Image Tamper Detection and Recovery Based on Dual Watermarks Sharing Strategy

Yi-Hui Chen[1], Chin-Chen Chang[2]
[1] Department of Applied Informatics and Multimedia
Asia University, Taichung, Taiwan 41354, R.O.C.
[2] Department of Information Engineering and Computer Science
Feng Chia University
Taichung 401724, Taiwan, R.O.C
chenyh@asia.edu.tw, ccc@cs.ccu.edu.tw

Journal of Digital Information Management

ABSTRACT: *Recently, verifiable secret sharing schemes have been proposed to keep participants from providing fake or illegal stego-images. However, the schemes do not consider the recovery mechanism when parts of the information in stego-images are lost or incidentally modified during the transmission process. This paper presents a novel verifiable and reversible secret sharing scheme based on Shamir's scheme, which can authenticate the shares in advance and then recover the inauthentic parts to reconstruct the original secret image. The experimental results provided positive results for the feasibility of the proposed scheme.*

Categories and Subject Descriptors:
I.4.10 [Image Representation]; D.4.6 [Security and Protection]: Authentication

General Terms: Steganography, Information Hiding, Image Tamper Detection, Dual Watermarking

Keywords: Secret sharing, Steganography, Authentication, Recovery

Received: 10 August 2011, Revised 30 September 2011, Accepted 4 October 2011

1. Introduction

Secret sharing, a so-called a (*t*, *n*)-threshold secret sharing scheme, is a technique which disperses secrets into *n* shares and then requires at least *t* shares to reconstruct the original secrets, where $t \leq n$. In other words, no information about it can be obtained as long as there are *t*-1 or fewer participants for joining in the reconstructing procedure. This concept was first proposed by Shamir in 1979 [12], who defined the idea of visual cryptography used for digital images [11]. The scheme applies the Xeroxing technique to distribute the secret image into several transparencies and then to stack them so it can be retrieved with the human eyes without any computations. Following Shamir's method, several studies [1, 7, 9, 13, 14] used the concept to provide related visual cryptography techniques with better performances.

Unfortunately, the image retrieved in this process is different from the original one that it is unsuited for use in some sensitive applications, such as military and medical image processing, domains in which any distortion will result in intolerable errors in evaluation. Moreover, when the shares are transmitted over the Internet, they are meaningless, seen as noise-like images suspicious to censors.

Several ideas for hiding shares in meaningful content have been proposed with steganographic techniques, in which the secret image can be fully reconstructed as presented in schemes [2, 4, 6, 14, 15]. Steganography is the art of hiding data to convey secrets behind the cover image and avoid arousing suspicion. In certain applications, there is a risk that wrong secret is obtained because the shares might be lost incidentally or modified intentionally. Therefore, many related studies have proposed ways to import authentication mechanisms by using a fragile watermark to verify the fidelity of all shares before the secret is reconstructed. When all the shares successfully pass the verification, the secret image should be completely reconstructed; however, if any of the shared data is inauthentic, the secret image will never be obtained.

In 2004, Lin and Tsai [8] proposed a secret sharing scheme with steganography and authentication mechanisms to hide the shares inside meaningful content. This approach uses the Shamir's method to hide the secret as constants in (*t*-1)-degree polynomials to build a (*t*, *n*) image secret sharing scheme. In the experiments, Yang et al. [16] mentioned three weaknesses of scheme [8]: (1) the method slightly distorts the quality of the stego-image, (2) it has a weak authentication mechanism in that the verification can be erroneous when the dishonest participants provide fake stego-images in which the authentication codes are made up by complying with the parity checking rule, and (3) it is a lossy polynomial-based

secret sharing scheme because the pixel values in the secret image must be truncated while the values are greater than 250.

To conquer such defects, Yang et al. [16] proposed an improved version that keeps the visual quality better but also dissuades unintended participants. Although Yang et al.'s method was feasible, Chang et al. [5] found in it two shortcomings: (1) it may slightly distort the visual quality of the stego-image, and (2) the weak authentication proposed can result in a fake stego image being passing the authentication process. To overcome these drawbacks, Chang et al. [5], based on the Chinese remainder theorem (CRT) combined the steganographic and authentication techniques to create a scheme that significantly promotes the visual quality of stego-images and enhances the ability to authenticate. In schemes [5, 8, 16], while parts of the information in the stego-images are lost during transmission over the Internet, the stego-images judged as inauthentic will cause the secret image never been obtained. With this safeguard, attackers could simply modify all the stego-images during the transmission process to obstruct the secret image extracting.

Recovery is another important issue that deals with recovering the secret image when parts of data are inauthentic as result of incidental loss during the transmission process or intentional tampering. In 2011, Chang et al. [2] proposed a meaningful secret-sharing scheme that included both authentication and remedy abilities to allow the detection of corrupted areas and to repair the secret image. However, the method cannot recover the secret image completely when it has been damaged by losing some bits during transfer. This is an important issue, because complete recovery is required in many domains, such as medical, military, and art applications, in which no distortions can be tolerated. In this paper, we proposed a novel secret sharing scheme with steganography techniques including both authentication and recovery abilities. The merits of the proposed scheme are that (1) the visual qualities of stego-images are higher than schemes [5, 8, 16] and do not create visually perceptible changes so invaders remain unaware of the existence of the secret, and (2) even if the parts of the information in the share are lost, the secret image can be completely reconstructed on the premise that all the shares are from licit participants.

The rest of this paper is organized as follows. Section 2 briefly reviews the literature related to the Shamir's method for secret sharing. Section 3 demonstrates our secret sharing, authentication, reconstruction and recovery procedures, along with an example. The experimental results and several evaluations are conducted in Section 4. Finally, conclusions are drawn in the last section.

2. Related Works

The Related Works section introduces two schemes, i.e., traditional (t, n)-threshold secret sharing and previous works [2, 5, 8, 16].

