Multimed Tools Appl (2017) 76:7677–7698
DOI 10.1007/s11042-016-3429-8

A secure image sharing scheme with high quality stego-images based on steganography

Junhui He[1] · **Weiqiang Lan**[1] · **Shaohua Tang**[1]

Received: 4 August 2015 / Revised: 28 February 2016 / Accepted: 3 March 2016 /
Published online: 14 March 2016

Abstract Image sharing can be utilized to protect important commercial, military or private images against a single point of failure. Many existing image sharing schemes may have one or more of the security weaknesses as follows: First, noise-like image shares may easily arouse the attackers' attention; Second, cheating in the recovery of the secret image cannot be prevented effectively; Third, the requisite size of cover images may be very large; Finally, poor quality of the stego-images may lessen camouflage effects. In this paper, a novel secure image sharing scheme with high quality stego-images is proposed. With the use of LOCO-I compression as a preprocessing approach, the statistical correlations between neighboring pixels of a secret image drop significantly, which may greatly enhance the visual security of the proposed scheme. And the necessary size of cover images is reduced. Moreover, the PSNR values of stego-images are much higher than the related works. In order to detect three kinds of deception during secret image reconstruction, the hash-based message authentication codes of an image share, the value of argument x and the identity ID of a participant are embedded into a cover image together with the image share. In addition, the application of dynamic embedding with a random strategy further enhance the security of our scheme.

Keywords Image sharing · Steganography · LOCO-I compression · Message authentication code · Dynamic embedding

✉ Junhui He
hejh@scut.edu.cn

Weiqiang Lan
lwq@mail.scut.edu.cn

Shaohua Tang
csshtang@scut.edu.cn

[1] School of Computer Science and Engineering, South China University of Technology, Guangzhou Higher Education Mega Centre, Guangzhou 510006, People's Republic of China

1 Introduction

With the rapid development of Internet and continuing increase of network bandwidth, images are used more and more often in commercial, military or personal areas. How to tighten image security has become an important problem nowadays. Image encryption is a conventional method to protect an image by making it unreadable without a proper key, but the storage of the noise-like encrypted image in a public cloud, or transmission over an insecure communication channel, may easily arouse attackers' attention, and the encrypted image may thus be destroyed, intercepted or decrypted with brute force. Image steganography is another kind of novel technique developed in recent years for keeping an image safe by embedding it into the innocuous cover media with the use of information hiding techniques. It conceals the existence of a secret image of great importance, as well as protecting its contents. However, an obvious flaws of steganographic methods is that a single misfortune, such as cloud storage breakdown, cover media damage and computer sabotage, can make the protected image inaccessible. The schemes of dealing with such kind of problems are worth the researcher's while to explore.

The concept of threshold secret sharing was pioneered independently by two cryptography researchers Shamir [20] and Blakley [2]. A (t, n) threshold scheme divides data D into n shares in such a way that D can be reconstructed with t or more shares, but even complete knowledge of $t - 1$ shares reveals no information about D. Threshold schemes originally focused on cryptographic keys management. Later in [19], Naor and Shamir presented a method of visual cryptography based on (t, n) visual secret sharing scheme. In this scheme, n image shares are printed on n transparencies respectively, any t out of which may be stacked together to reveal the secret image. The most important characteristic of visual cryptography is that decryption can be done without any cryptographic computations. But the size of image shares may be expanded and the contrast of the revealed secret image is usually poor. With the inspiration from (t, n) threshold schemes based upon polynomial interpolation, a method of secret image sharing is proposed by Thien and Lin [21]. It generates n image shares whose size is only $1/t$ of the secret image's, and any t $(t \leq n)$ image shares can be used to restore the secret image, but arbitrarily $t - 1$ image shares cannot get sufficient information to reveal the secret image.

Instead of Shamir's polynomials, cellular automata (CA) may also be used for image sharing [1, 13]. Yang and Chu [26] presented a novel (t, n) image sharing scheme in which the secret image may be gradually revealed with t or more image shares. Image sharing may not only be used as a kind of image protection method against a single misfortune, but also be applied in image authentication [11, 15]. In the reconstruction process of Thien-Lin's scheme, each image share plays an equivalent role. Guo et al. [10] proposed a hierarchical threshold secret image sharing scheme based on Birkhoff interpolation, in which the image shares are partitioned into several levels of different privileges. However, the secret image may be partially restored even if some of the hierarchical threshold requirements are not satisfied, which may leak out visual information about the original secret image. In other words, the scheme is not visual security. Recently, Yuan [28] presented two secret sharing methods, in which the secret revealing process is based on simple Boolean operations. But Yuan's two schemes are only applicable to sharing binary images and 2-bit images respectively. Chen et al. [7] also proposed a novel highly efficient secret image sharing scheme based on Boolean operations. It can share n secret images among n participants and recover all n secret images from n image shares.

As mentioned in [21], the property of an image may be helpful in restoring the secret image, so random permutation of a secret image is applied before sharing to eliminate the correlations between neighboring pixels. But just applying random permutation is not enough for its visual security, which will be illustrated in Section 3. Moreover, the noise-like image shares may arouse suspicion from attackers. In order not to expose the existence of the image shares, steganographic techniques are suggested to hide them in user-selected camouflage images (also called cover images) [6, 8, 16, 17, 22, 25, 27]. Although more complex steganographic methods have been used besides LSB substitution, such as dynamic programming strategy [4], exploiting modification direction (EMD) method [22], modulus function [5] and (s, c) hiding method based on PSNR estimation [16], the quality of stego-images is not proved to be so good as expected. One of the most important reasons is that the size of image shares to be embedded in the cover images is large, and there is a contradiction between the steganographic capacity and the quality of stego-images. Moreover, the size of cover images used in the existing methods [5, 17, 25, 27] is 4 times as big as that of a secret image, which would not be favorable for their later storage, transmission or other processing. In [23], a secret image sharing methods with smaller image shares is proposed with the application of Huffman codec.

When the stego-images hidden with image shares are uploaded to public cloud storage, or distributed among participants, some of them may be modified incidentally or intentionally, which will cause problems in the extraction of image shares and subsequent restoration of the secret image. Therefore, it is desirable to check authenticity of the image shares which will be used to reconstruct a secret image. In Lin-Tsai's scheme [17], a fragile watermark is embedded into each four-pixels block of the stego-image with the use of parity-bit checking to provide authentication ability. But the stego-image may be easily manipulated to have the same even or odd parity as the genuine stego-image and successfully pass the parity verification. In the subsequent research, k check bits generated by a hash function [25, 27] or using Chinese Remainder Theorem (CRT) [6] are embedded into each 2×2 block of a cover image. The probability of detecting modification in each stego-image block is $(2^k - 1)/2^k$, where $k = 1$ in [25, 27] and $k = 4$ in [6]. Obviously, larger value of k means better authentication ability. In [8], a new authentication-chaining method is introduced to achieve $15/16$ detection probability with the use of 2 check bits. Islam et al. [12] presented a secure (n, n) image sharing approach with the help of n key-images and homomorphic properties of public key cryptosystems. But it is possible to obtain a low quality version of the secret image with $n - 1$ image shares. Lin and Chang [18] proposed a novel secret image sharing scheme that can resist cheating attacks. Both the secret image and the host image may be restored without distortion. In [14], Fletcher-16 checksum is used to detect the counterfeit. Ulutas et al. [22] developed a scheme with adaptive authentication strength by selecting the number of check bits proportional to block size. If most blocks of a stego-image are altered, the probability that the fake stego-image would be validated is low. However, if only a small portion of the stego-image blocks are manipulated, there may be a real possibility that the fake stego-image passes the verification. In addition, a large amount of computation is needed to generate authentication bits for every block of a stego-image.

In this paper, a novel secure secret image sharing scheme with steganography is proposed. With the use of LOCO-I compression, the statistical correlations between neighboring image pixels of a secret image drop sharply and thus the visual security is greatly improved. The size of image shares is also reduced and can be easily embedded into cover images. Moreover, the necessary size of cover images is generally not bigger than $2/t$ times of a secret image. And LSB substitution steganography with an optimal pixel adjustment process is employed to hide image shares. Consequently, the quality of stego-images are much higher than the existing schemes'. In order to prevent three kinds of cheating described in Section 4.3 during secret image reconstruction, the hash-based message authentication codes of a image share, the value of argument x and the identity ID of a participant are calculated and embedded into a cover image together with the image share. Because of less number of evaluations of hash function for a stego-image, the amount of calculation is decreased. In addition, dynamic embedding with a random strategy is applied to further enhance the security of our scheme.

