary images. Because this scheme has no pixel expansion and can achieve rapid recovery through logical "OR" and "XOR" operation, many improved methods based on this scheme have emerged in recent years [18,19].

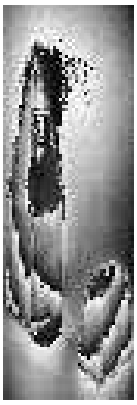
Suppose that 1 represents black pixel and 0 represents white pixel. For the (2,2) threshold, the sharing and recovery methods based on the random-grid VSS scheme are given in algorithms.

Thien and Lin's recovery algorithm
Input: any k shadow images $SC_{i_1}, SC_{i_2}, ..., SC_{i_k}, i_k \in (1, n)$, with size of $(H, W/k)$
Output: recovered grayscale secret image S_{11}, with size of (H, W)
Algorithm: 1. Set $P = 251$ 2. Take out the pixels from the k shadow images in turn $SC_{i_1}(h, j), SC_{i_2}(h, j), ..., SC_{i_k}(h, j)(1 \le h \le H, 1 \le j \le W/k)$ 3. Set $f(i_1) = SC_{i_1}(h, j), f(i_2) = SC_{i_2}(h, j), ..., f(i_k) = SC_{i_k}(h, j)$. Lagrange interpolation is used to solve the following equations: $f(i_1) = (a_0 + a_1 i_1 + ... + a_{k-1} {i_1}^{k-1}) \bmod P$ $f(i_2) = (a_0 + a_1 i_2 + ... + a_{k-1} {i_2}^{k-1}) \bmod P$ ... $f(i_k) = (a_0 + a_1 i_k + ... + a_{k-1} {i_k}^{k-1}) \bmod P$ 4. Arrange the calculated coefficients $a_0, a_1, ..., a_{k-1}$ in turn 5. If the shadow image pixels are not analyzed, go to 2 6. Generate the image S'_{11} according to the combination of all coefficient sequences calculated in step 4 7. Use key to decrypt S'_{11} and restore the image S_{11}

Sharing algorithm of (2,2) threshold random-grid VSS
Input: binary image S_2, with size of (H, W), threshold (2,2)
Output: 2 shadow images S_2C_1,S_2C_2, with size of (H, W)
Algorithm: 1. Randomly select 0 or 1 to generate each pixel $S_2C_1(h, w), 1 \le h \le H, 1 \le w \le W$ of the shadow image S_2C_1 2. Cycle to read pixels $S_2(h, w), (1 \le h \le H, 1 \le w \le W)$ of S_2 3. If $S_2(h, w) = 1$, thus $S_2C_2(h, w) = \overline{S_2C_1(h, w)}$ 4. If $S_2(h, w) = 0$, thus $S_2C_2(h, w) = S_2C_1(h, w)$ 5. Generate another shadow image S_2C_2 from $S_2C_2(h, w)$

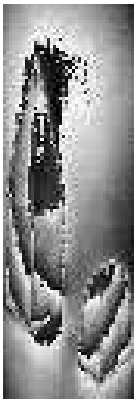
Recovery algorithm of (2,2) threshold random-grid VSS
Input: 2 shadow images S_2C_1,S_2C_2, with size of (H, W)
Output: binary image S'_2,with size of (H, W)
Algorithm: 1. For each pixel $S'_2(h, w), (1 \le h \le H, 1 \le w \le W)$ of S'_2, if stacking recovery (or operation),perform the following formula $S'_2(h, w) = S_2C_1(h, w) \otimes S_2C_2(h, w)$ If lossless recovery (XOR operation), perform the following formula $S'_2(h, w) = S_2C_1(h, w) \oplus S_2C_2(h, w)$ 2. Another shadow image S'_2 will be generated from $S'_2(h, w)$.

(a) Original image

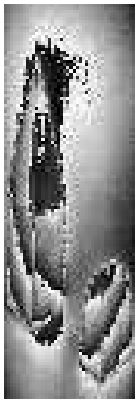


(b) Shadow 1

(c) Shadow 2

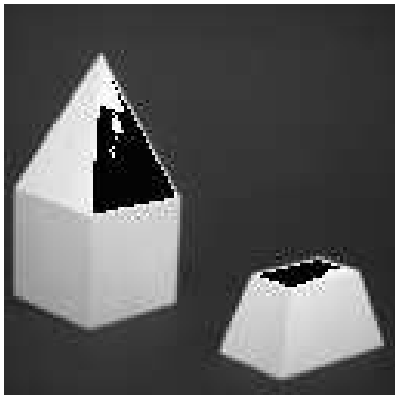


(d) Shadow 3

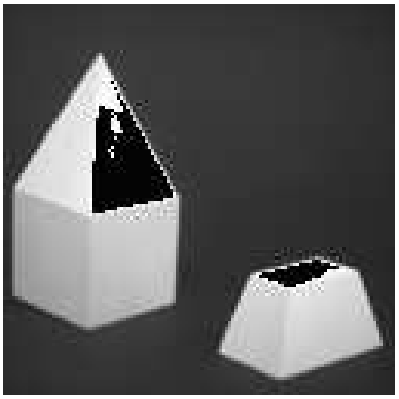


(e) Shadow 4

(f) Decoded secret image S' with 2 shadows



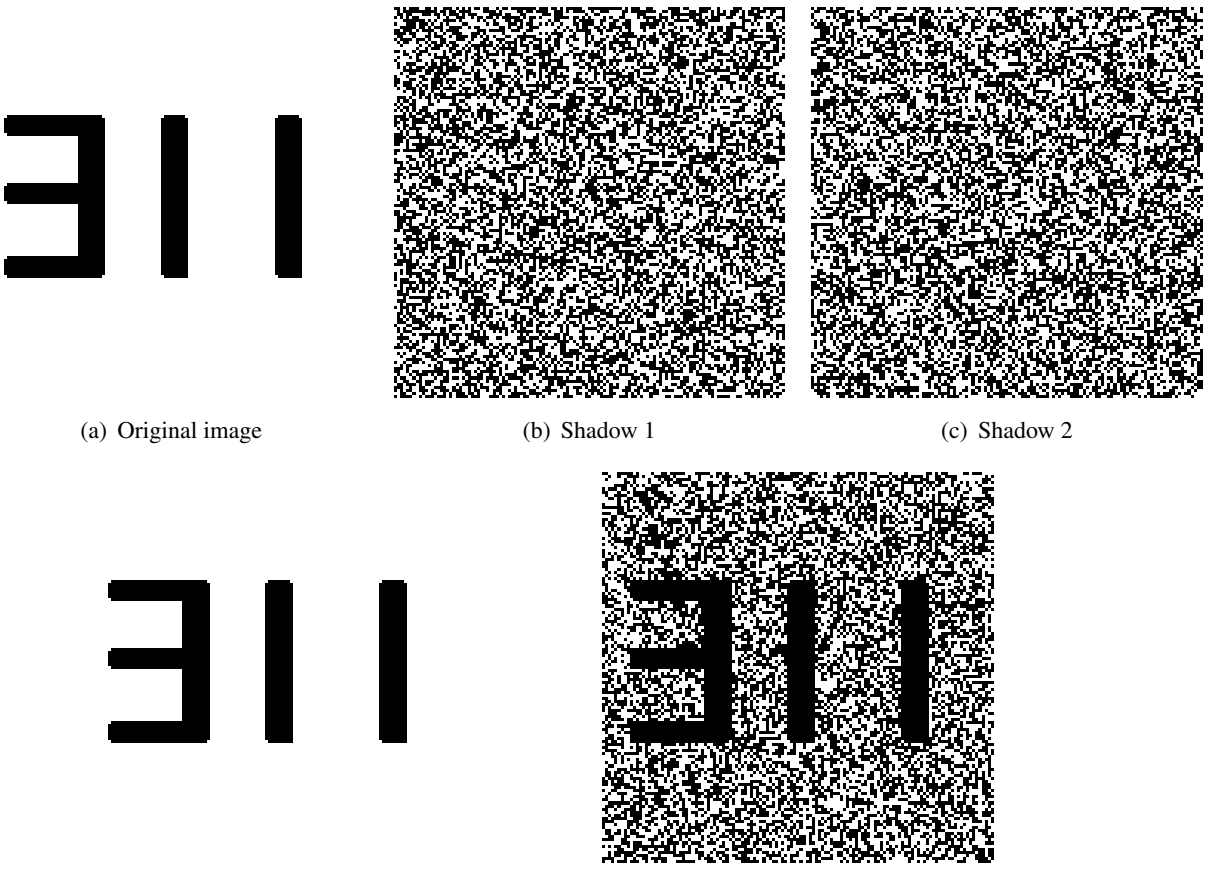
(g) Decoded secret image S' with 3 shadows