2.1 The (t, n)-threshold secret sharing scheme

The (t, n)-threshold secret sharing, proposed by Shamir, where t is always less than n. In the scheme, the secret is treated as the parameter r_1 and the other parameters $r_2, r_3, \ldots, r_t$ are chosen at random to construct a $(t-1)$-degree polynomial, shown as Equation (1):

$$R(x) = r_1 + r_2 \text{ x } x + r_3 \text{ x } x^2 + \ldots + r_t \text{ x } x^{t-1} \quad (1)$$

where the value of x is depicted as x_i for the share being dispatched to the i^{th} participant, and all x_i's are exclusive from each other. Thus, n participants will build n $(t-1)$-degree polynomials, as shown in Equation (2):

$$\begin{aligned} R(x_1) &= r_1 + r_2 \text{ x } x_1 + r_3 \text{ x } x_1^2 + \ldots + r_t \text{ x } x_1^{t-1} \\ R(x_2) &= r_1 + r_2 \text{ x } x_2 + r_3 \text{ x } x_2^2 + \ldots + r_t \text{ x } x_2^{t-1} \\ &\vdots \\ R(x_t) &= r_1 + r_2 \text{ x } x_n + r_3 \text{ x } x_n^2 + \ldots + r_t \text{ x } x_n^{t-1} \end{aligned} \quad (2)$$

With the polynomial interpolation as shown in Equation (3), any t participants attend to the constructing procedure, and the reconstructed $(t-1)$-degree polynomial will be generated, so the unknown messages r_1 can be finally resolved.

$$\begin{aligned} R(x) = R(x_1) &\left(\frac{x - x_1}{x_1 - x_2}\right)\left(\frac{x - x_3}{x_1 - x_3}\right) \cdots \left(\frac{x - x_t}{x_1 - x_t}\right) \\ + R(x_2) &\left(\frac{x - x_1}{x_2 - x_1}\right)\left(\frac{x - x_3}{x_2 - x_3}\right) \cdots \left(\frac{x - x_t}{x_2 - x_t}\right) \\ + \ldots + R(x_t) &\left(\frac{x - x_1}{x_t - x_1}\right)\left(\frac{x - x_3}{x_t - x_2}\right) \cdots \left(\frac{x - x_t}{x_t - x_{t-1}}\right) \end{aligned} \quad (3)$$

2.2 Previous works

Lin and Tsai [8] proposed a meaningful secret sharing scheme to embed the shares into cover images. Their scheme limits the pixel values of the secret image to the range of 0 to 250 because they used the Galois Field GF(251) to define their proposed secret sharing formula. If the value of x is greater than 250, the pixel x must be replaced with 250, which degrades the image quality of the reconstructed secret image. In addition, the generation of the authenticated code is insecure because the authentication can be guessed easily by attackers. In 2007, Yang et al. [16] provided a better version that incorporated two improvements: (1) the secret image can be reconstructed completely by using GF(28) instead of GF(251) to define the secret sharing formula and (2) the authentication code is generated by the robustness function HMAC (hashed-based message authentication code). In 2008, Chang et al. [5] found that Yang et al.'s scheme had the weakness of producing imprecise authentication results. To reduce the effect of this weakness, Chang et al. used a Chinese remainder theory-based (CRT-based) approach to generate the authentication code and used Thien and Lin's scheme [14] to define the secret sharing formula.

Although the schemes [5, 8, 16] provide an authentication mechanism that verifies whether the shares are authentic, no remedy was provided for the cases in which some stego-image information has been tampered with or was lost. To provide a remedy, Chang et al. [2] imported a mechanism for repairing the inauthentic area in a secret image. However, the inauthentic area cannot be reconstructed completely, so it still does not meet the requirements of special domains in which no distortions can be allowed.

3. The Proposed Scheme

The proposed scheme is divided into three procedures: (1) the secret sharing procedure, which portrays how the secret image can be shared for any three participants, (2) the authentication and reconstruction procedure, which evaluates whether the participant shares are valid, after which determination any three licit participants can completely restore the original secret image, and (3) the recovery procedure, which demonstrates how the inauthentic area is completely recovered.

3.1 Secret Sharing Procedure

The secret sharing procedure, as shown in Figure 1, contains three phases: the initial phase, share and authentication code generation, and the hiding phase. The procedure for the initial phase is to input the secret image and cover image in such a way that the pixels in the secret image can look for their mapping pixels. Let the pixel and its mapping pixel be treated as a partner-pixel pair. In the share and authentication code generation phase, the partner-pixel pair are inputted one-by-one, and then the generated share and authentication code are hidden in the pixels of the cover image illustrated in the hiding phase to output the stego-pixels. The secret sharing procedure will be terminated only when all partner-pixel pairs are processed. Details of each phase are portrayed as follows.

p_id, where $p\_id \in [1, N]$, and N is the total number of pixels in the secret image. Then, the p_id in secret image is shuffled by applying the pseudo random number generator (PRNG) with a secret key K_1. The p_id in the secret image after shuffling can be treated as a look-up table, which records the shuffled results. For clarity, the $p_id$without shuffling is called an original table. Next, a partner-pixel pair is composed to contain two pixels which are with p_id at the same position in the original and look-up tables. An example of how to find the partner-pixel pairs is shown in Figure 2, where N=16. Figure 2(a) is the original table and Figure 2(b) is the look-up table of 2(a), and pixels with p_id in the original table and the look-up table are named “original pixels” and “mapping pixels,” respectively. In this example, as for the upper left corner, the mapping address of pixel p_id1 is 9, so that pixels with p_id 1 and 9 are treated as a partner-pixel pair. To clarify this concept for later use, some notations are identified in advance as follows. We assume that the secret image is the size of N, the number of partner-pixel pairs is N, and a partner-pixel pair is depicted as G_λ, where λ indicates the λ^{th} partner-pixel pair and $1 \le \lambda \le N$. The pixels in G_λ obtained from the original table and the look-up table are denoted as G_λ^1 and G_λ^2, respectively.

Another important issue described in the initial phase is that it divides a cover image into several blocks of 2 × 2 pixels, also called four-pixel blocks and depicted as p_k, where k denotes the k^{th} four-pixel block in the cover image and $1 \le k \le N$. A p_k is used for embedding authentication codes and shares that will be described thereafter. The pixels located in the four-pixel block p_k are presented as p_k^1, p_k^2, p_k^3 and p_k^4, respectively, as illustrated in Figure 3.