The remainder of this paper is organized as follows. In Section 2, the preliminaries are introduced. Then visual security of a secret image sharing scheme is analyzed in Section 3. Section 4 described the proposed scheme in detail. The experimental results and security analysis are given in Section 5. The conclusions are stated finally in Section 6.

2 Preliminaries

2.1 Simple (t, n) threshold scheme

Shamir [20] proposed a (t, n) threshold scheme based on polynomial interpolation. The threshold scheme divides data D into n shares $D_1, D_2, \cdots, D_n$ in such a way that D is easily reconstructed from t $(t \le n)$ or more shares. But knowledge of any $t-1$ or fewer shares leaves D completely undetermined. To split D into shares D_i, we pick a random $t-1$ degree polynomial $f(x)$. And modulus arithmetic instead of real arithmetic is applied in computing the value of $f(x)$ to avoid ambiguity.

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \mod p \tag{1}$$

where p is a prime number bigger than both D and n, $a_0 = D$ and a_k $(k = 1, 2, \cdots, t-1)$ are randomly chosen from a uniform distribution over the integers in $[0, p-1]$. In order to calculate the share D_i, an non-negative integer value x_i smaller than p is freely chosen for x, all x_i must be distinct from one another, e.g. $x_i = i$ may be used. And then evaluate

$$D_i = f(x_i) \mod p \quad \text{for} \quad i = 1, 2, \cdots, n. \tag{2}$$

From (2), it can be seen that the size of each share D_i will not exceed the size of the original data D, i.e. all of them lie in $[0, p-1]$.

Given any t or more pairs of $\{(x_i, D_i)\}_{i=1}^{n}$, the $t-1$ degree polynomial $f(x)$ can be reconstructed. Without loss of generality, we assume that the known t pairs are $(x_1, D_1), (x_2, D_2), \cdots, (x_t, D_t)$. And the original polynomial $f(x)$ may be reconstructed by Lagrange interpolation.

$$f(x) = \sum\nolimits_{i=1}^{t} \prod\nolimits_{j=1, j\ne i}^{t} \frac{x - x_j}{x_i - x_j} D_i \mod p \tag{3}$$

By (3), the original data $D = f(0) \mod p$ can be obtained.

2.2 Secret image sharing

To protect an important image from a single misfortune which may make the image inaccessible, Thien and Lin [21] proposed a secret image sharing method derived from the (t, n) threshold scheme mentioned above. In Thien-Lin's method, a 8-bit grayscale secret image is first permuted randomly, and then divided into several units of the same size without an overlap. Each unit has t consecutive pixels. And the t pixel values in a unit are used as the t coefficients $a_0, a_1, \cdots, a_{t-1}$ of the polynomial $f(x)$ in (1), and the prime number p is chosen to be 251, which is the greatest prime number not larger than 255.

Let $x_i = i$ $(i = 1, 2, \cdots, n)$ and be substituted into (1), n output values $f(x_1)$, $f(x_2)$, $\cdots$, $f(x_n)$ are generated for the unit and sequentially assigned to n image shares. After all of the units are processed similarly, n image shares will be produced. The size of each image share is only $1/t$ of the secret image's size, which gives the benefit in future process of the image shares, such as storage, transmission, or image hiding.

Any t or more of the n image shares can be used to restore the secret image. In the reveal phase, t pixel values together with their corresponding x_i's constitute t pairs $\{(x_i, f(x_i))\}_{i=1}^{t}$, each of which is from one of the t image shares separately, are used to reconstruct the polynomial $f(x)$ by (3). Then the t coefficients of $f(x)$ are collected as t pixel values of the permuted secret image. When all pixels of the t image shares are processed, the original secret image may be recovered by applying the inverse-permutation.

2.3 LSB substitution steganography with OPAP

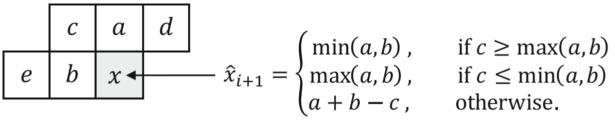
LSB substitution steganography is one of the conventional techniques capable of hiding large secret message in a cover image without introducing many perceptible distortions. It works by replacing the k rightmost LSBs of randomly selected pixels in the cover image with the secret message bits. The choice of pixels is determined by a secret key. The secret message bits may be extracted directly from the k rightmost LSBs of these pixels which are used during embedding. In [3], an optimal pixel adjustment process (OPAP) is proposed to enhance the image quality of the stego image obtained by the simple LSB substitution steganography. Let p_v, p'_v, p''_v be the corresponding pixel value of the v-th pixel in the cover image, the stego image produced by l rightmost LSB steganography in which l least significant bits of p_v are replaced by l message bits, and the refined stego image obtained by LSB steganography with OPAP, and $e_v = p'_v - p_v$ be the embedding error between p_v and p'_v. The OPAP can be described as the following equation.

$$p''_v = \begin{cases} p'_v - 2^l & \text{if } e_v \in (2^{l-1}, 2^l) \text{ and } p'_v \geq 2^l. \\ p'_v & \text{if } e_v \in (2^{l-1}, 2^l) \text{ and } p'_v < 2^l, \text{ or } e_v \in [-2^{l-1}, 2^{l-1}], \\ & \text{or } e_v \in (-2^l, -2^{l-1}) \text{ and } p'_v \geq 256 - 2^l. \\ p'_v + 2^l & \text{if } e_v \in (-2^l, -2^{l-1}) \text{ and } p'_v < 256 - 2^l. \end{cases} \tag{4}$$

2.4 LOCO-I lossless image compression

LOCO-I (LOw COmplexity LOssless COmpression for Images) [24] is the core algorithm of the new ISO/ITU standard for lossless and near-lossless compression of continuous-tone images, JPEG-LS. Its basic components are summarized in the following.

- Predictor: The prediction model and the guessed value of the current pixel x_{i+1} is depicted in Fig. 1.
- Context Modeling: A two-sided geometric distribution is adopted to model the prediction residuals. This simple context model may approach the capability of the more complex universal techniques for capturing high-order dependencies.



$$\hat{x}_{i+1} = \begin{cases} \min(a,b), & \text{if } c \geq \max(a,b) \\ \max(a,b), & \text{if } c \leq \min(a,b) \\ a+b-c, & \text{otherwise.} \end{cases}$$

Fig. 1 Predictor used in LOCO-I

- Coder: In a low complexity framework, the two-sided exponentially decaying distribution can be efficiently encoded with an extended family of Golomb-type codecs, which are adaptively chosen during encoding.

3 Visual security of image sharing

In general, there exists great spatial redundancy in an image, that is, the adjacent pixels are highly correlated. If a secret image is directly shared using the method stated in Section 2, the resulting image shares may reveal many visual contents of the secret image, which should be unacceptable for protecting an important image. A (2, 2) threshold scheme is used to demonstrate the concept. As shown in the Fig. 2, Elaine (a) and SCUT (f) are two secret images, which are divided into two image shares, (b)–(c) and (g)–(h), respectively using (2, 2) threshold scheme. By Lagrange polynomial interpolation, the secret images can be successfully restored from their corresponding image shares, which are demonstrated in (d) and (i). Both of them are the same as the original secret images respectively. However, it may be easily seen that both the image shares (b) and (c) of the secret image (a) give away much information about the secret image (a), so as (g) and (h). Moreover, if only one image share (less than the threshold $t = 2$) is applied in the reconstruction by letting $(x_1, f(x_1)) = (x_2, f(x_2))$. We can obtain the output images (e) and (j), which may be recognized as the sketches of the secret images (a) and (f).