(h) Decoded secret image S' with 4 shadows

Figure 1. (3, 4) threshold image sharing and recovery of Thien and Lin's scheme.

It can be seen from the above sharing and recovery methods that, when generating a shadow image, all pixels of one shadow can be randomly generated first, and then another shadow image can be generated according to the original binary secret image. In the recovery phase, logic OR operation or logical XOR operation can be used for recovery. If it is a logical OR operation, it is enough to overlay the shadow image visually. When the original image pixel is 1, the OR operation of the two subpixels must be 1; when the original image pixel is 0, the OR operation of the two subpixels may be 0 or 1. Therefore, OR operation can recover the original image with noisy background, which belongs to lossy recovery. If logical XOR operation is adopted, all pixels can be recovered accurately according to the same rule of 0 and 1, which can achieve the effect of lossless recovery. Figure 2 shows an example of constructing (2, 2) threshold image sharing using the above scheme.



(a) Original image (b) Shadow 1 (c) Shadow 2

(d) "XOR" lossless recovery (e) OR lossy recovery

Figure 2. (2, 2) threshold VSS sharing and recovery.

4. The designed secret image sharing method with shadow verification ability

According to the improvement and fusion of the above two types of schemes, we can construct an SIS scheme with the ability of shadow image verification.

4.1. The main sharing frame

This method improves the polynomial-based SIS scheme described in Section 3.1, and uses the generated verification information in Section 3.2 for lossless embedding. The framework of the proposed method is shown in Figure 3.

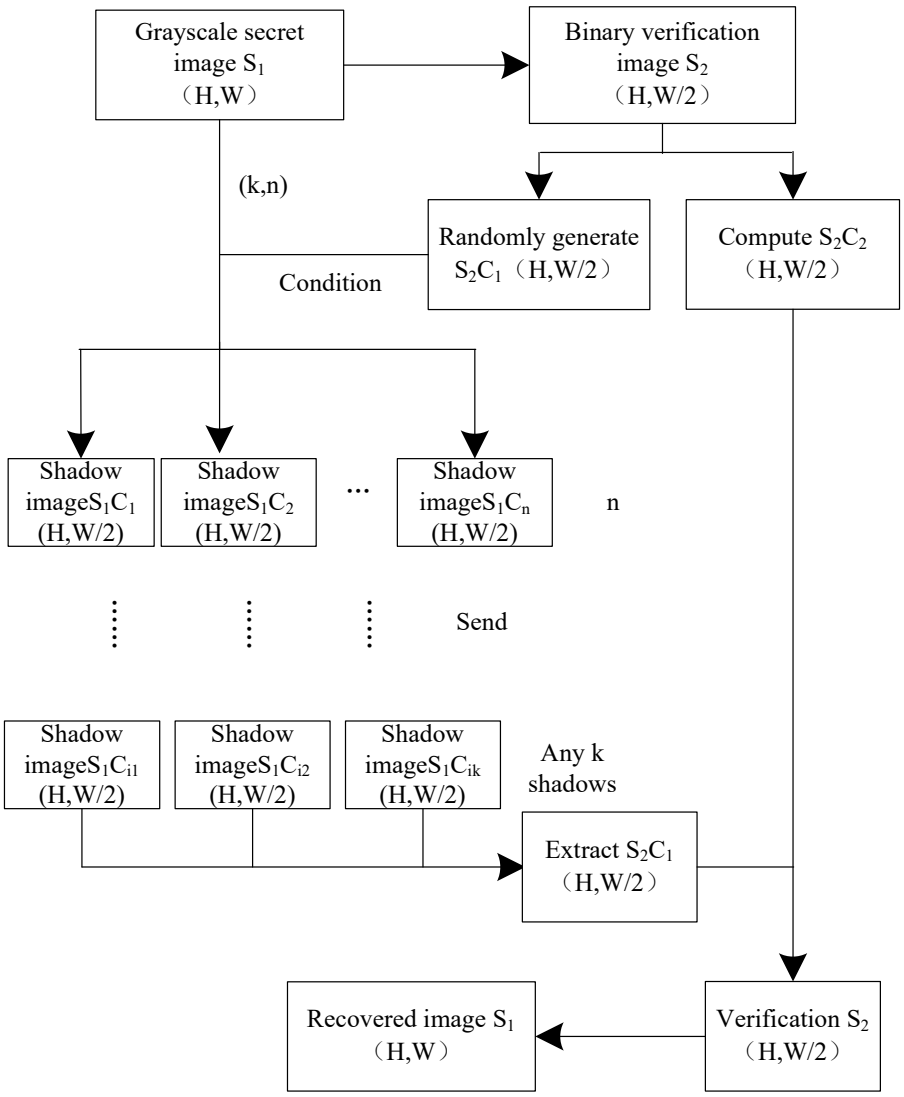


Figure 3. Framework of SIS method with verification ability.

In Figure 3, the sharing phase starts from the given secret image S_1. According to the size height and width (H, W) of S_1, a binary image S_2 with the same height and half width is first selected and generated. Then, two shadow images S_2C_1 and S_2C_2 are generated according to the random-grid VSS described in Section 3.2, in which S_2C_2 is used to verify the authenticity of the shadow image during recovery, and S_2C_1 is used to verify the authenticity of the shadow image. In the process of splitting S_1 into n shadow images, lossless embedding is performed in the lowest two bits of each shadow image pixel. The specific method is detailed in Section 4.2. The sharing algorithm is given in "Sharing algorithm of the proposed method".