In the second phase, pixels G_λ^1 and G_λ^2 in the partner-pixel pair G_λ are first transformed into 7-based notations and

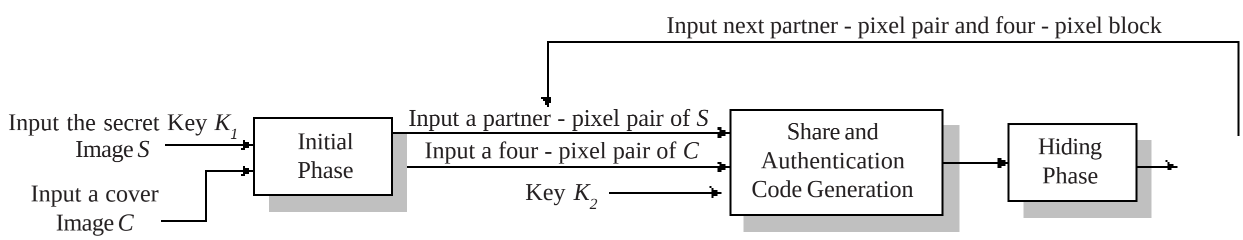


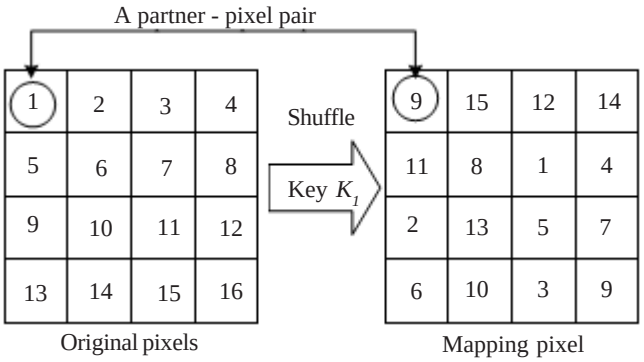
Figure 1. The flowchart of secret sharing procedure

depicted as $(a_\lambda^\mu, b_\lambda^\mu, c_\lambda^\mu)_7$, where $\mu \in \{1, 2\}$, and μ indicates the μ^{th} item in G_λ. Later, the notations a_λ^μ, b_λ^μ, and c_λ^μ are applied to generate a formula as defined in Equation (4).

$$R_\lambda^\mu(x_i) = a_\lambda^\mu + b_\lambda^\mu x_i + c_\lambda^\mu x_i^2 \bmod 7, \qquad (4)$$

where x_i is the ID of the i^{th} participant, and μ and λ indicate the Equation (4) made from the μ^{th} item of the λ^{th} partner-pixel pair. For example, a partner-pixel pair contains two pixels, and , the value of is 156 and that of is 202, and the ID of the participant is 5 (i.e., x_i = 5). The values and are re-expressed by applying 7-based notations as $(312)_7$ and $(406)_7$, respectively. In other words, the values of $a_\lambda^1, b_\lambda^1, c_\lambda^1, a_\lambda^2, b_\lambda^2$ and c_λ^2 are 3, 1, 2, 4, 0 and 6, respectively.

Therefore, two formulas are built through Equation (4) as $R_\lambda^1(x_i) = 3 + 1x + 2x^2 \bmod 7$ and $R_\lambda^2(x_i) = 4 + 0x + 6x^2 \bmod 7$. After inputting the value of x_i, values of $R_\lambda^1(5)$ and $R_\lambda^2(5)$ are calculated as 2 and 0, respectively. Next, the authentication code generator produces the authentication code $\tilde{A}_\lambda(x_i)$ for the λ^{th} partner-pixel pair with Equation (5).



(a) Original table (b) Look-up table

Figure 2. The flowchart of secret sharing procedure

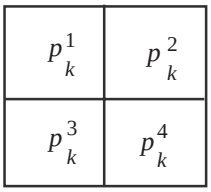


Figure 3. Illustration of a four-pixel block

$$A_\lambda(x_i) = HMAC(k \| R^1_\lambda(x_i) \| x_i),$$

$$\tilde{A}_\lambda(x_i) = LSB_3(HMAC(A_\lambda(x_i) \| R^2_\lambda(x_i) \| x_i)) \quad (5)$$

where k is represented as the ID of the current four-pixel block, $R^1_\lambda(x_i)$ and $R^2_\lambda(x_i)$ are computed with Equation (4), x_i is the ID of participant and the LSB_3 function gets three LSBs of the value returned from the HMAC (key-hashing for message authentication code) function. The HMAC function uses a specific algorithm that includes a cryptographic hash function in combination with a secret key K_2 to generate a message authentication code. Subsequently, authentication code $\tilde{A}_\lambda(x_i)$, and values of $R^1_\lambda(x_i)$ and $R^2_\lambda(x_i)$ are translated into a decimal as *embed_num* by using multiple-base notational system through Equation (6), for which the notation is depicted as $(\sigma_3, \sigma_2, \sigma_1)$, where $0 \le \sigma_w < b_w$ for $w = 1, 2$ and 3. Therefore, notation $((R^1_\lambda(x_i), R^2_\lambda(x_i), \tilde{A}_\lambda(x_i))_{7,7,8}$ can be calculated as $embed\_num = R^1_\lambda(x_i) \times 7 \times 8 + R^2_\lambda(x_i) \times 7 + \tilde{A}_\lambda(x_i)$.

$$embed\_num = \sum_{w=2}^{3}(\sigma_w \times \prod_{j=1}^{w} b_j) + \sigma_1 \quad (6)$$

Finally, in the hiding phase, the value of *embed_num* will be hidden into an inputted four-pixel block p_k. Before the message is hidden, the value *embed_num* must be re-expressed into a 5-based notation as $(e^1_\lambda, e^2_\lambda, e^3_\lambda, e^4_\lambda)_5$; then notations e^1_λ, e^2_λ, e^3_λ and e^4_λ are hidden data to conceal into the pixels p^1_k, p^2_k, p^3_k and p^4_k in the four-pixel block p_k, respectively. The pseudo code of embedding the algorithm uses Java language and is as shown in Figure 4. In the algorithm, the hidden message e^α_λ is hidden in the pixel p^α_k, for $\alpha = 1, 2, .., 4$. The notation s is a temporary value used for later updating the pixel value p^α_k to be $\hat{p}^\alpha_k$, in which $\hat{p}^\alpha_k$ is the so-called stego-pixel.

```
Hiding method (int p^α_k , int e^α_λ ) {
  s = p^α_k % 5;        // p^α_k mod 5
  if ((e^α_λ - s ) > 2)
     p̂^α_k = p^α_k - (5 - (e^α_λ - s));
  else if ((e^α_λ - s ) < -2 )
     p̂^α_k = p^α_k + ( 5 + (e^α_λ - s ));
  else
     p̂^α_k = p^α_k + (e^α_λ - s );

 //The stego-pixel value is limited to range from 0 to 255
  if ( p̂^α_k > 255)
     p̂^α_k = 255 - (5 - e^α_λ);
  if ( p̂^α_k < 0)
     p̂^α_k = e^α_λ;
}
```