To improve the visual security of secret image sharing, random permutation is advised to destroy the correlations between neighboring pixels before sharing process [21]. As illustrated in Fig. 3, all of the image shares (a)–(b) and (f)–(g) appear noise-like images if the input secret images are permuted randomly. For the security ability contributed by image sharing instead of depending on confidential procedure is concerned here, it is reasonable to assume that the details of the preprocessing methods are known during secret image reconstruction. By the method mentioned above, i.e. let $(x_1, f(x_1)) = (x_2, f(x_2))$, the secret image may be reconstructed by any one of the two image shares which are produced by

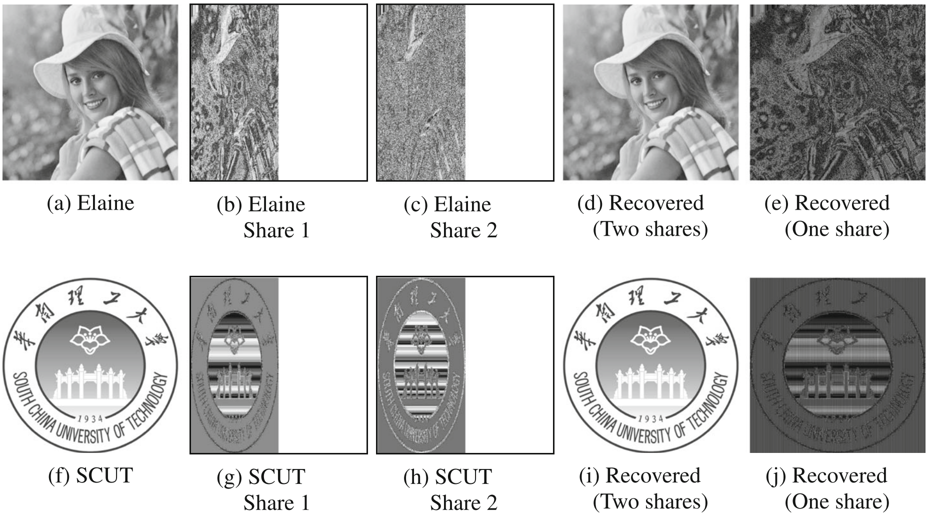

(a) Elaine
(b) Elaine Share 1
(c) Elaine Share 2
(d) Recovered (Two shares)
(e) Recovered (One share)
(f) SCUT
(g) SCUT Share 1
(h) SCUT Share 2
(i) Recovered (Two shares)
(j) Recovered (One share)

Fig. 2 (2, 2) Image sharing scheme and visual security

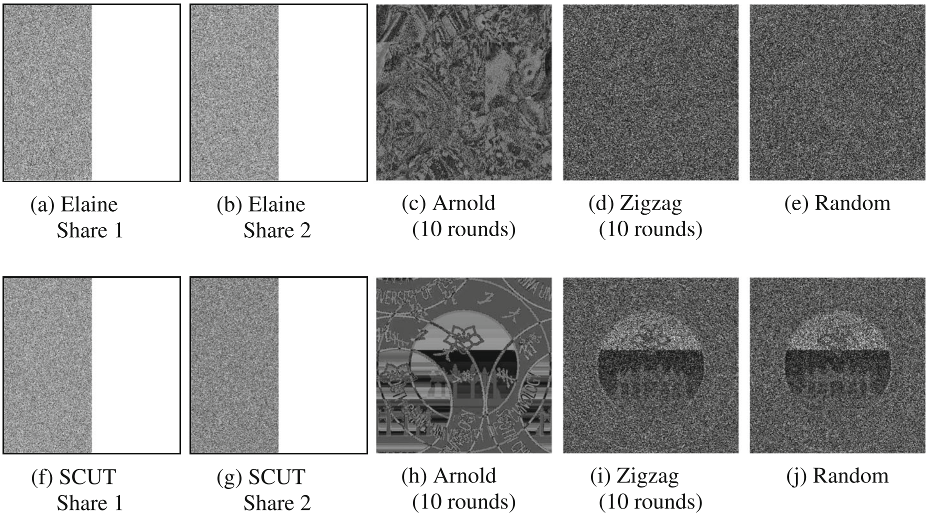


Fig. 3 Visual security when three different image scrambling methods are used

(2, 2) threshold scheme with random permutation, Arnold transformation or Zigzag scrambling. Though visual information may not be disclosed by the image shares themselves, the restored images shown in Fig. 3(c)–(e) and (h)–(j) exhibit many similarities to the original secret images, especially for the simpler image SCUT.

4 The proposed image sharing scheme

In this section, we describe the proposed scheme in detail. A secret image is firstly LOCO-I compressed, followed by (t, n) image sharing on the compressed image. And then n image shares are generated. Each of the image shares is then embedded into a cover image together with the authentication bits. The n stego images may be distributed to n participants or transmitted to n cloud-based storage. Any t or more out of the n stego-images may be employed to restore the original secret image provided that t pairs of $(x_i, f(x_i))$ are authentic.

Let S denote a grayscale secret image to be protected, the size of which is $H \times W$ and its pixels are denoted as $s_{h,w} \in [0, 255]$, $h = 1, 2, \cdots, H$; $w = 1, 2, \cdots, W$. And n cover images are denoted by C_i $(i = 1, 2, \cdots, n)$ corresponding with n participants (or cloud storage) P_i $(i = 1, 2, \cdots, n)$. The pixels of the i-th cover image C_i are represented by c_{iv} $(v = 1, 2, \cdots, |C_i|)$, whose binary representation is $c_{iv}^7 c_{iv}^6 c_{iv}^5 c_{iv}^4 c_{iv}^3 c_{iv}^2 c_{iv}^1 c_{iv}^0$, here c_{iv}^7 is the most significant bit and c_{iv}^0 is the least significant bit of the pixel c_{iv}. We denote the n stego-images after embedding data into C_i as G_i $(i = 1, 2, \cdots, n)$, G_i's pixels are represented by g_{iv} $(v = 1, 2, \cdots, |G_i|)$ and the corresponding binary representation is $g_{iv}^7 g_{iv}^6 g_{iv}^5 g_{iv}^4 g_{iv}^3 g_{iv}^2 g_{iv}^1 g_{iv}^0$.

4.1 Image preprocessing using LOCO-I compression

It has been illustrated with Fig. 2 that if a secret image is shared directly using (t, n) threshold scheme, its visual contents may be revealed in the image shares. Moreover, when less than t shares are used for reconstruction, the lacked information may be supplemented by image properties such as two adjacent pixel values are usually similar, which has been utilized by letting $(x_1, f(x_1)) = (x_2, f(x_2))$ in the discussion of the visual security of (2, 2) threshold scheme. It has also been demonstrated that applying simple preprocessing immediately preceded image sharing is not enough yet for visual security (See Fig. 3).

As known to us, image compression is based on the fact that neighboring pixels are highly correlated, and the correlations between pixels can be greatly reduced by image compression, including lossless or lossy methods. Since image sharing is designed to protect important images, the kind of compression methods suitable for preprocessing in image sharing must be lossless. In this paper, LOCO-I compression is used as a preprocessing approach. In order to understand the elimination of the correlations between image pixels, the measure R_{xy} based on the Pearson correlation coefficient is defined as follows.

$$A_{x,y} = \frac{1}{H \times W} \sum\nolimits_{h=1}^{H-x} \sum\nolimits_{w=1}^{W-y} s_{h,w} \tag{5a}$$

$$A'_{x,y} = \frac{1}{H \times W} \sum\nolimits_{h=1}^{H-x} \sum\nolimits_{w=1}^{W-y} s_{h+x,w+y} \tag{5b}$$

$$D_{x,y} = \left[\sum\nolimits_{h=1}^{H-x} \sum\nolimits_{w=1}^{W-y} \left(s_{h,w} - A_{x,y}\right)^2 \right]^{1/2} \tag{5c}$$

$$D'_{x,y} = \left[\sum\nolimits_{h=1}^{H-x} \sum\nolimits_{w=1}^{W-y} \left(s_{h+x,h+y} - A'_{x,y}\right)^2 \right]^{1/2} \tag{5d}$$

$$R_{x,y} = \frac{1}{D_{x,y} \times D'_{x,y}} \sum_{h=1}^{H-x} \sum_{w=1}^{W-y} \left(s_{h,w} - A_{x,y}\right) \left(s_{h+x,w+y} - A'_{x,y}\right) \tag{5e}$$

This measure is applied to the 512 × 512 grayscale Lena image and the LOCO-I compressed Lena image while varying x and y independently from 0 to 4. The values of $R_{x,y}$ are given in Table 1. It is clear that the correlation coefficients have significantly dropped, which may leads to better visual security of image sharing. In addition, for LOCO-I compressed image, incorrect input stream of compression bits will make decompression impossible.