After sharing, the (k, n) threshold image can be transmitted independently by multi-channel. When

Sharing algorithm of the proposed method
Input: grayscale secret image S_1, with size of (H, W),threshold (k,n), n values of x, denoted by $x_1, \ldots x_n$
Output: n shadow images $\{SC_i, 1 \leq i \leq n\}$, image size (H,W/2) and one verification shadow image S_2C_2
Algorithm: 1. According to S_1, the binary image S_2 with the size of (H,W/2) is selected to be generated 2. Using (2,2) threshold random-grid algorithm for image S_2 to generate shadow images S_2C_1 and S_2C_2 3. Select $P = 257$, set H = 0, j = 0 4. For pixels $S_1(h, 2j-1) \in \{S_1(h, w)\|1 \leq h \leq H, 1 \leq w \leq W\}$, $j \in (1, W/2)$ construct $k-1$ degree polynomial $f(x) = (a_0 + a_1x + ... + a_{k-1}x^{k-1}) \bmod P$ Among them: $a_0 = S_1(h, 2j$ - $1)$, $a_{k-1} = S_1(h, 2j)$, $a_i \in$ (0,255) is random ,i $\in (1, k-2)$ 5. Calculate the pixels values of $sc_1 = f(x_1)$, $sc_2 = f(x_2)$, ..., $sc_n = f(x_n)$ to be embedded to shadow image S_2C_1 according to the constraint conditions 6. Assign $sc_1, sc_2, ..., sc_n$ to $SC_1(h, j), SC_2(h, j), ..., SC_n(h, j)$ 7. Select the next pair of pixels according to the current (h,j), and repeat steps 4-6 until $h = H$, $j = W/2$ 8. Output n shadow images $SC_1, SC_2, ..., SC_n$

secret image recovery is needed, the binary image S_2C_1 can be extracted from each received shadow image, with S_2C_2 XOR or superimposing can obtain image S_2 to complete verification; then Lagrange interpolation method is used to calculate and restore the original secret image. When there are k or more shadow images that are verified as true, i.e., the binary image S_2C_1 can be generated, we can realize the lossless recovery of the secret image. The algorithm of recovery phase is given in "Recovery algorithm of the proposed method".

In this scheme, two improvements have been made to the traditional Thien and Lin's scheme [3]. One is to choose to set the P value of the prime field to 257, so that the secret image can be restored without loss. As for the case of the invalid value 256, the limitation will be added to the pixel generation method of the shadow image described in Section 4.2. The second is to embed two secret pixels into the coefficient a_0 and a_{k-1} each time, which can not only avoid the serious information leakage problem of the shadow image, but also avoid the high time complexity of the algorithm in the recovery phase.

In Thien and Lin's scheme [3], because k secret pixels are embedded each time, the loss of the randomness of coefficients will lead to serious information leakage, so additional encryption must be used to ensure the security. In addition, the time complexity of the scheme is high because it needs to calculate all k coefficients of the polynomial in the recovery phase. In our scheme, two secret pixels are embedded each time, keeping the randomness of coefficient setting. At the same time, considering the efficiency of Lagrange interpolation in the recovery phase, the computational complexity can be kept consistent with that of embedding only one secret pixel.

The solution formula of Lagrange interpolation method for known k points, denoted by $(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)$, is in formula 4.1.

Recovery algorithm of the proposed method
Input: any k $x_{i_1}, x_{i_2}, ..., x_{i_k}$ and corresponding shadow images $SC_{i_1}, SC_{i_2}, ..., SC_{i_k}, i_k \in (1, n)$ size (H,W/2), verification shadow image S_2C_2
Output: recovered grayscale secret image S'_1, with size of (H, W)
Algorithm: 1. Read the lowest two bit values of each pixel for k shadow images in turn, use "XOR" to generate a binary image $S_2C'_1$ 2. Calculate the "XOR" of $S_2C'_1$ and S_2C_2 to generate image S'_2, and check whether it is the same as the original binary image S_2, If not, the verification fails and the algorithm stops; 3. Set P=257,h=0,j=0 4. Take out the pixels from the k-shadow images in turn denoted by $SC_{i_1}(h, j), SC_{i_2}(h, j), ..., SC_{i_k}(h, j)(1 \le h \le H, 1 \le j \le W/2)$ 5. Set $f(x_{i_1}) = SC_{i_1}(h, j), f(x_{i_2}) = SC_{i_2}(h, j), ..., f(x_{i_k}) = SC_{i_k}(h, j)$ Lagrange interpolation is used to solve the following equations: $f(x_{i_1}) = (a_0 + a_1 x_{i_1} + ... + a_{k-1} x_{i_1}^{k-1}) \bmod P$ $f(x_{i_2}) = (a_0 + a_1 x_{i_2} + ... + a_{k-1} x_{i_2}^{k-1}) \bmod P$ ..... $f(x_{i_k}) = (a_0 + a_1 x_{i_k} + ... + a_{k-1} x_{i_k}^{k-1}) \bmod P$ 6. Calculate the secret pixel values $S'_1(h, 2j-1) = a_0, S'_1(h, 2j) = a_{k-1}$ 7. Select the next pair of pixels according to the current (h, j) value, and repeat steps 4-6 until $h = H, j = W/2$ 8. Arrange the secret pixel values to generate the restored image S'_1

$$L(x) = \sum_{i=1}^{k} y_i l_i(x) \tag{4.1}$$

$$l_i(x) = \prod_{j=1, j\neq i}^{k} \frac{x - x_j}{x_i - x_j} = \frac{x - x_1}{x_i - x_1} \times ... \times \frac{x - x_{i-1}}{x_i - x_{i-1}} \times \frac{x - x_{i+1}}{x_i - x_{i+1}} \times ... \times \frac{x - x_k}{x_i - x_k} \tag{4.2}$$

The time complexity to reconstruct the polynomial coefficients is mainly decided by the summation process of Eq. 4.1 and the multiplication times of Eq. 4.2. Substituting Eq. 4.2 into Eq. 4.1 we can get Eq. 4.3

$$L(x) = y_1(\frac{x - x_2}{x_i - x_2} \times ... \times \frac{x - x_k}{x_i - x_k}) + ... + y_i(\frac{x - x_1}{x_i - x_1} \times ... \times \frac{x - x_{i-1}}{x_i - x_{i-1}} \times \frac{x - x_{i+1}}{x_i - x_{i+1}} \times ... \times \frac{x - x_k}{x_i - x_k}) + ... y_k(\frac{x - x_1}{x_i - x_1} \times ... \times \frac{x - x_k}{x_i - x_k}) \tag{4.3}$$

For the multiplication times in the above process, such as computing $y_i(\frac{x-x_1}{x_i-x_1} \times ... \times \frac{x-x_{i-1}}{x_i-x_{i-1}} \times \frac{x-x_{i+1}}{x_i-x_{i+1}} \times ... \times \frac{x-x_k}{x_i-x_k})$, given $(x_1, y_1), (x_2, y_2)..., (x_k, y_k)$, $\frac{y_i}{(x_i-x_1)\times...\times(x_i-x_{i-1})\times(x_i-x_{i+1})\times...\times(x_i-x_k)}$ is a fixed item mainly decided by $(x - x_1) \times ... \times (x - x_{i-1}) \times (x - x_{i+1}) \times ... \times (x - x_k)$. According to the combination selection method, the calculation method of the constant coefficient a_0 is as follows: $C_{k-1}^0 \times (-x_1) \times ... \times (-x_{i-1}) \times (-x_{i+1}) \times ... \times (-x_k)$, where $C_{k-1}^0 = 1$ and $C_{k-1}^{k-1} = 1$.