Figure 4. The pseudo code for embedding method

3.2 Authentication and Reconstruction Procedure

In this procedure, to prevent a legal participant incidentally bringing an erroneous stego-image or an attacker from intentionally providing a fake image or a false key for trying to join the reconstructing procedure, the proposed method includes an authentication mechanism to determine in advance whether participants are authentic. The main purposes of this procedure are broken into two points, the first one to authenticate whether the shares are valid before reconstructing the secret image, and the second to reconstruct the secret image. The flowchart of this procedure, which consists of the initial phase, the extraction and authentication phase, and reconstruction phase, is shown in Figure 5.

The proposed scheme is that any three legal stego-images can be used to reconstruct the original secret image. Therefore, three stego-images are input into the initial phase. Subsequently, each stego-image is divided into several blocks of 2×2 pixels. Next, three four-pixel blocks from three distinct stego-images are input into the extraction and authentication phase. Because the extracting and authentication phase for each of input four-pixel blocks is the same, we provide only one sample here to describe how to extract and authenticate the hidden information.

In the extracting and authentication phase, we assume that the pixels in a four-pixel block are depicted as $\hat{p}^1_k$, $\hat{p}^2_k$, $\hat{p}^3_k$, and $\hat{p}^4_k$, where k is the k^{th} four-pixel block in stego-image. The extracting hidden information from pixels $\hat{p}^1_k$, $\hat{p}^2_k$, $\hat{p}^3_k$ and $\hat{p}^4_k$ are denoted as e^1_λ, e^2_λ, e^3_λ and e^4_λ, respectively, through Equation (7), where the extracted data e^1_λ, e^2_λ, e^3_λ

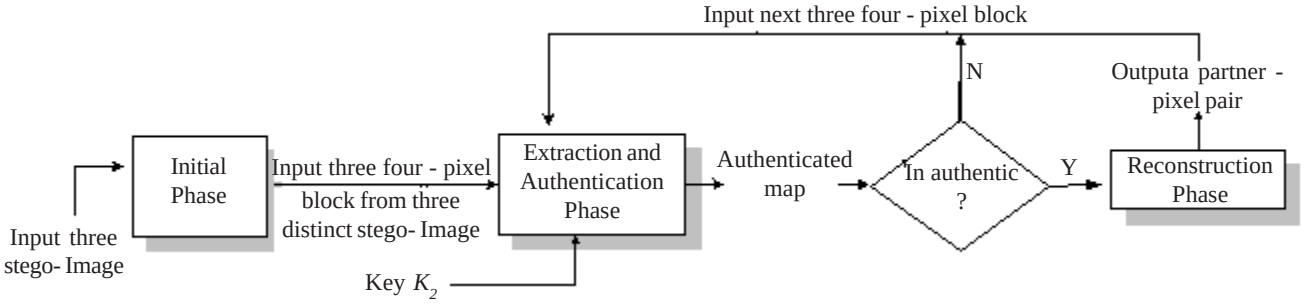


Figure 5. The flowchart of authentication and reconstruct procedure

and e_λ^4 will be used to recover the values of the λ^{th} partner-pixel pair.

$$e_\lambda^\alpha = \hat{p}_k^\alpha \bmod 5 \text{ for } 1 \le \alpha \le 4. \quad (7)$$

Later, the extracted data *embed_num* can be represented as a decimal value by using a 5-based notation system as $embed\_num = (e_\lambda^1, e_\lambda^2, e_\lambda^3, e_\lambda^4)_5$. The values of $R_\lambda^1(x_i)$, $R_\lambda^2(x_i)$ and $(\tilde{A}_\lambda(x_i)$ can be derived from *embed_num* through Equations (8.1), (8.2) and (8.3), respectively, where x_i indicates ID of the i^{th} participant.

$$R_\lambda^1(x_i) = embed\_num / (7 \times 3). \quad (8.1)$$

$$R_\lambda^2(x_i) = [embed\_num \bmod (7 \times 3)] / 8. \quad (8.2)$$

$$\tilde{A}_\lambda(x_i) = [embed\_num \bmod (7 \times 3)] \bmod 8. \quad (8.3)$$

During the authentication phase, $R_\lambda^1(x_i)$ and $R_\lambda^2(x_i)$ can be used to generate the authentication code $\hat{A}_\lambda$ with Equation (5) and to check whether values of $\tilde{A}_\lambda$ and $\hat{A}_\lambda$ are the same. If they are equivalent, the extracted data are marked as authentic and then go to the reconstruction phase; otherwise, the data are marked as inauthentic and skip the reconstruction phase. The next loop is performed until all pixels in the stego-images are processed. In line with the previous explanation, three pairs $\{R_\lambda^1(x_1), R_\lambda^2(x_1)\}$, $\{R_\lambda^1(x_2), R_\lambda^2(x_2)\}$ and $\{R_\lambda^1(x_3), R_\lambda^2(x_3)\}$ can be obtained from three four-pixel blocks of three distinct stego-images while IDs of participants are 1, 2 and 3, respectively. The results about whether authentic and inauthentic are recorded into an authenticated map.

As for the reconstruction phase, the three pairs $\{R_\lambda^1(x_1), R_\lambda^2(x_1)\}$, $\{R_\lambda^1(x_2), R_\lambda^2(x_2)\}$ and $\{R_\lambda^1(x_3), R_\lambda^2(x_3)\}$ are able to perform the Lagrange's interpolation together to reconstruct two pairs of three parameters a_λ^μ, b_λ^μ and c_λ^μ, shown as Equation (9), where x_1, x_2 and x_3 are the IDs of the participants and $\mu \in \{1, 2\}$.