Table 1 Correlation coefficients before and after LOCO-I compression

(a) $R_{x,y}$ of original Lena					
$y \backslash x$	0	1	2	3	4
0	1.000	0.983	0.957	0.931	0.908
1	0.970	0.957	0.937	0.916	0.895
2	0.924	0.915	0.902	0.887	0.872
3	0.881	0.873	0.864	0.853	0.842
4	0.841	0.834	0.826	0.818	0.809
(b) $R_{x,y}$ of LOCO-I compressed Lena					
0	1.000	0.006	0.001	0.003	0.002
1	0.019	0.011	0.009	0.004	-0.005
2	0.023	0.006	0.003	0.006	-0.001
3	0.013	0.006	0.006	-0.003	-0.002
4	0.023	0.008	0.003	0.001	-0.003

Thus, the reconstructed compression data from less than t image shares in a (t, n) threshold scheme may not be successfully decompressed with LOCO-I algorithm, not to mention restoration of the secret image. In the following, the LOCO-I compressed image of a secret image S is denoted by I.

4.2 Generation of image shares

As described in Section 2.1, the evaluation of the polynomials used in (1)–(3) should be based on modulus arithmetic to avoid ambiguity. The prime number $p = 251$, which is the greatest prime number not larger than 255, was recommended to be applied in the calculation of $t - 1$ degree polynomial in the previous works [6, 17, 21, 25]. In these sharing schemes, the pixel values 251–255 of a secret image need to be truncated to 250, which results in a lossy version of image sharing. By using two pixels to represent the grayscale values 251-255, as described in [21], a lossless solution can be obtained. But it may lead to an increase in the size of image shares. To overcome the weaknesses just mentioned, all the calculations in our scheme are done in a power-of-two Galois Field $GF(2^8)$ with the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$ using modular polynomial arithmetic.

Without loss of generality, for the r-th unit U_r of the LOCO-I compressed image I, the coefficient a_k in (1) is chosen to be equal to u_{rk}, and the polynomial is evaluated using mod 2^8 instead of mod p as described above. Therefore, (1) may be rewritten as

$$f_r(x) = u_{r0} + u_{r1}x + \cdots + u_{r,t-1}x^{t-1} \mod 2^8 \tag{6}$$

where the argument x will be replaced by x_i for the i-th participant. The x_i for the i-th participant consists of 8 most significant bits from the first two pixels of the cover image c_i, more specifically, $x_i = (c^7_{i1}c^7_{i2}c^6_{i1}c^6_{i2}c^5_{i1}c^5_{i2}c^4_{i1}c^4_{i2})_2$. For the reason that all x_i must be distinct from one another, if x_{i+1} equals to one of the values $x_1, x_2, \cdots, x_i$, it must be added or decreased by one alternatively to differentiate them as proposed in [17]. It should be noted that the modification of x_i must be synchronized back to the two pixels c_{i1} and c_{i2}, or else the reconstruction of the secret image may fail. The procedure of the generation of image shares is summarized in the following.

Step 1. Compress a secret image S with the use of LOCO-I algorithm and produce the compressed image I.

Step 2. Partition the compressed image I sequentially into m non-overlapping units denoted as U_r $(r = 1, 2, \cdots, m)$. Each unit U_r is composed of t bytes (similar to t pixels of a grayscale image), which are represented by u_{rk} $(k = 0, 1, \cdots, t - 1)$ with the value $u_{rk} \in [0, 255]$.

Step 3. Take the values u_{rk} $(k = 0, 1, \cdots, t - 1)$ of a not-yet-used unit U_r to construct a polynomial as shown (6).

Step 4. Substitute n values of x, i.e. $x_1, x_2, \cdots, x_n$, into (6), a pixel for each of the n image shares is calculated as follows.

$$h_{ir} = f_r(x_i) \mod 2^8 \quad \text{for} \quad i = 1, 2, \cdots, n. \tag{7}$$

Step 5. Repeat Steps 3 and 4 until all units of I are processed.

Step 6. Combine all the pixels h_{ir} $(r = 1, 2, \cdots, m)$ generated in the immediately previous three steps for each image share. Then n image shares denoted by H_i $(i = 1, 2, \cdots, n)$ are obtained.

4.3 Hash-based authentication code

In the potential application of a (t, n) threshold image sharing, the image shares of a secret image may be given out to n participants, or be uploaded to n cloud storage provided by different third-party companies. It is unpractical to assume that each participant or third-party company is honest. If an image sharing scheme doesn't have the ability to authenticate the image shares, the recovery of the secret image may be interfered in the following way.

(1) Image shares H_i may be manipulated incidentally or even deliberately by some of the participants or cloud storage providers, which may prevent successful recovery of secret image no matter how many modifications have been made to the image shares.

(2) Though H_i remains unchanged, the corresponding x_i may be modified, which results in an incorrect pair (x_i, H_i). The image restored using t pairs of (x_i, H_i) with errors will be a false secret image.

(3) In the reconstruction process, an attacker may bring a correct image share along with the matching x_i obtained illegally from an valid participant or cloud storage provider. Together with other $t - 1$ authentic image shares, the attacker can learn the contents of the secret image.

Therefore, it is very helpful to verify the fidelity of all image shares H_i, x_i and the original possession of the pairs (x_i, H_i) before the recovery of the secret image starts.

In our scheme, a cryptographic message authentication code (MAC) based on a hash function such as SHA-512, denoted by HMAC, is used as a means of authentication. With this additional authentication procedure, it may be easy to find out that which one of the above three cases happens (refer to Section 4.5). Let K be a secret key selected by the owner of a secret image or the dealer in the sharing process. In practice, each participant may has a individual ID number, or a cloud storage provider may register its distinct domain name. Let ID_i $(i = 1, 2, \cdots, n)$ denote the identification string corresponding to the i-th participant or cloud storage, and M_{i1}, M_{i2} and M_{i3} denote the HMAC of the image share H_i, x_i and ID_i respectively, that is,

$$\begin{array}{ll} M_{i1} = \mathrm{HMAC}(H_i, K), & \text{for} \quad i = 1, 2, \cdots, n \\ M_{i2} = \mathrm{HMAC}(x_i, K), & \text{for} \quad i = 1, 2, \cdots, n \\ M_{i3} = \mathrm{HMAC}(ID_i, K), & \text{for} \quad i = 1, 2, \cdots, n \end{array} \tag{8}$$

4.4 Dynamic embedding of image shares and authentication code

In order to achieve camouflage effect and provide authentication capability, the image shares H_i generated in Section 4.2 and the hash-based authentication codes (M_{i1}, M_{i2} and M_{i3}) calculated in (8) should be embedded into the cover image C_i. Let B_i denote the total data to be hidden in C_i, and b_{iq} $(q = 1, 2, \cdots, |B_i|)$ denote the binary representation of B_i, we obtain

$$B_i = M_{i1}||M_{i2}||M_{i3}||H_i = (b_{i1}b_{i2}\cdots b_{i,|B_i|})_2 \tag{9}$$

where $|B_i|$ is the total number of bits in B_i.

Unlike most of the available secret image sharing schemes with steganography [6, 17, 25, 27], which substitute statically 2 or 3 least significant bits of each pixel value in a 2 × 2 cover image block with the image shares and authentication bits, dynamic embedding strategy is adopted in our proposed scheme. More specifically, the number of least significant bits of the cover image C_i which will be used for embedding depends on $|B_i|$, that is, the maximum embedding depth E of a cover image pixel is defined in the following.

$$E = \lceil |B_i|/|C_i| \rceil \tag{10}$$

where $\lceil x \rceil$ is the smallest integer not less than x. It can be seen from the above definition in (10) that the value of E is also related to the size of the cover image C_i besides the data B_i to be hidden. In our experiments, E is not larger than 3 and usually equals to 2. In addition, we embed the data B_i randomly into the cover image to further enhance the security of our scheme. The details of the embedding procedure are described as follows.

Step 1. Set $G_i = C_i$,i.e. $g_{iv} = c_{iv}$ $(v = 1, 2, \cdots, |C_i|)$, of course $|G_i| = |C_i|$. It means that the data embedding procedure won't change the size of the cover image C_i.

Step 2. Use the same secret key K as used in (8) to generate a random permutation of the integers $1, 2, \cdots, |C_i|$ denoted by $v_1, v_2, \cdots, v_{|C_i|}$.