Since there is only one selected method for the generation of these two coefficients, there are k items in Eq. 4.3 and each item needs to calculate nearly a constant multiplication of k. Therefore, the time complexity of computing $a_0$and a_{k-1} is $O(k^2)$, which is the same as that of embedding secret pixel only into a_0.

Except these two coefficients, the calculation of any other coefficient of x^i can not be completed within $O(k^2)$. For example, to solve the coefficient a_1, since $C_{k-1}^1 = k - 1$ means that taking one x item from $k - 1$ items, that is, $x \times (-x_2) \times ... \times (-x_{i-1}) \times (-x_{i+1}) ... \times (-x_k) + (-x_1) \times x \times ... \times (-x_{i-1}) \times (-x_{i+1}) \times ... \times (-x_k) + ... + (-x_1) \times ... \times (-x_{i-1}) \times (-x_{i+1}) \times ... \times x$,whose time complexity is $O(k^2)$. In Eq. 4.3, there are k items, and each item needs such time complexity, so the time complexity of the total secret pixels recovery calculation reaches $O(k^3)$.

Therefore, in our scheme, a_0 and a_{k-1} embedded secret pixels can ensure that the time efficiency is consistent with that of only calculating a_0. When k is greater than 3, better hiding effect can be achieved. Although the minimum value of k is limited to some extent, considering that more than three transmission paths are usually selected in the practical application of multi-channel transmission, this kind of comprehensive efficiency and the characteristics including compression rate are practically valuable.

It should be noted that since this method generates half size shadow image, when the width value of the secret image is not even, it needs to be even processed first, such as increasing 0. In the recovery process, if the last column of data is found to be special, then delete it.

4.2. Embedding the verification bit into grayscale images

One key step of our method is to embed the verification information into the shadow pixels. Through the improvement of the traditional polynomial-based algorithm in Section 4.1, when k is not less than 3, random variables will participate in the calculation process. For example, when k is 3, the polynomial can be constructed by selecting two given secret pixels s_{11} and s_{12} are in formula 4.4.

$$f(x) = (s_{11} + a_1 x + s_{12} x^2) \bmod 257 \tag{4.4}$$

Among them, $a_1 \in [0, 255]$ is generated by random function.

Because the algorithm uses module 257, we can avoid the situation of $f(x) = 256$ by changing the random value of a_1 when generating the specific value of $f(x)$, because 256 is beyond the range of grayscale image pixel value.

According to the above change idea, similarly, the value of $f(x)$ can meet certain characteristics by adjusting a_1. Here, we choose to set the XOR operation result of the lowest two bits of $f(x)$ to a specific value, that is, in the process of generating the shadow images $S_1C_i(i \in [1, n])$, we adjust the XOR operation result of the lowest two bits of $S_1C_i(h, w)(i \in [1, n], h \in [1, H], w \in [1, W/2])$ to be the bit value of the verification shadowed pixel $S_2C_1(h, w)(h \in [1, H], w \in [1, W/2])$, so as to realize the lossless embedding of the verification information.

For the two consecutive secret pixel values s_{11} and s_{12} in S_1, the corresponding bit value of s_2c_1 and (k, n) threshold generation algorithm is given in "The generation algorithm for shadow image with verification information embedded into the least two bits".

The generation algorithm for shadow image with verification information embedded into the least two bits
Input: (k, n) threshold, $x_1, \ldots x_n$, secret pixel values s_{11} and s_{12}, and embedded bit value s_2c_1
Output: n image pixel values s_1c_1,s_1c_2,...,s_1c_n
Algorithm: 1. Set $a = s_{11}$,a_{k-1}=s_{12}, randomly generate values of a_1 to a_{k-2} from (0,255) to construct polynomial $f(x) = (a_0 + a_1 x + ... + a_{k-1}x^{k-1}) \bmod P$ 2. Initialize count index $i = 0$ 3. Set $i = i + 1$, and calculate the value of $f(x_i)$ 4. If $f(x_i) > 255$, go to 1 5. If the XORed result of the lowest bit and the second lowest bit of $f(x_i)$! =s_2c_1, go to 1 6. Assign the value of $f(x_i)$ to s_1c_i 7. If $i < n$, go to 3 8. Output the pixel values s_1c_1,s_1c_2,...,s_1c_n of n shadow images

Embedding verification information into the XORing result of the lowest two bits is mainly based on two considerations. First one is to hide the feature of the specific bit of the shadow image, so that each bit of the pixels in the same position of the generated n shadow image has no relevance, so as to prevent malicious users from generating the forged shadow image when obtaining some real shadow images by analyzing the bit relation. The other one is to keep good randomness to ensure the effective generation of shadow images. Under random condition, the probability of generating 0 or 1 is 50%. With the same randomness, the probability that the value generated by XORing result of the lowest two bits is 0 or 1 is also 50%. However the other binocular operations do not have such performance, such as "AND", "OR", etc. The weak randomness of the result will lead to a rapid increasing of repeated calculations. Please refer to section 5.2 for experimental verification.

In addition, the randomness retained by XOR operation can be extended to the multi object operation, that is, the XORing result of the lowest two bits can be extended to any multiple bits participating

in the XORing operation. Under the condition of successfully generating the shadow bits, the shadow image still has good randomness, so that the proposed scheme has good expansion.

4.3. Security analysis

Because two kinds of secret sharing methods are integrated into the generation of shadow image, which has a certain impact on the original random distribution probability, so we need to analyze the security of embedding secret image.

Assuming that the secret image S_1 is linearly independent of the verification image S_2, the influence of this scheme on the verification image S_2 is analyzed. we use the random-grid VSS for (2,2) threshold to split S_2 into S_2C_1 and S_2C_2. Each pixel of S_2C_1 is randomly generated, and each pixel value of S_2C_2 is changed according to the value of S_2C_1. Therefore, this scheme does not change the security of the original random-grid VSS scheme. Assuming that the number of pixels in S_2 is m, the probability of brute-force cracking the original image S_2 is 2^m on the premise that any one of the shadow images is known. Here, we assume the m value is 128×64, which is far beyond the current limit that general-purpose computers can brutally crack.

For the secret image S_1, the improved polynomial-based SIS scheme is used in our scheme. Since we share two secret pixels in a polynomial, the k value in the (k, n) threshold of this scheme must start from at least 3 so as to ensure the randomness of the sharing process. Thus at least one coefficient can be randomly selected in the polynomial construction process, such as a_1 in formula 4.1. But at the same time, because the n pixels generated by each calculation of $f(x)$ have the constraint, only one coefficient is used to calculate the value of the n pixels. The random value in [0,255] may greatly reduce the randomness and even fail to meet the constraint conditions (see the experimental instructions in Section 5.3). Therefore, the actual k value in our scheme starts from 4, that is, the minimum degree form of the polynomial is:

$$f(x) = (s_{11} + a_1 x + a_2 x^2 + s_{12} x^3) \bmod 257 \tag{4.5}$$

In this way, there are two coefficients randomly selected in [0,255], which can ensure that when S_{11},S_{12} and $x_1, x_2, \ldots, x_n$ are given, by randomly changing the values of a_1 and a_2, the n equal values of XORed result of the lowest two bits can be generated. It should be noted that the larger n is, the stronger the constraint conditions are. Even if k is 4, when n is very large, there may still be cased that we can not meet the constraint conditions. We will give the specific experimental results in the experimental part.