As for the reconstruction phase, the three pairs $\{R_\lambda^1(x_1), R_\lambda^2(x_1)\}$, $\{R_\lambda^1(x_2), R_\lambda^2(x_2)\}$ and $\{R_\lambda^1(x_3), R_\lambda^2(x_3)\}$ are able to perform the Lagrange's interpolation together to reconstruct two pairs of three parameters a_λ^μ, b_λ^μ and c_λ^μ, shown as Equation (9), where x_1, x_2 and x_3 are the IDs of the participants and $\mu \in \{1, 2\}$.

$$a_\lambda^\mu = \frac{1}{(x_1 - x_2)(x_1 - x_2)} \times R_\lambda^\mu(x_1) + \frac{1}{(x_2 - x_1)(x_2 - x_3)} \times R_\lambda^\mu(x_2) + \frac{1}{(x_3 - x_1)(x_3 - x_2)} \times R_\lambda^\mu(x_3) \bmod 7$$

$$b_\lambda^\mu = \frac{(x_2 + x_3)}{(x_1 - x_2)(x_1 - x_2)} \times R_\lambda^\mu(x_1) + \frac{(x_1 + x_3)}{(x_2 - x_1)(x_2 - x_3)} \times R_\lambda^\mu(x_2) + \frac{(x_1 + x_2)}{(x_3 - x_1)(x_3 - x_2)} \times R_\lambda^\mu(x_3) \bmod 7$$

$$c_\lambda^\mu = \frac{x_2\, x_3}{(x_1 - x_2)(x_1 - x_2)} \times R_\lambda^\mu(x_1) + \frac{x_1\, x_3}{(x_2 - x_1)(x_2 - x_3)} \times R_\lambda^\mu(x_2) + \frac{x_1\, x_2}{(x_3 - x_1)(x_3 - x_2)} \times R_\lambda^\mu(x_3) \bmod 7 \quad (9)$$

Then a 7-based notational system is used to invert $(a_\lambda^1, b_\lambda^1, c_\lambda^1)_7$ and $(a_\lambda^2, b_\lambda^2, c_\lambda^2)_7$ to be two decimal values to present the values of the pixel and its mapping pixel, respectively. Finally, the processor outputs the authentic map and puts those two reconstructed pixels into two distinct reconstructed images; the reconstructed secret image and the shuffled secret image.

3.3 Recovery Procedures

The basic idea of the recovery mechanism is that the destroyed pixel located in a reconstructed image can be

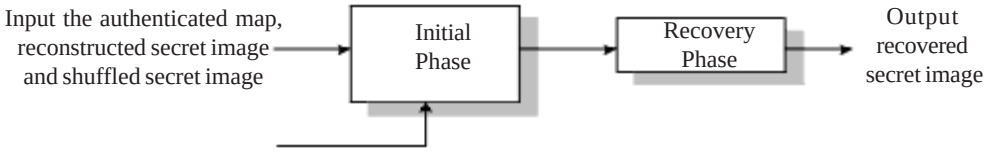
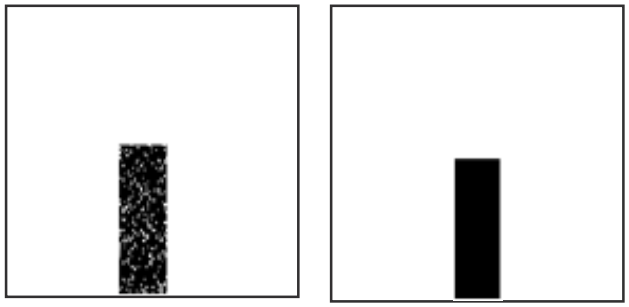


Figure 6. The flowchart of recovery procedure

recovered with the corresponding pixel located in a shuffled secret image. The flowchart of the recovery procedure is shown in Figure 6 and consists of two phases: the initial phase and the recovery phase.

After the authenticated map, reconstructed secret image and shuffled secret image, which are generated by authentication and reconstruction procedure, are input in the initial phase, the shuffled image is reversed to the original address by using the secret key K_1 as a re-shuffled image. To ensure all the modified parts are marked, an adjustment mechanism is proposed in initial phase for re-defining the authenticated map. As shown in Figure 7(a), authentic and inauthentic areas are colored as white and black pixels, respectively. Some of the inauthentic pixels are not pointed out in Figure 7(a), so the adjustment mechanism sets a smallest area in which covers all the inauthentic pixels, and then let all pixels in the area are treated as inauthentic, as shown in Figure 7(b). In the last phase, according to the authenticated map, the pixels located in inauthentic area are replaced with the pixels in the re-shuffled image at the same position. Finally, the processor outputs the recovered secret image.