Step 3. Compute the embedding depth E by (10). If $E = 1$, replace all the least significant bits $g^0_{i,v1}, g^0_{i,v2}, \cdots, g^0_{i,v|B_i|}$ with $b_{i1}, b_{i2}, \cdots, b_{i,|B_i|}$ respectively; If $E = 2$, first replace all the least significant bits $g^0_{i,v1}, g^0_{i,v2}, \cdots, g^0_{i,v_{|C_i|}}$ with $b_{i1}, b_{i2}, \cdots, b_{i,|C_i|}$ respectively, and then substitute the penultimate bits $g^1_{i,v1}, g^1_{i,v2}, \cdots, g^1_{i,v_{|B_i|-|C_i|}}$ with $b_{i,|C_i|+1}, b_{i,|C_i|+2}, \cdots, b_{i,|B_i|}$ separately, and the steganographic procedure for $E = 3$ is similar.

Step 4. After all bits of B_i are embedded into the cover image C_i, apply an optimal pixel adjustment process as described in Section 2.3 to the stego-image generated in the previous step. Then we obtain the final stego-image G_i $(i = 1, 2, \cdots, n)$.

After applying the above hiding method to embed n image shares into n cover images, n stego-images may be obtained and distributed to n participants or n cloud storage. The framework of the sharing process of our proposed scheme is summarized in Fig. 4.

4.5 Verification and image reconstruction

Any t stego-images can be collected to recover the secret image. Without loss of generality, we assume the t stego-images are G_k provided by t participants P_k $(k = 1, 2, \cdots, t)$ respectively. The reconstruction algorithm is depicted below.

Step 1. Generate an integer sequence by permuting the integers $1, 2, \cdots, |G_k|$ randomly with the secret key K. Based on the same sequence as that used during embedding, the data B_k can be extracted from the corresponding stego-image G_k.

Step 2. According to (9), B_k is divided into four parts M_{k1}, M_{k2}, M_{k3} and H_k. And set $x'_k = (g^7_{k1}g^7_{k2}g^6_{k1}g^6_{k2}g^5_{k1}g^5_{k2}g^4_{k1}g^4_{k2})_2$.

Step 3. According to (8), calculate the message authentication codes M'_{k1}, M'_{k2} and M'_{k3}, more specifically, $M'_{k1} = \text{HMAC}(H_k, K)$, $M'_{k2} = \text{HMAC}(x_k, K)$, and $M'_{k3} = \text{HMAC}(ID_k, K)$. If $M_{k1} = M'_{k1}$, $M_{k2} = M'_{k2}$ and $M_{k3} = M'_{k3}$, continue to the next step. However, if $M_{k1} \neq M'_{k1}$, the case (1) described in Section 4.3 occurs; $M_{k2} \neq M'_{k2}$ means that the case (2) has happened and $M_{k3} \neq M'_{k3}$ indicates the case (3), the recovery process should be stopped.

Step 4. Each pixel h_{kr} $(r = 1, 2, \cdots, |H_k|)$ of the image share H_k and x_k forms a pair. By collecting t pairs of $\{(x_k, h_{kr})\}_{k=1}^{t}$ and substituting them into (3) with the use of mod 2^8 instead of mod p, we get t pixels of the compressed image.

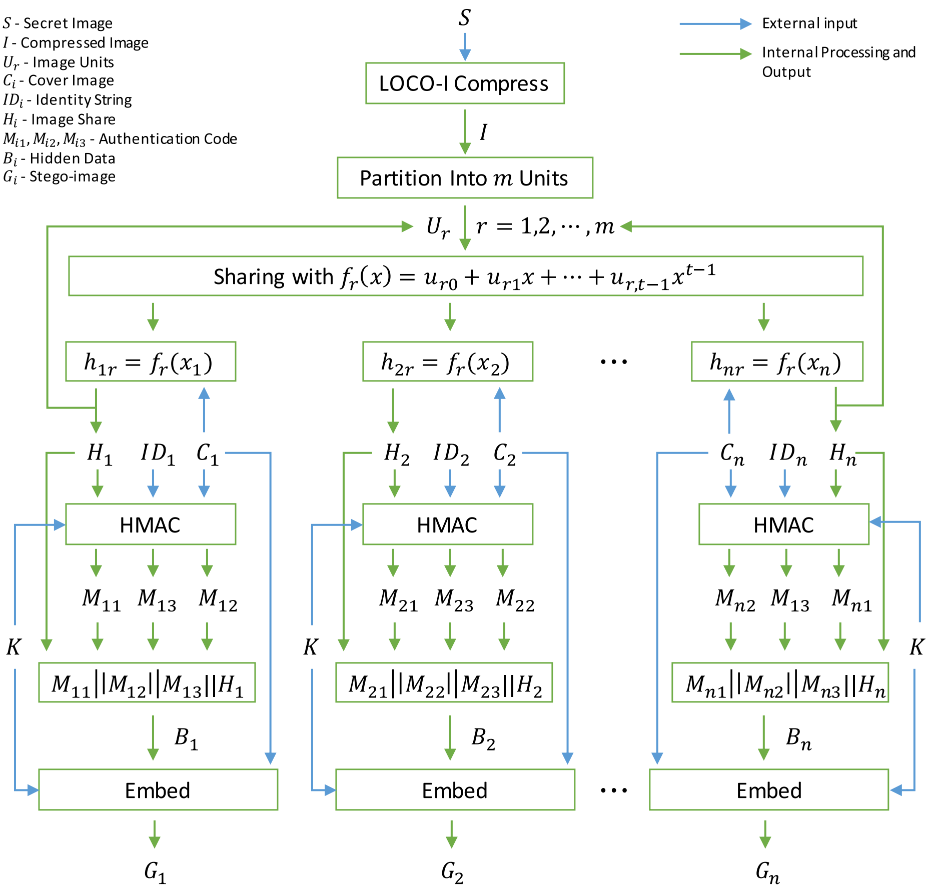


Fig. 4 Framework of the proposed scheme of secure image sharing

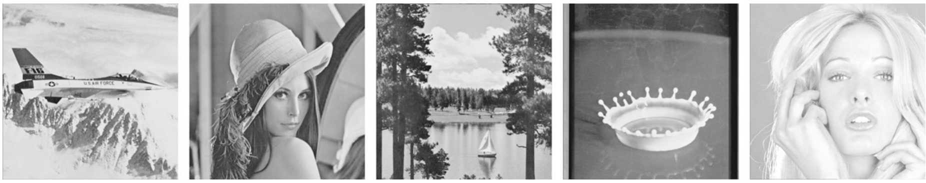
Step 5. When all pixels of H_k have been processed, we get the compressed image I. By applying LOCO-I algorithm to decompress I, the secret image S will be obtained.

5 Experimental results and analysis

In this section, we employ a (2, 4) threshold scheme as an example to demonstrate the effectiveness of the proposed method. Most of the images used in our experiments are downloaded from the well-known USC-SIPI image database (http://sipi.usc.edu/database), which is widely used in the related works. Moreover, experimental results on three other image databases are also given in the final part of this section.

5.1 Size of image shares

In our experiments, five secret images of size 512 × 512 Airplane, Lena, Sailboat, Splash and Tiffany are transformed into 8-bit grayscale images, which are shown in Fig. 5. The size of the image shares generated by the methods of Lin [17], Wang [23],Yang [27],



(a) Airplane (b) Lena (c) Sailboat (d) Splash (e) Tiffany

Fig. 5 512 × 512 grayscale secret images

Chang [6], Eslami [8], Ulutas [22], Li [16] and this paper are given in Table 2. It can be easily seen that the size of our image shares is much smaller than that of the other seven methods. More specifically, the image share's size of our scheme is about a quarter of Lin's and Yang's size, half the size of Chang's, Eslami's, Ulutas's and Li's, and 80 % of Wang's size. In Lin's and Yang's scheme, only one pixel value of the secret image is shared each time by a $t - 1$ degree polynomial. However, t pixel values are shared simultaneously with a $t - 1$ degree polynomial in Chang's, Eslami's, Ulutas's and Li's schemes. And in Wang's scheme, the preprocessing based on Huffman codec leads to smaller size of image shares. In our scheme, t bytes of LOCO-I compressed image are shared with the use of the same $t - 1$ degree polynomial, which results in a great decrease of image shares' size. Small size of image shares would be favorable for their later storage, transmission or other processing.