5. Experiments and discussion

5.1. Image illustration

Based on the designed scheme in the previous section, we implement it with Python code. The grayscale image of 128×128 is selected as the secret image to be shared, and the binary image of 128×64 is set as the verification image. The experiments of SIS from (4,4) threshold to (8,10) thresholds are performed.

Among them, the (6,8) threshold SIS experiment is shown in Figure 4. Figure 4 (a) is a secret image to be shared, and Figure 4 (b) is a verification image. The two verification shadow images

generated by (2,2) threshold random-grid VSS method are shown in Figure 4 (c) and Figure 4 (d). The 8 shadow images generated by integrating the bit pixels of Figure 4 (c) into the sharing process of Figure 4 (a) are shown in Figure 4 (e) - (l).

In the verification and recovery phase, XORed value of the least two bits is extracted from each shadow image participating in the recovery to generate the verification information, and the "XOR" operation is performed on Figure 4 (d) to verify the effectiveness of the shadow image, as shown in Figure 5.

Figure 5 shows eight shadow images lena1.bmp to lena8.bmp and a fake shadow image fake.bmp in the previous sharing process. The image on its right side is the corresponding generated verification image. It can be seen that the true shadow image can effectively recover the verification information, while the fake shadow image can not pass the verification.

After verification, the shadow image is calculated to restore the result image, and some of the recovered results are shown in Figure 6.

It can be seen from Figure 6 that when the number of shadow images is less than 6, the original secret image can not be recovered, and there is no secret leakage problem; when the number of shadow images is not less than 6, if all the shadow images are true, the lossless recovery of the original secret image can be realized, and if there are forged shadow images, the effective recovery cannot be achieved.

5.2. Performance comparison and analyses

Because the verification information is embedded into the shadow image, the coefficients might not be randomly generated, and the required n shadow image pixels might not be generated, so the additional calculation will be increased when the scheme is implemented. In the experiment, the additional calculation times are compared and analyzed from two different aspects.

The first kind of comparative performance experiment is used to analyze the effect of the "XOR" in embedding verification information. In addition to the XOR mode, the binary logic operation of bit also has other modes such as AND and OR. In addition, this scheme is based on the improvement of the scheme in reference [16]. How about the performance compared with reference [16].

Let *count* denote the number of times that binomial coefficients are repeatedly selected for calculation when n shadow pixels are generated at one time and the constraint conditions are not satisfied. The constraint conditions include four cases: the lowest two bits of n shadow pixels have the same XORed value (this scheme), the lowest two bits OR operation, the lowest two bits have the same operation and the highest bit is the same (in [16]). For the original secret image with size of 128×64, generation average times $\overline{count}$ of repeated calculation for each image pixel and different (k, n) thresholds, are shown in Table 1.

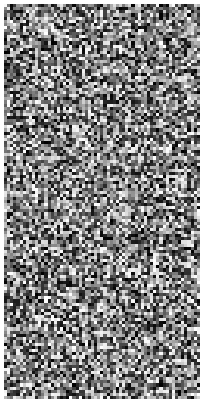
It can be seen from table 1 that as the increase of n, the number of times of repeated calculation increases faster, but the value and growth rate of XOR mode are roughly consistent with that of high bit mode, which is significantly better than that of AND and OR mode. Because there is the same random probability between the operation value of XOR and the method of highest bit condition, that is, the probability of generating constraint value of 0 or 1 when binomial coefficient is randomly selected is