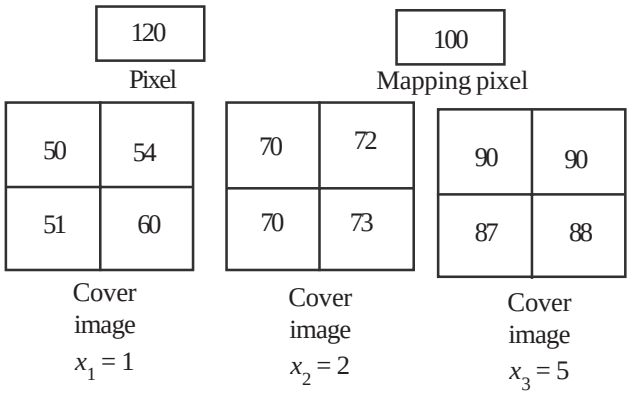


(a) Authentic and inauthentic area (b) Adjusted area

Figure 7. An example of adjustment mechanism

3.4 Example of Proposed Scheme

An example of the secret sharing and reconstructing procedures can help to illustrate the proposed method more clearly. We assume that the values of a secret pixel and its mapping pixel are 120 and 100, respectively, and that we would like to hide the secret into three four-pixel blocks from three distinct cover images for which the IDs of participants are 1, 5, 2, respectively, as shown in Figure 8(a). The pixel values 120 and 100 are translated into 7-based notations as $(231)_7$ and $(202)_7$, respectively. Therefore, the values of $a_\lambda^1, b_\lambda^1, c_\lambda^1, a_\lambda^2, b_\lambda^2$ and c_λ^2 are 2, 3, 1, 2, 0 and 2, respectively. The formulas are built through Equation (4) as $R_\lambda^1(x_i) = 2+3x_i+x_i^2 \bmod 7$ and $R_\lambda^2(x_i)=2+2x_i^2 \bmod 7$. Subsequently, we input the IDs of participants 1, 5 and 2 into above two formulas to retrieve the values of $R_\lambda^1(x_1), R_\lambda^1(x_2), R_\lambda^1(x_3), R_\lambda^2(x_1), R_\lambda^2(x_2)$ and $R_\lambda^2(x_3)$ as 6, 5, 0, 4, 3 and 3, respectively, as shown in Figure 8(b). Next, we input the IDs of participants and values of $R_\lambda^1(x_i)$ and $R_\lambda^2(x_i)$ to construct three authentication codes with Equation (5), as shown in Figure 8(b). After that, we assemble the values of $R_\lambda^1(x_i)$, $R_\lambda^2(x_i)$and $\tilde{A}_\lambda(x_i)$ to calculate the *Embed_num* with Equation (6), where x_i =1, 2 and 5, for i =1, 2 and 3, respectively. Finally, the values of *Embed_num* are transformed into 5-base notations for $e_\lambda^1, e_\lambda^2, e_\lambda^3$and e_λ^4. The hidden data $e_\lambda^1, e_\lambda^2, e_\lambda^3$ and e_λ^4 are embedded into the four-pixel blocks with an embedding algorithm as shown in Figure 8(c).



(a) Embedding pixels in the secret image and four-pixel block in the cover image

$120 = (231)_7$

$x_1=1, x_2=2, x_3=5$

$a_\lambda^1 = 2,\ b_\lambda^1 = 3,\ c_\lambda^1 = 1$

$R_\lambda^1(x_i) = 2+3x_i + x_i^2 \bmod 7$

$R_\lambda^1(x_1) = 6, R_\lambda^1(x_2) = 5, R_\lambda^1(x_3) = 0$

$100 = (202)_7$

$x_1=1, x_2=2, x_3=5$

$a_\lambda^2 = 2,\ b_\lambda^2 = 0,\ c_\lambda^2 = 2$

$R_\lambda^2(x_i) = 2+2x_i^2 \bmod 7$

$R_\lambda^2(x_1) = 4, R_\lambda^2(x_2) = 3, R_\lambda^2(x_3) = 3$

Authentication code for x_1=1 $\tilde{A}_\lambda(x_1) = 001 = 1$

Authentication code for x_2=2 $\tilde{A}_\lambda(x_2) = 111 = 7$

Authentication code for x_3=5 $\tilde{A}_\lambda(x_3) = 101 = 5$

(b) $R_\lambda^1(x_i)$, $R_\lambda^2(x_i)$ and $\tilde{A}_\lambda(x_i)$ generation

In the secret reconstructing phase, three four-pixel blocks from three distinct stego-images can extract two secret pixels: the original pixel and its mapping pixel. Each of the stego-images can extract four hidden data, such as $e_\lambda^1, e_\lambda^2, e_\lambda^3$ and e_λ^4, and then transform them into a decimal integer. Next, the decimal integer can be translated into multiple-base notations with Equations (8.1), (8.2) and (8.3) to retrieve the values of $R_\lambda^1(x_i)$, $R_\lambda^2(x_i)$and $\tilde{A}_\lambda(x_i)$, respectively. The resulting values of $R_\lambda^1(x_i)$ and $R_\lambda^2(x_i)$ are used to compute the authentication code as $\tilde{A}_\lambda(x_i)$ with Equation (5). Then, we compare the value of $\hat{A}_\lambda(x_i)$ with that of $\tilde{A}_\lambda(x_i)$: the pixel is authentic while they are equal and inauthentic otherwise. If the pixel is authentic, by applying the Equation (9), we input $R_\lambda^1(1)$, $R_\lambda^1(2)$ and $R_\lambda^1(5)$ to compute the values of a_λ^1, $b_\lambda^1$and c_λ^1, and also input $R_\lambda^2(1)$, $R_\lambda^2(2)$ and $R_\lambda^2(5)$ to compute the values of a_λ^2, b_λ^2 and c_λ^2. Finally, the values of $a_\lambda^1, b_\lambda^1, c_\lambda^1, a_\lambda^2, b_\lambda^2$ and $c_\lambda^2$are transformed

$x_1 = 1$
Embed_num = $(6, 4, 1)_{7,7,8} = 365$
$(e_\lambda^1, e_\lambda^2, e_\lambda^3, e_\lambda^4) = (2430)_5$

52	54
53	60

Strego-image

$x_2 = 2$
Embed_num = $(5, 3, 7)_{7,7,8} = 308$
$(e_\lambda^1, e_\lambda^2, e_\lambda^3, e_\lambda^4) = (2213)_5$

72	72
71	73

Strego-image

$x_3 = 5$
Embed_num = $(0, 3, 5)_{7,7,8} = 26$
$(e_\lambda^1, e_\lambda^2, e_\lambda^3, e_\lambda^4) = (0101)_5$

90	91
85	86

Strego-image

(c) The embedding phase

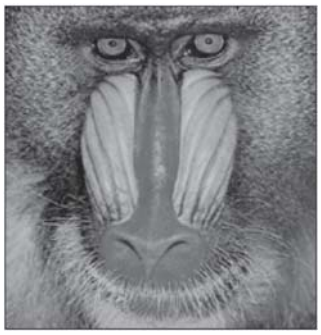
Figure 8. An example of secret sharing

into two decimal integers to reconstruct the values of the original pixel and its mapping pixel. The mapping pixel can re-shuffle to the original location with the secret key K_1. While the pixel is judged as inauthentic, it can be replaced with the mapping pixel in the re-shuffled image at the same position. After that, the inauthentic pixels are recovered as the original pixels, so the secret image is restored completely.

4. Experiments

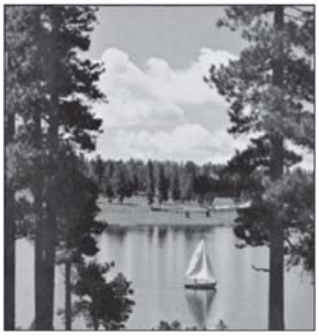
This section discusses the high performance and feasibility of our proposed scheme here. The secret image and cover images are 256×256 pixels and 512×512 pixels: "Butterfly" (the secret image) and cover images "Baboon", "F16", "Sailboat", "Lena" and "Pepper" were picked up from the USC-SIPI image database, as shown in Figure 9(a) and Figures 9(b)-(f), respectively.