5.2 Quality of Stego-images

PSNR (Peak Signal-to-Noise Ratio) is commonly used to measure the objective quality of the stego-images. Generally, larger PSNR values mean better image quality. The definition of PSNR is given as follows.

$$\begin{aligned} MSE &= \tfrac{1}{|C_i|} \textstyle\sum_{v=1}^{|C_i|} (G_{iv} - C_{iv})^2 \\ \text{PSNR} &= 10 \times \log_{10}(255^2/MSE) \text{ (dB)} \end{aligned} \tag{11}$$

where C_i and G_i denote the i-th cover image and the corresponding stego-image respectively, and $|C_i|$ is the number of pixels in the cover image C_i. The PSNR values of the stego-images generated by our proposed scheme are compared to those of the stego-images produced by the existing image sharing schemes with steganography such as Lin [17], Yang [27], Chang [6], Eslami [8], Ulutas [22] and Li [16] in our experiments. The secret image and four cover images used in our experiments are shown in Fig. 6. The size of the secret image is 256 × 256, and the size of four cover images is 512 × 512, i.e. four times of the secret image's size, which is required by some of the existing methods for each pixel of

Table 2 Size (Bytes) of image shares

Methods	Lin [17]	Wang [23]	Yang [27]	Chang [6]	Eslami [8]	Ulutas [22]	Li [16]	Ours
Airplane	262144	74571	262144	131072	131072	131072	131072	62186
Lena	262144	81827	262144	131072	131072	131072	131072	69398
Sailboat	262144	92606	262144	131072	131072	131072	131072	81652
Splash	262144	69271	262144	131072	131072	131072	131072	59442
Tiffany	262144	77340	262144	131072	131072	131072	131072	67451

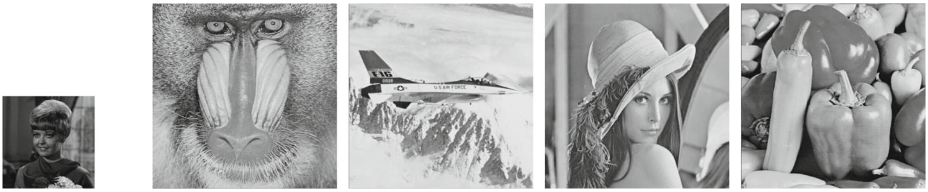

(a) Girl (b) Baboon (c) Airplane (d) Lena (e) Pepper

Fig. 6 A 256 × 256 gray secret image and four 512 × 512 gray cover images

the secret image being embedded into a four-pixels block of the cover images, such as Lin [17] and Yang [27]. The PSNR values are presented in Table 3, from which we can observe that the PSNR values of our proposed method are apparently higher than those of the other methods. Therefore, the quality of our stego-images are much higher than that of the other ones'.

Additionally, the necessary size of cover images used in our proposed scheme may be small, which will be discussed in Section 5.6. For example, when $t = 2$, the requisite size of a cover image is not larger than the size of the secret image. In our experiments, we use four grayscale images Couple, House, Tree and Woman, which are of the same size as a 256 × 256 secret image Bean. The grayscale secret image and the resulting stego-images produced by our sharing method are illustrated in Fig. 7 along with their PSNR values. It can be observed that the PSNR values are still high.

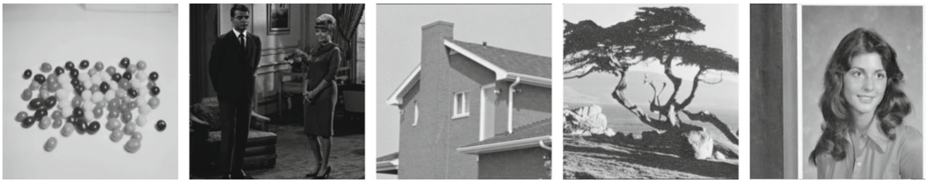
Good PSNR values obtained in our scheme owe much to the small size of image shares. Embedding an image share of smaller size into a cover image means less data will be hidden into the cover image. Fewer modification will thus be made in the cover image during embedding, which may not only lead to higher PSNR values of the corresponding stego-image, but also enhance the camouflage effect of steganographic methods. Moreover, OPAP (see Section 2.3) and dynamic embedding, which will be discussed in the following, are applied in our scheme to further improve the quality of stego-images.

5.3 Dynamic embedding

As described in Section 4.4, dynamic embedding is employed in our scheme. The embedding depth varies depending on the secret image, the threshold value and the size of cover images. In our experiments, we use three secret images (one is 512 × 512 Lena, and the other two are 256 × 256 Girl and Bean), four threshold values ($t = 2$, 3, 4 and 5) and cover images of two different sizes (one is of the same size as the secret image, and the other is four times of the secret image's size). The results are given in Table 4. It can be observed

Table 3 PSNR (dB) of stego-images using a 256 × 256 gray secret image with four 512 × 512 gray cover images

Methods	Lin [17]	Yang [27]	Chang [6]	Eslami [8]	Ulutas [22]	Li [16]	Ours
Baboon	39.18	41.60	40.94	48.13	48.41	48.13	54.00
Airplane	39.23	41.63	40.83	48.14	48.40	48.13	54.00
Lena	39.17	42.44	40.44	48.09	48.40	48.13	54.02
Pepper	39.18	41.38	40.15	48.06	48.40	48.14	54.02



(a) Bean (Secret Image) (b) Couple (PSNR=49.10) (c) House (PSNR=49.12) (d) Tree (PSNR=49.10) (e) Woman (PSNR=49.11)

Fig. 7 Stego-images and their PSNR values using a 256 × 256 gray secret image with four 256 × 256 gray cover images

that the embedding depth is less than 2 most of the time as long as $t > 2$ and the size of cover images is equal to or bigger than that of a secret image.

When the cover image is of the same size as the secret image and the threshold value $t = 2$, 3, or 4, two or three least significant bit-planes of cover image have been modified during embedding, which may bring a small decrease of the stego-images' quality (see Fig. 7 with $t = 2$); when the threshold value $t = 5$, only least significant bits of the pixels in the cover images of two different sizes are used for embedding, which may not introduce noticeable distortions into the cover images. However, in [8], the size of each block is dependent on the size of hidden data and cover images, and when the size of cover images and that of the secret image are identical, the embedding depth of [8] is as large as five, which would bring apparent distortions into the cover images.

When the size of cover images is four times that of the secret image, only the least significant bits of less than 60 % cover image pixels may be modified and the remaining 40 % pixels keep unchanged, which leads to very good quality of stego-images as shown in the last column of Table 3. But in [8], more than one least significant bits need to be modified. And three least significant bits will be modified in [4], in which dynamic programming strategy was used to find an optimal substitution table. The PSNR values obtained in [4] are not higher than 46, which is much lower than ours.

5.4 Security analysis

Our proposed scheme is a (t, n) threshold scheme. If only $t - 1$ authentic stego-images are collected from $t - 1$ of n participants or cloud storage providers, we can obtain $t - 1$ pairs of $(x_k, f_r(x_k))$ as stated in Section 4.5. Without loss of generality, these $t - 1$ pairs

Table 4 Percent (%) of bit-planes 0-3 modified during embedding when different size of cover images and threshold values are used

Secret Image		Lena (512 × 512)						Girl (256 × 256)						Bean (256 × 256)					
Cover Image		512 × 512			1024 × 1024			256 × 256			512 × 512			256 × 256			512 × 512		
Bit-plane		0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
Threshold	2	100	100	11.9	53.0	0	0	100	100	7.0	51.8	0	0	100	30.0	0	32.5	0	0
	3	100	41.4	0	35.3	0	0	100	38.2	0	34.6	0	0	86.8	0	0	21.7	0	0
	4	100	6.1	0	26.5	0	0	100	3.8	0	26.0	0	0	65.3	0	0	16.3	0	0
	5	84.9	0	0	21.2	0	0	83.2	0	0	20.8	0	0	52.4	0	0	13.1	0	0

of $(x_k, f_r(x_k))$ may be denoted as $(x_1, f_r(x_1))$, $(x_2, f_r(x_2))$, $\cdots$, $(x_{t-1}, f_r(x_{t-1}))$. By substituting them into (6), we obtain the following system of linear equations.

$$\begin{cases} f_r(x_1) & = u_{r0} + u_{r1}x_1 + \cdots + u_{r,t-1}x_1^{t-1} \\ f_r(x_2) & = u_{r0} + u_{r1}x_2 + \cdots + u_{r,t-1}x_2^{t-1} \\ & \vdots \\ f_r(x_{t-1}) & = u_{r0} + u_{r1}x_{t-1} + \cdots + u_{r,t-1}x_{t-1}^{t-1} \end{cases} \tag{12}$$

where mod 2^8 is omitted for simplicity. There are t unknown coefficients u_{r0}, u_{r1}, $\cdots$, $u_{r,t-1}$ in the above system of $t-1$ linear equations. It is a underdetermined system and may have 2^8 solutions in the Galois field $GF(2^8)$, that is, the probability of guessing right the t pixels u_{r0}, u_{r1}, $\cdots$, $u_{r,t-1}$ in a unit U_r of the compressed image I is 2^{-8}. Therefore, the probability of successful restoration of the secret image is 2^{-8m}, where m is the number of units in the compressed image I. For example, when the proposed (2, 4) threshold scheme is applied to share Lena image, the compressed image I may have $m = 69398$ units, the probability of recovering the Lena image without errors is $2^{-555184}$, which is very small. In this sense, we may consider the proposed (t, n) threshold based image sharing scheme is secure.