(a) Original secret image

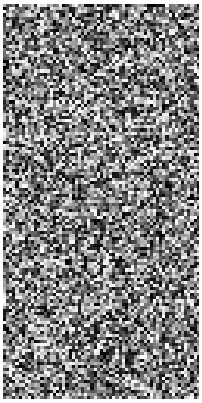
(b) Verification image

(c) Verification shadow 1

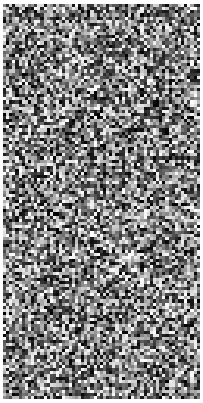
(d) Verification shadow 2



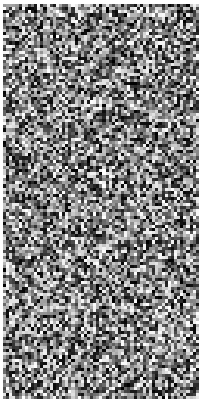
(e) Shadow 1



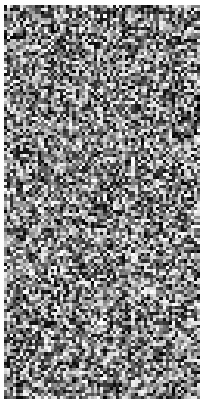
(f) Shadow 2



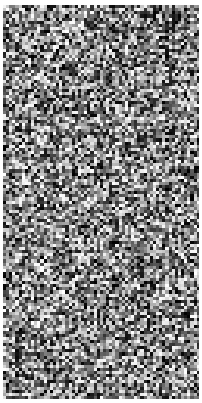
(g) Shadow 3



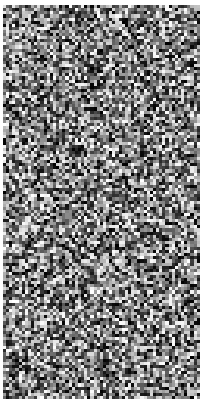
(h) Shadow 4



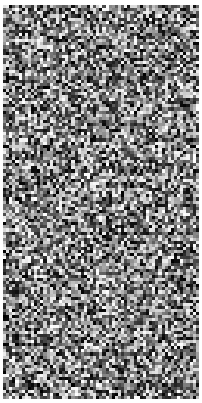
(i) Shadow 5



(j) Shadow 6

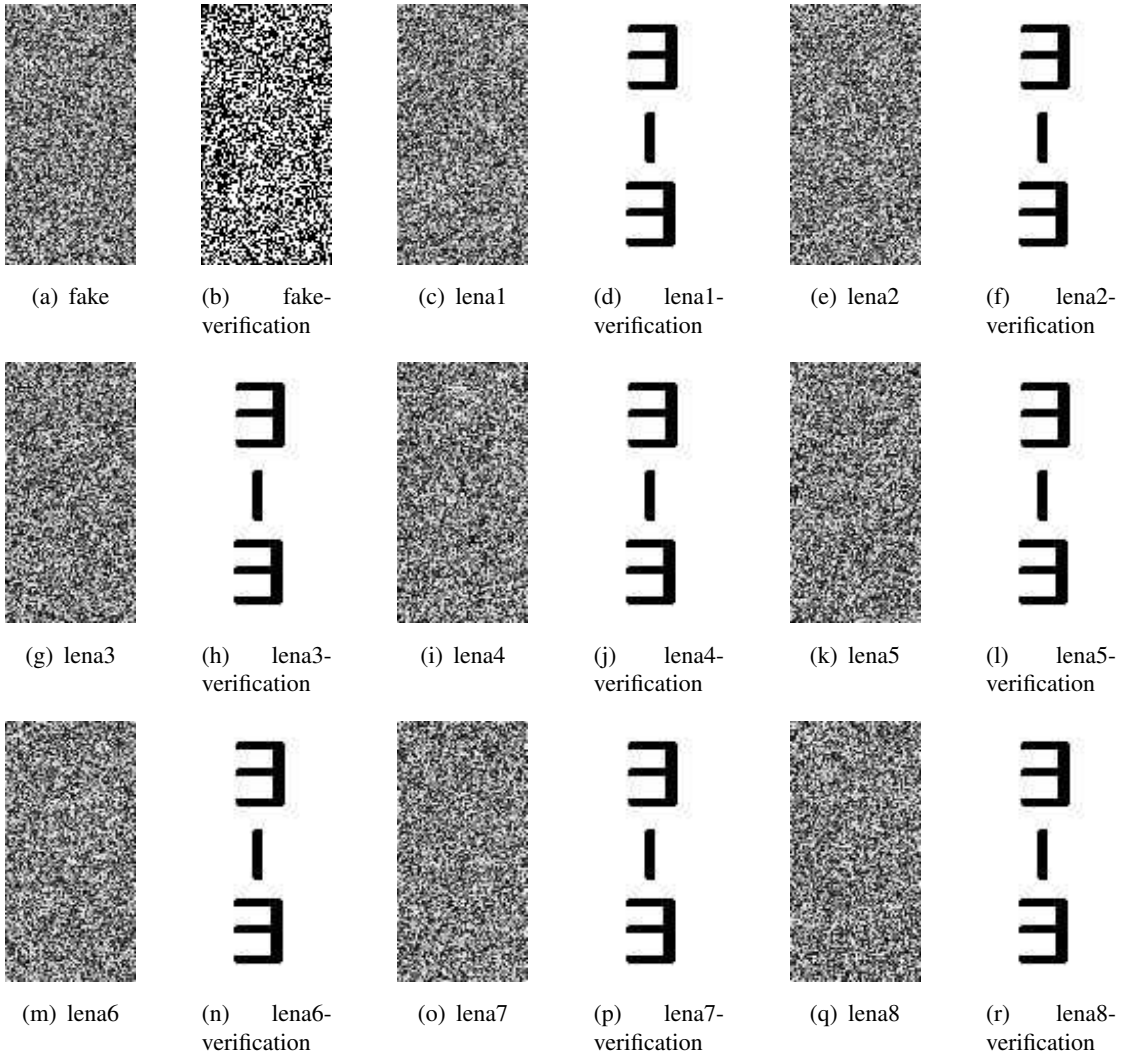


(k) Shadow 7



(l) Shadow 8

Figure 4. (6, 8) threshold SIS sharing.



(a) fake (b) fake-verification (c) lena1 (d) lena1-verification (e) lena2 (f) lena2-verification

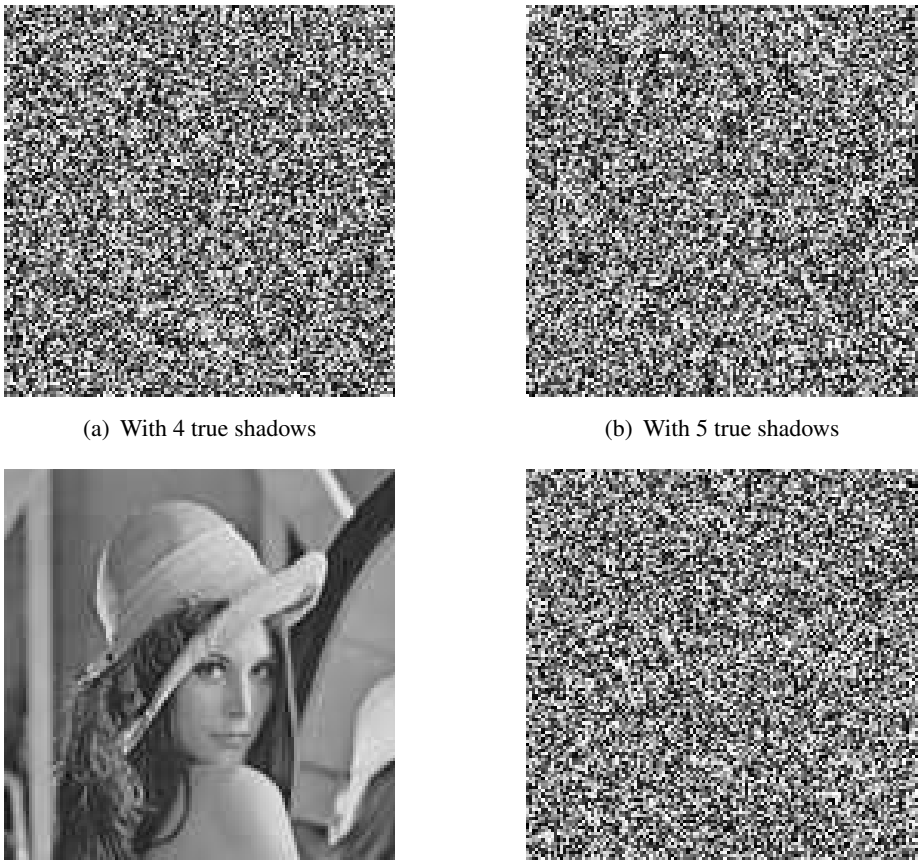
(g) lena3 (h) lena3-verification (i) lena4 (j) lena4-verification (k) lena5 (l) lena5-verification

(m) lena6 (n) lena6-verification (o) lena7 (p) lena7-verification (q) lena8 (r) lena8-verification

Figure 5. (6, 8) threshold shadow image verification.

Table 1. The average number of times of repeated calculation for generating the shadowed pixels in four cases (unit: Times).

(k, n)	**XOR Ours**	**AND**	**OR**	**Highest bit [16]**
(4,4)	15.19	236.71	236.89	15.35
(4,5)	31.37			31.36
(4,6)	66.73			65.16
(5,5)	31.46	524.25	520.34	31.54
(5,6)	63.09	2072.20	2090.10	63.83
(5,7)	129.25	11379.73	11307.26	128.56
(6,6)	63.46	2028.00	2146.88	63.71
(6,7)	129.48	8143.37	8498.96	130.05
(6,8)	262.07	32989.34	34368.82	256.86



(a) With 4 true shadows

(b) With 5 true shadows

(c) With 6 true shadows

(d) With 6 shadows (one fake)

Figure 6. (6, 8) threshold recovered results (part).

50%. This random selection mechanism with equal probability plays an important role in finding the values satisfying the constraint condition quickly. When "AND" and "OR" are used, the probability of generating constraint value 0 or 1 when binomial coefficient is randomly selected is biased. For "AND" mode, only when both values are 1, the result is 1, that is, the probability of generating 1 is 25%, and the probability of generating 0 is 75%. For "OR" mode, the biased probability leads to more times of repeated calculation with faster growth rate. When k is small, the feasible solution may not be generated due to the small random range, as shown in Table 1 (4,5) and (4,6) thresholds.