(a) Butterfly(Secret Image)



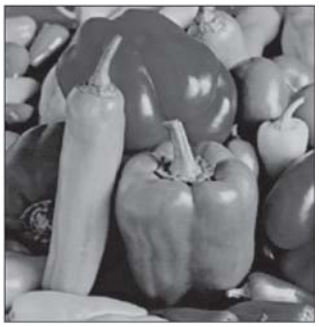
(b) Baboon (Cover Image)

(c) F16 (Cover Image)



(d) Sailboat (Cover Image)

(e) Lena (Cover Image)



(f) Pepper (Cover Image)

Figure 9. A secret image and five test images

The experiments are broken into three parts: visual quality of stego-images, the comparison of performances between the proposed scheme and the other schemes [5, 8, 16], and recovery ability, which demonstrates the feasibility of the proposed authentication and recovery mechanisms when undergoing attack. The visual quality is measured by using PSNR (peak-signal-to noise ratio), defined as Equation (10). Generally speaking, higher PSNR will bring lower distortions, i.e., higher performance.

$$PSNR = 10 \text{ x } \log_{10} \frac{255^2}{MSE} \text{dB} \qquad (10)$$

Here, *MSE* is abbreviated from mean-square error, which

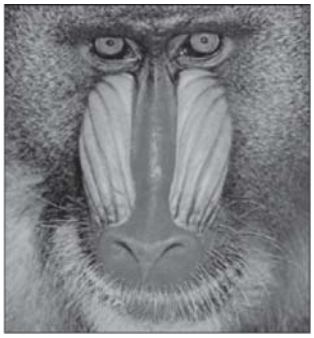
calculates the distortions between the cover image and the stego-image, defined as Equation (11).

$$MSE = \frac{1}{h \times w} \sum_{i=1}^{k} \sum_{j=1}^{w} (p_{ij} - p_{ij}^{t})^2 \qquad (11)$$

Here, h and w are the height and weight of the cover image, and p_{ij} and p_{ij}^{t} are the pixel values of the cover image and the stego-image, respectively.

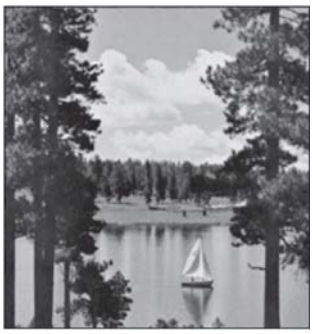
In the first experiments (Figure 10), the visual qualities of the stego-images are, on average, higher than 45.11 dB, and it is difficult to distinguish the differences between the cover images and the stego-images with unassisted eyesight.

To evaluate the performance, we compare the visual quality (i.e., PSNR) of the proposed scheme with schemes proposed in [5, 8, 16] (see Table 1). The visual quality of



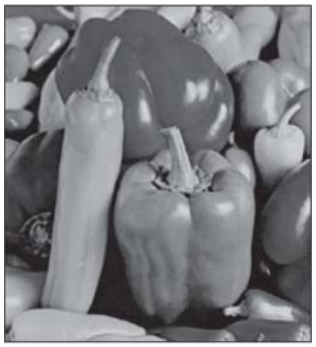
(a) Baboon (Stego-image)
PSNR: 45.11 dB

(b) F16 (Stego-image)
PSNR: 45.13



(c) Sailboat (Stego-image)
PSNR: 45.11

(d) Lena (Stego-image)
PSNR: 45.11



(e) Pepper (Stego-image)
PSNR: 45.11

Figure 10. Five stego-images

Images	Baboon	F16	Sailboat	Lena	Pepper	Complete Recovery
Scheme [2]	45.10	45.11	45.10	45.12	45.12	No
Scheme [5]	40.93	40.99	40.98	40.97	40.96	No
Scheme [8]	37.71	38.35	38.49	38.60	38.29	No
Scheme [16]	40.06	40.15	40.97	41.10	40.66	No
Proposed scheme	45.11	45.13	45.11	45.11	45.11	Yes

Table 1. Comparison of performances between our proposed scheme and schemes in [2, 5, 8, 16]

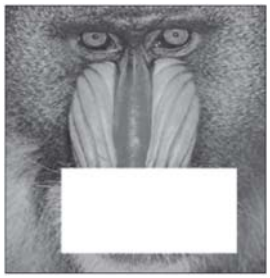
our proposed scheme is much higher than the others, about 6 dB greater than scheme [8] and 3 dB greater than schemes [5, 16]. Additionally, none of the other schemes [5, 8, 16] include any mechanism for secret image recovery. In comparison with schemes [2], the visual qualities have a little bit improvements, but we have the ability of complete recovery to remedy the corrupted areas.

To measure our authentication and recovery abilities, six examples are conducted according to different case discussions, as illustrated in Figure 11. The modified images are shown in Figures 11(a1)-11(a6), and after the authentication and recovery mechanisms, the authentic results, adjusted results and recovery results are shown in Figures 11(b1)-11(b6), 11(c1)-11(c6) and 11(d1)-11(d6),

(a1) Some lost information

(a2) Copy modification

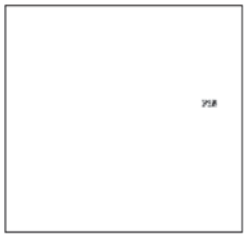
(a3) A word “F16” addition



(a4) Cropping modification

(b1) Authenticated area

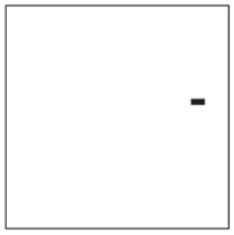
(b2) Authenticated area



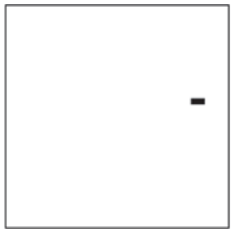
(b3) Authenticated area

(b4) Authenticated area

(c1) Adjusted area



(c2) Adjusted area



(c3) Adjusted area

(c4) Adjusted area

(d1) Secret recovery

(d2) Secret recovery

(d3) Secret recovery

(d4) Secret recovery

(a5) A fake image

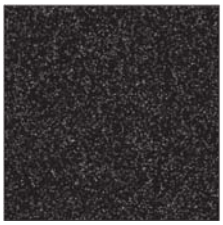
(a6) Provided a false key K_2

(b5) Authenticated area

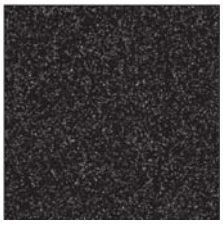
(b6) Authenticated area

(c5) Adjusted area

(c6) Adjusted area



(d5) Secret recovery



(d6) Secret recovery

Figure 11. Examples for secret image recovery

respectively. In those figures, the inauthentic and authentic bits are marked with black color and white color, respectively.