The hash-based message authentication codes generated from the image share H_i, the corresponding value of argument x, and the identity string ID_i of a participant are embedded in company with image shares. The underlying hash function and the second preimage resistant property of MAC guarantee that it is almost impossible to find another H_i', x_i' and ID_i' with the same MACs as the original ones. This make the proposed scheme have the authentication ability of finding out whether a cheat exists or not. Moreover, dynamic embedding with a random strategy may assure that it is hard to correctly extract the hidden data B_i, and thus to find the correct pairs of keyed hash value and its input message. The proposed scheme offers low possibilities of a brute-force attempt to determine the MAC key with known text-tag pairs. Therefore, any modification of the image shares and the values of argument x, or a false identity will be discovered during the recovery process.

In addition, as described in Section 4.1, LOCO-I compression is employed as a preprocessing procedure before image sharing. And it is demonstrated that the correlations between the neighboring secret image pixels drop greatly, which may be of little help in finding the solution of the system of $t-1$ linear equations as shown in (12). Also, a false restored compressed image I from less than t image shares may be unable to be decompressed with the use of LOCO-I algorithm, not to mention recovering the secret image.

5.5 Computation cost

In our experiments, our proposed method and six existing image sharing schemes with steganography are employed to share a secret image and then embed the generated image shares into four cover images shown in Fig. 6. A PC running Windows 10 with 3.2GHz CPU E5800 and 4GB memory is used for all the experiments. And all the codes are developed with the use of OpenCV (http://www.opencv.org) and Crypto++ (http://www.cryptopp.com). The mean processing time and peak memory usage in 1000 times running are given in Table 5.

Owing to the smaller number of evaluations of hash function, the computational speed of the proposed method is faster than four of the existing methods, including Yang's, Chang's, Eslami's and Ulutas's methods. But it is slightly slower than Lin's and Li's method. In Lin's

Table 5 Comparison of peak memory usage and computation time

Methods	Lin [17]	Yang [27]	Chang [6]	Eslami [8]	Ulutas [22]	Li [16]	Ours
Memory (KB)	11431	11466	11443	11710	11704	11622	13264
Time (ms)	36.5	306.3	187.7	183.3	237.8	32.2	66.8

scheme, simple even or odd parity is used; and only 128 check bits for 128 stego blocks are calculated in Li's scheme.

And it can be observed that the peak memory usage in our scheme is a little higher than that of the related works. The likely reason may be that LOCO-I compression is added as a preprocessing procedure in our scheme, which may bring an effect on our scheme's efficiency. However, compared to the features benefited from LOCO-I compression, the small negative impact on the computation cost may be acceptable.

5.6 Comparison of features

In order to present an overview of the difference between the known image sharing schemes and ours, we give the feature comparisons in Table 6. The properties used for comparing are described in the following:

- RCTS: It is defined as the ratio of the cover image size to the secret image size. As steganography is concerned, larger size of images used as cover media may easily arouse the attackers' attention and increase the amount of data to be transmitted or stored. Therefore, it would be favorable that the value of RCTS is small.
- PSNR: It measures the quality of stego-images in terms of PSNR (dB). High values generally indicate good stego-image quality.

Table 6 Features comparison of different schemes

Property	Lin [17]	Wang [23]	Yang [27]	Chang [6]	Eslami [8]	Ulutas [22]	Li [16]	Ours
RCTS	4	-	4	$\frac{4}{t}$	$> \frac{2}{t}$	$\geq \frac{6}{t}$	$\frac{c}{t \times s}^{\diamond}$	$\leq \frac{2}{t}$
PSNR	LOW	-	LOW	LOW	MIDDLE	MIDDLE	MIDDLE	HIGH
DEPTH	3	-	3	3	2	3	$\lceil \frac{8 \times s+1}{c} \rceil^{\diamond}$	Table 4
AuthWho	NO	NO	NO	NO	NO	NO	NO	Yes
AuthModi	≈ 1	NO	$(\frac{1}{2})^{B^{\dagger}}$	$(\frac{1}{16})^{B^{\dagger}}$	$(\frac{1}{16})^{B^{\dagger}/2}$	$(\frac{1}{16})^{B^{\dagger}}$	$(\frac{1}{2})^{128 \times S^{\circ}}$	$(\frac{1}{2})^{W^{\ddagger}}$
ShSize	1	$\frac{CR}{t}^{*}$	1	$\frac{1}{t}$	$\frac{1}{t}$	$\frac{1}{t}$	$\frac{1}{t}$	$\frac{CR}{t}^{*}$
HashNum	$\propto$ ShSize	-	$\propto$ ShSize	$\propto$ ShSize	$\propto$ ShSize	$\propto$ ShSize	$\propto$ ShSize	3
Preprocess	-	Huffman Codec	-	-	-	-	Encryption	LOCO-I

-means not available

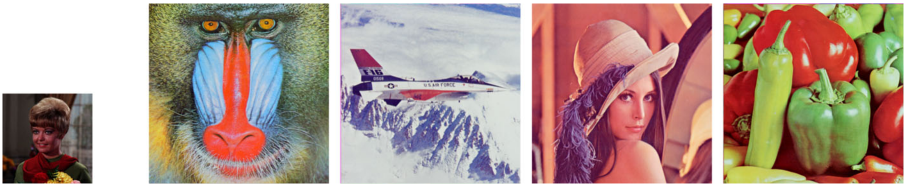
$^{\diamond}(s, c)$ hiding method means a s-pixel block of the image share is embedded in a c-pixel block of the cover image

$^{\dagger}B$ is the number of stego-image blocks modified by an attacker

$^{\circ}S$ is the number of sections, which is composed of 128 s-pixel blocks, altered by an attacker

$^{\ddagger}W$ is the length of the hash code, e.g. if SHA-512 is used, $W = 512$

$^{*}CR$ is the ratio of the size of compressed secret image to that of the original one



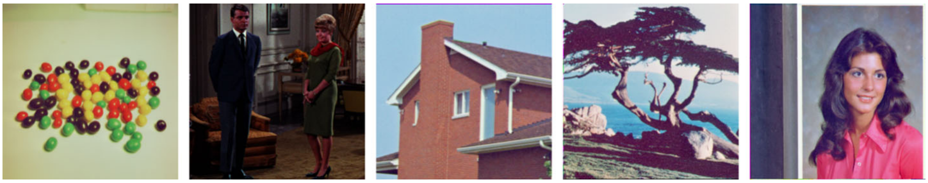
(a) Girl (Secret Image) (b) Baboon (PSNR=53.61) (c) Airplane (PSNR=53.62) (d) Lena (PSNR=53.62) (e) Pepper (PSNR=53.62)

Fig. 8 Stego-images and their PSNR values using a 256 × 256 color secret imagewith four 512 × 512 color cover images

- DEPTH: It is the maximum number of least significant bit-planes of a cover image will be altered during image shares' embedding. Small embedding depth may lead to less modification to cover images.
- AuthWho: It means that the scheme have the capability to judge whether a participant is the intended one to whom an image share is originally distributed.
- AuthModi: It stands for the probability that a modified image share can not be detected when it is provided to restore the secret image. Lowe values suggest strong authentication ability.
- ShSize: It represents the ratio of the size of image share to that of the secret image. Low ratios would be preferred for later embedding.
- HashNum: It is the number of calculations of parity or hash function needed for an image share. Large values may increase the computational cost.
- Preprocess: It describes how a secret image is processed before sharing. As illustrated in Section 3, if an image is shared directly without preprocessing, the image shares may disclose perceptual information about the secret image.