The second type of comparative experiment is to quantitatively analyze the scale of additional calculation times generated by the scheme itself with the change of (k, n) values. When k is not less than 4, we take the average value of multiple calculations to get the experimental results for (k, n) threshold from (4, 4) to (8, 10) as shown in Table 2.

Table 2. Average times of repeated calculation for pixel generation of shadow image with different thresholds (unit: time).

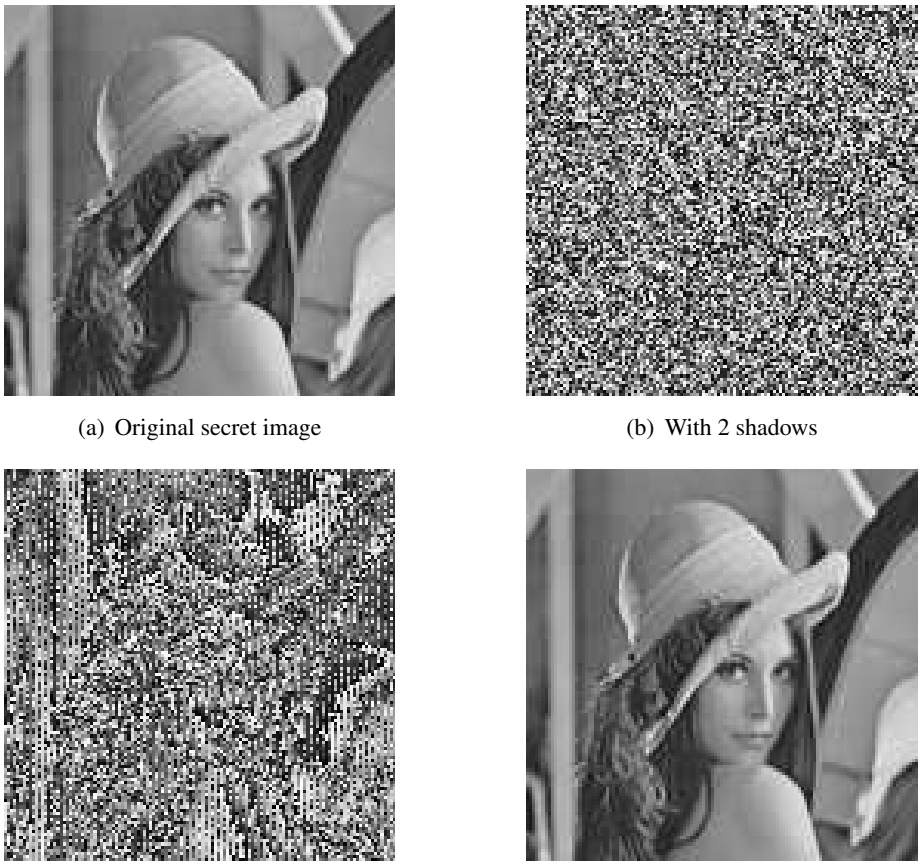
n	k	**4**	**5**	**6**	**7**	**8**
4		15.19				
5		31.37	31.46			
6		66.73	63.09	63.46		
7		140.85	129.25	129.48	130.43	
8		309.32	266.70	262.07	259.49	258.42
9		756.24	536.18	531.68	528.90	520.78
10			1061.84	1060.83	1053.71	1053.51

It can be seen from Table 2 that as the increase of (k, n) threshold, the number of times of repeated calculation shows a certain rule with the change of k and n. When n is fixed, the number of times of repeated calculation changes little with the increase of k, but when k is small, the number of times of repeated calculation will increase due to the small range of random value; while when k is fixed, the number of times of repeated calculation increases obviously with the increase of n, and the more image values that meet the constraint conditions need to be generated at the same time, showing a growth scale of about twice if n is too large. If the value of k is large and the value of k is small, such as (4, 10) threshold, there may also be the case that the threshold of n is larger than that of k.

5.3. Experiment for security

Because the scheme adopts the mechanism of compressed shadow image, that is, embedding two secret pixels into one shadow image pixel each time, and limiting the random value condition of coefficient, it is necessary to carry out security experiment analysis on whether there is image leakage in the scheme.

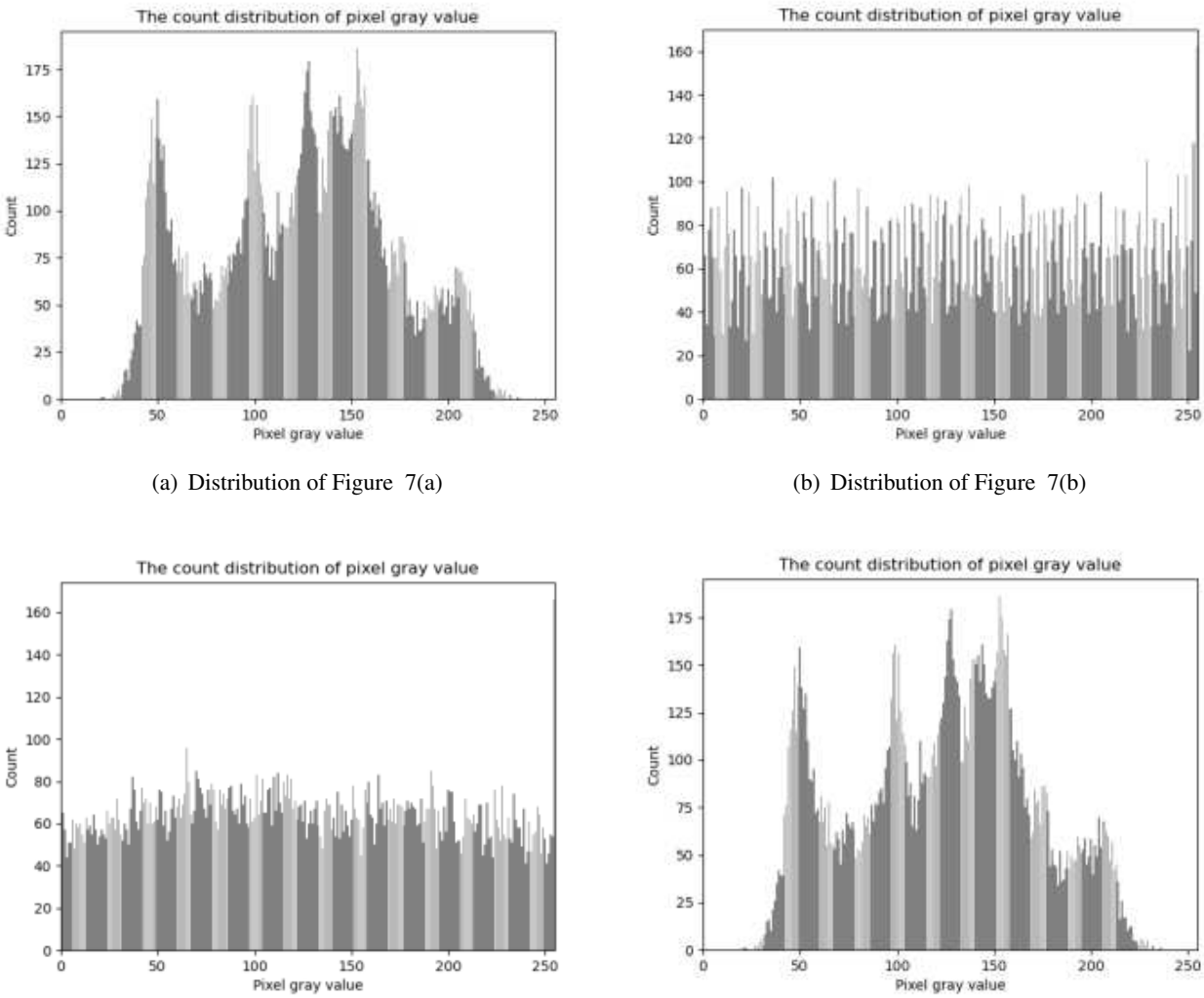
In the experiment, we first try to generate a (3,4) threshold SIS example. Because only one coefficient can change randomly when $k = 3$, we can not find a feasible solution in the process of pixel generation of some shadow images due to the constraints. Therefore, we mainly analyze whether there is information leakage in the sharing process when $k = 4$. Figure 7 and Figure 8 respectively show the recovered results of (4,4) threshold and the distribution of shadow image pixel values.