In the first case shown in Figure 11, we assume some information has been lost during transmission (Figure 11(a1)), and the lost information is drawn with black color as an inauthentic area (Figure 11(b1)). After performing the adjustment mechanism, the adjusted area is illustrated (Figure 11(c1)), and its corresponding broken parts are restored (Figure 11(d1)).

As shown in Example 2, the boat in the stego-image is copied and pasted into the same stego-image (Figure 11(a2)), and the inauthentic area is found (Figure 11(b2)). Finally, the adjusted area is pointed out (Figure 11(c2)), and the secret image can be completely reversed (Figure 11(d2)).

To test the fine-grained performance evaluated in the proposed scheme, a small word "F16" is added into the stego-image shown in Figure 11(a3); its corresponding authentic area, adjusted area and recovery secret image are shown in Figure 11(b3), 11(c3) and 11(d3), respectively. From the results of Figure 11(c3), the location of the "F16" can be marked precisely, and the secret image can be completely reconstructed so it can confirm that the authentication area can achieve a fine-grained tamper detection and recovery.

When the stego-image is cropped in a big area, as shown in Figure 11(a4), the area can be marked as authentic area (Figure 11(b4)) and adjusted area (Figure 11(c4)). Also, the secret image can be completely extracted by using the proposed recovery mechanism as shown in Figure 11(d4).

To confirm that the proposed scheme can resist a false image attack, as demonstrated in Example 5 (Figure 11(a5)), the attacker provides a fake image for the authentic area (Figure 11(b5)). By using the adjustment mechanism, the authentic area will result in adjusted area with all-blocked area (Figure 11(c5)), which means that no area is authentic and the secret image is never obtained (Figure 11(d5)).

The final example demonstrates what happens if an attacker uses the wrong key to participate in the secret sharing process. The authenticated results are shown in Figure 11(b6), in which all areas in adjusted area are denoted as inauthentic as shown in Figure 11(c6). As shown in Figure 11(d6), the secret image can not be extracted at all.

5. Conclusions

In this paper, we proposed a novel secret sharing scheme for tamper detection and recovery. With the steganography technique, the secret is protected in a way that the shares concealed in stego-images are difficult for censors using human vision to detect. With the proposed system, the tampered regions are detected and can be completely recovered while the participants are legal. In the experimental results, the proposed scheme has great resilience and ability to detect the inauthentic regions as well as to recover the inauthentic regions.

References

[1] Chen, Y. F., Chan, Y. K., Huang, C. C., Tsai, M. H., Chu, Y. P. (2007). A multiple-level visual secret-sharing scheme without image size expansion. *Information Sciences* 177 (21) 4696-4710.

[2] Chang, C. C., Chen, Y. H., Wang, H. C. (2011). Meaningful secret sharing technique with authentication and remedy abilities. *Information Sciences* 181 (14) 3073–3084.

[3] Chang, C. C., Lin, C. C., Lin, C. H., Chen, Y. H. (2008). A novel secret image sharing scheme in color images using small shadow images. *Information Sciences* 178 (11) 2433-2447.

[4] Chang, C. C., Lin, C. Y., Tseng, C. S. (2007). Secret image hiding and sharing based on the (t, n)-threshold. *Fundamenta Informaticae* 76 (4) 399-411.

[5] Chang, C. C., Hsieh, Y. P., Lin, C. H. (2008). Sharing secrets in stego images with authentication. *Pattern Recognition* 141(10) 3130-3137.

[6] Feng, J. B., Wu, H. C., Tsai, C. S., Chu, Y. P. (2005). A new multi-secret images sharing scheme using Lagrange's interpolation. *Journal of Systems and Software* 76 (3) 327-339.

[7] Fang, W. P., Friendly progressive visual secret sharing (2008). *Pattern Recognition* 41 (4) 1410-1414.

[8] Lin, C. C., Tsai, W. H. (2004). Secret image sharing with steganography and authentication. *Journal of Systems and Software* 73 (3) 405-414.

[9] Lou, D. C., Tso, H. K., Liu, J. L. (2007). A copyright protection scheme for digital images using visual cryptography technique. *Computer Standards and Interfaces* 29 (1) 125-131.

[10] Lukac, R., Plataniotis, K. N. (2005). Bit-level based secret sharing for image encryption. *Pattern Recognition* 38 (5) 767-772.

[11] Naor, M., Shamir, A. (1995). Visual cryptography, *Advances in Cryptology -EuroCrypt'94*, In: LNCS 950 Springer, Berlin, pages 1-2. 1995.

[12] Shamir A. (1979). How to share a secret. *Communications of the Association for Computing Machinery*, p. 612-613.

[13] Shyu, S. J., Huang, S. Y., Lee, Y. K., Wang R. Z., Chen, K. (2007). Sharing multiple secrets in visual cryptography. *Pattern Recognition* 40 (12) 3633-3651.

[14] Thien, C. C., Lin, J. C. (2002). Secret image sharing, *Computers and Graphics* 26 (1) 765-770.

[15] Tsai, C. S., Chang, C. C., Chen, T. S. (2002). Sharing multiple secrets in digital images. *Journal of Systems and Software* 64 (2) 163-170.

[16] Yang, C. N., Chen, T. S., Yu, K. H., Wang, C. C. (2007). Improvements of image sharing with steganography and authentication. *Journal of Systems and Software* 80 (7) 1070-1076.