5.7 Application to various images

Our proposed scheme may be easily extended to share color images. For a color secret image, each of its three components in RGB color space may be taken as a gray image and shared after LOCO-I compression respectively, and then embedded into the corresponding color components of color cover images. In our experiments, a 256 × 256 color secret image shown in Fig. 8(a) is shared into image shares and then hidden into four 512 × 512 color cover images respectively with the proposed scheme. The generated stego-images are illustrated in Fig. 8(b)–(e) in company with their PSNR values. Moreover, the experimental



(a) Bean (Secret Image) (b) Couple (PSNR=48.69) (c) House (PSNR=48.70) (d) Tree (PSNR=48.69) (e) Woman (PSNR=48.67)

Fig. 9 Stego-images and their PSNR values using a 256 × 256 color secret image with four 256 × 256 color cover images

Table 7 PSNR (dB) of stego-images in different image databases (The resolution of the secret image and cover images are identical in each experiment)

Image Database	BOWS-2	CorelDraw	Dresden		
Resolution	512 × 512	768 × 512	2560 × 1920	3072 × 2304	4000 × 3000
Color Depth	8-bit Grayscale	24-bit Color	24-bit Color	24-bit Color	24-bit Color
PSNR	46.84	46.38	48.25	48.04	47.61

results using color cover images of the same size as that of the secret image are illustrated in Fig. 9. From the two figures, it can be seen that there are no visible distortions in both cases. And all of the color stego-images are of good quality in terms of PSNR, especially when the size of cover images is four times that of the secret image.

Furthermore, in order to investigate the suitability of the proposed scheme, we apply our method to share images in three other image databases BOWS-2 (http://bows2.ec-lille.fr), CorelDraw (http://www.corel.com) and Dresden Image Database [9]. The images in these databases are of different color depth and resolution. In each experiment, a secret image and four cover images are randomly selected from an image database. We conducts 80 experiments and 400 images of a specific resolution are chosen at random in total. The mean PSNR values of stego-images of the same resolution in each image database are given in Table 7. It showed the good performance of our new method with color or gray images of different resolutions.

6 Conclusion

In this paper, we present a novel (t, n) image sharing scheme. Experimental results and security analysis demonstrated the performance of our scheme. Any t or more image shares can be used to restore the secret image in a lossless manner, but arbitrarily $t - 1$ (or less) image shares cannot get sufficient information to reveal the secret image. Visual security of the proposed scheme is greatly improved by using LOCO-I compression as a preprocessing approach. And the size of image shares is small, which leads to a decrease in requisite size of cover images and an improvement on the quality of stego-images. Moreover, three types of cheating may be discovered by embedding the hash-based message authentication codes of the image shares, the value of argument x and the identity string of a participant or cloud storage provider into cover images together with image shares. Also dynamic embedding with a random strategy is employed to further enhance the security of our scheme.

Acknowledgments The authors thank the anonymous reviewers for their insightful comments and suggestions. This work was partially supported by the 973 Program (No. 2014CB360501), the National Natural Science Foundation of China (No. U1135004 and No. 61170080), Guangdong Provincial Natural Science Foundation (No. 2014A030308006), and Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2011).

References

1. Alvarez G, Hernández Encinas L, Martín del Rey A (2008) A multisecret sharing scheme for color images based on cellular automata. Inf Sci 178(22):4382–4395. doi:10.1016/j.ins.2008.07.010

2. Blakley G (1979) Safeguarding cryptographic keys. In: Proceedings of national computer conference (AFIPS1979), vol 48, pp 313–317
3. Chan CK, Cheng LM (2004) Hiding data in images by simple LSB substitution. Pattern Recogn 37(3):469–474
4. Chan CS, Sung PE (2010) Secret image sharing with steganography and authentication using dynamic programming strategy. In: First International Conference on Pervasive Computing Signal Processing and Applications (PCSPA), pp 382–385. doi:10.1109/PCSPA.2010.98
5. Chan CS, Lin CY, Lin YH (2012) Apply the modulus function to secret image sharing. International Journal of Innovative Computing Information and Control 8(1A):375–385
6. Chang CC, Hsieh YP, Lin CH (2008) Sharing secrets in stego images with authentication. Pattern Recogn 41(10):3130–3137
7. Chen CC, Wu WJ, Chen JL (2015) Highly efficient and secure multi-secret image sharing scheme. Multimedia Tools and Applications:1–16
8. Eslami Z, Ahmadabadi J (2011) Secret image sharing with authentication-chaining and dynamic embedding. J Syst Softw 84(5):803–809
9. Gloe T, Bhme R (2010) The 'Dresden Image Database' for benchmarking digital image forensics. In: Proceedings of the 25th symposium on applied computing (ACM SAC 2010), vol 2, pp 1585-1591
10. Guo C, Chang CC, Qin C (2012) A hierarchical threshold secret image sharing. Pattern Recogn Lett 33(1):83–91
11. Hsieh SL, Tsai IJ, Yeh CP, Chang CM (2011) An image authentication scheme based on digital watermarking and image secret sharing. Multimedia Tools and Applications 52(2–3):597–619
12. Islam N, Puech W, Hayat K, Brouzet R (2011) Application of homomorphism to secure image sharing. Opt Commun 284(19):4412–4429
13. Jin J, hong Wu Z (2012) A secret image sharing based on neighborhood configurations of 2-D cellular automata. Opt Laser Technol 44(3):538–548
14. Khosravi M, Naghsh-Nilchi A (2014) A novel joint secret image sharing and robust steganography method using wavelet. Multimedia Systems 20(2):215–226
15. Lee CW, Tsai WH (2012) A secret-sharing-based method for authentication of grayscale document images via the use of the PNG image with a data repair capability. IEEE Trans Image Process 21(1):207–218
16. Li P, Kong Q, Ma Y (2014) Image secret sharing and hiding with authentication based on PSNR estimation. Journal of Information Hiding and Multimedia Signal Processing 5(3):353–366
17. Lin CC, Tsai WH (2004) Secret image sharing with steganography and authentication. J Syst Softw 7(3):405–414
18. Lin PY, Chang CC (2011) Cheating resistance and reversibility-oriented secret sharing mechanism. IET Inf Secur 5(2):81–92
19. Naor M, Shamir A (1995) Visual cryptography. In: Advances in cryptology - EUROCRYPT'94, Perugia, Italy, LNCS, vol 950, pp 1–12
20. Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613
21. Thien CC, Lin JC (2002) Secret image sharing. Comput Graph 26(5):765–770
22. Ulutas G, Ulutas M, Nabiyev VV (2013) Secret image sharing scheme with adaptive authentication strength. Pattern Recogn Lett 34(3):283–291
23. Wang RZ, Su CH (2006) Secret image sharing with smaller shadow images. Pattern Recogn Lett 27(6):551–555
24. Weinberger M, Seroussi G, Sapiro G (2000) The LOCO-i lossless image compression algorithm: Principles and standardization into JPEG-LS. IEEE Trans Image Process 9(8):1309–1324
25. Wu CC, Kao SJ, Kuo WC, Hwang MS (2008) Enhance the image sharing with steganography and authentication. In: Proceedings of international conference on intelligent information hiding and multimedia signal processing (IIHMSP'08). Harbin, China, pp 1177–1181
26. Yang CN, Chu YY (2011) A general (k, n) scalable secret image sharing scheme with the smooth scalability. J Syst Softw 84(10):1726–1733
27. Yang CN, Chen TS, Yua KH, Wang CC (2007) Improvements of image sharing with steganography and authentication. J Syst Softw 80(7):1070–1076
28. Yuan HD (2014) Secret sharing with multi-cover adaptive steganography. Inf Sci 254:197–212

Junhui He received the B.S., M.S., and Ph.D. degrees from Central South University, South China University of Technology, and Sun Yat-Sen University, China, in 1997, 2000, and 2006, respectively. He is currently an associate professor with the School of Computer Science and Engineering, South China University of Technology. His current research interests include multimedia security, steganography and steganalysis.

Weiqiang Lan received the B.S. degree in software engineering from South China Agricultural University, China, in 2013. He is currently studying for M.S. degree in computer science at South China University of Technology. His current research interests include multimedia security and application.

Shaohua Tang received the B.S. and M.S. degrees in applied mathematics, and the Ph.D. degree in communication and information system all from the South China University of Technology, in 1991,1994, and 1998, respectively. He was a visiting scholar with North Carolina State University, and a visiting professor with the University of Cincinnati, Ohio. He has been a full professor with the School of Computer Science and Engineering, South China University of Technology since 2004. His current research interests include information security, networking, and information processing. He is a member of the IEEE and the IEEE Computer Society.