(a) Original secret image (b) With 2 shadows

(c) With 3 shadows (d) With 4 shadows

Figure 7. (4, 4) threshold recovered results with different shadow images.



(a) Distribution of Figure 7(a)

(b) Distribution of Figure 7(b)

(c) Distribution of Figure 7(c)

(d) Distribution of Figure 7(d)

Figure 8. Pixels distribution of Figure 7.

It can be seen from Figure 7 and Figure 8 that in the minimum threshold experiment of this scheme, there is a weak non random situation only when three shadow images are used for recovery, but the recovery results when k is less than 4 cannot be effectively analyzed.

In a word, we can see that this scheme can effectively realize the compressible SIS process with the ability of shadow image verification on the basis of adding a certain amount of repeated calculation. Compared with the reference [16], the advantages of this scheme are mainly reflected in two aspects. One is to improve the highest embedding verification bit to the lowest two bits "XOR" value to set the verification bit, so that there is no linear relationship between any bits of the generated shadow image, hiding the embedding rule of verification information, and improving the security from the perspective of anti-shadow image analysis; the other is to set the "XOR" value of the lowest two bits as the verification bit. On the one hand, two secret pixels are used to calculate the pixel value of the shadow image at one time. Although the number of repetitions of the calculation is equivalent to the highest embedding due to the limitation of the lowest two XORed values each time, the overall generation efficiency is doubled because the generated shadow image is 1/2 times of the shadow image in [16].

6. Conclusion

In order to realize the verification of secret image sharing (SIS) and shadow image, based on the current polynomial-based SIS scheme and the (2,2) threshold random-grid VSS with superposing in the "XOR" way, by improving the existing scheme strategy in the process of polynomial-based SIS mod 257, a scheme of compressed SIS with the ability of shadow image verification is implemented. The scheme has the ability of shadow image verification, pixel compression, loss tolerance and lossless recovery. Compared with the existing similar schemes, this scheme has some advantages in hiding pixel relation and high generating efficiency.

Based on the existing research situation, in the future we will continue to carry out research to improve the efficiency of shadow image generation: first, to study whether it can improve the current process of completely selecting coefficients in a random way, and reduce the number of repeated calculations to generate effective shadow image pixels; second, to study other embedding methods of verification information, so as to make the embedding of verification information more efficient.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported in part by the National Natural Science Foundation of China (grant number: 61602491), and the Key Program of the National University of Defense Technology (grant number: ZK-17-02-07).

Conflict of interest

The authors declared that they have no conflicts of interest to this work.

References

1. A. Shamir, How to share a secret, *Commun. ACM*, **22** (1979), 612–613.

2. M. Naor, A. Shamir, Visual cryptography, in *Workshop on the Theory and Application of of Cryptographic Techniques*, Springer, 1994, 1–12.
3. C. C. Thien, J. C. Lin, Secret image sharing, *Comput. Graphics*, **26** (2002), 765–770.
4. X. Yan, Y. Lu, L. Liu, X. Song, Reversible image secret sharing, *IEEE Trans. Inf. Forens.*, Early Access.
5. Y. Liu, C. Yang, Scalable secret image sharing scheme with essential shadows, *Signal Process. Image Commun.*, **58** (2017), 49–55.
6. G. Wang, F. Liu, W. Q. Yan, Basic visual cryptography using braille, *Int. J. Digital Crime Forens.*, **8** (2016), 85–93.
7. X. Yan, Y. Lu, L. Liu, J. Liu, G. Yang, Chinese remainder theorem-based two-in-one image secret sharing with three decoding options, *Digital Signal Process.*, **82** (2018), 80–90.
8. R. De Prisco, A. De Santis, On the relation of random grid and deterministic visual cryptography, *IEEE Trans. Inf. Forens.*, **9** (2014), 653–665.
9. C.-N. Yang, C.-C. Wu, D.-S. Wang, A discussion on the relationship between probabilistic visual cryptography and random grid, *Inf. Sci.*, **278** (2014), 141–173.
10. T.-H. Chen, K.-H. Tsao, User-friendly random-grid-based visual secret sharing, *IEEE Trans. Circuits Syst. Video Technol.*, **21** (2011), 1693–1703.
11. M. Tompa, H. Woll, How to share a secret with cheaters, *J. Cryptol.*, **1** (1989), 133–138.
12. P. Li, P. Ma, X. Su, Image secret sharing and hiding with authentication, in *2010 First International Conference on Pervasive Computing, Signal Processing and Applications*, IEEE, 2010, 367–370.
13. C.-C. Chang, Y.-P. Hsieh, C.-H. Lin, Sharing secrets in stego images with authentication, *Pattern Recognit.*, **41** (2008), 3130–3137.
14. Y. Liu, C.-C. Chang, A turtle shell-based visual secret sharing scheme with reversibility and authentication, *Multimed. Tools. Appl.*, **77** (2018), 25295–25310.
15. Y.-X. Liu, Q.-D. Sun, C.-N. Yang, (k, n) secret image sharing scheme capable of cheating detection, *EURASIP J. Wirel. Commun. Netw.*, **2018** (2018), 72.
16. X. Yan, Q. Gong, L. Li, G. Yang, Y. Lu, J. Liu, Secret image sharing with separate shadow authentication ability, *Signal Process. Image Commun.*, **82** (2020), 115721.
17. X. Wu, W. Sun, Random grid-based visual secret sharing with abilities of or and xor decryptions, *J. Visual Communi. Image Represent.*, **24** (2013), 48–62.
18. X. Yan, S. Wang, X. Niu, C.-N. Yang, Random grid-based visual secret sharing with multiple decryptions, *J. Visual Communi. Image Represent.*, **26** (2015), 94–104.
19. X. Yan, Y. Lu, Progressive visual secret sharing for general access structure with multiple decryptions, *Multimed. Tools. Appl.*, **77** (2018), 2653–2672.

AIMS Press

© 2020 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)