Journal of Information Hiding and Multimedia Signal Processing ©2014 ISSN 2073-4212
Ubiquitous International Volume **5**, Number **3**, July **2014**

Meaningful Shadows for Image Secret Sharing with Steganography and Authentication Techniques

Chin-Chen Chang[1,4], Yi-Hui Chen[2,*], and Lin-Yi Chuang[3]

[1]Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan.
E-mail: ccc@cs.ccu.edu.tw

[2]Department of Applied Informatics and Multimedia,
Asia University, Taichung 41354, Taiwan
*Correspondence Autor
E-mail: chenyh@asia.edu.tw

[3]Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi 621, Taiwan, R.O.C.
E-mail: cly95m@cs.ccu.edu.tw

[4]Department of Computer Science and Information Engineering
Asia University, Taichung 41354, Taiwan
Received February, 2013; revised September, 2013

Abstract. *In this paper, we proposed a lossless secret sharing scheme using a steganography technique to improve the performances of Lin and Tsai's as well as Yang et al.'s schemes. To ensure the reconstructed image is the true secret image, an authentication mechanism is imported into the proposed scheme to verify whether all the shadows are validated before reconstructing the secret image. In comparison with Lin and Tsai's and Yang et al.'s schemes, our proposed scheme delivers much more effective performances.*
Keywords: Secret sharing, meaningful shadows, steganography, authentication segmentation.

1. **Introduction.** Secret sharing (SS) [2, 3, 6, 9, 11, 13-15] is also called a (t, n)-threshold secret sharing scheme used in the protection of secrets. This scheme breaks secrets down into n shadows while any t shadows are subsequently used to reconstruct the original secrets, where $t \leq n$. The basic criterion is that the true secrets can never been obtained when any of the shadows is a fake shadow made up by an illegal participant. Up to now, a significant amount of research has proposed visual secret sharing (VSS) schemes [1, 5, 8, 10, 12], which encrypt secrets into n meaningless shadows as random noises. In the reconstruction phase of the VSS scheme, secrets can be presented by stacking t shadows without any computations with the human visual system (HVS). However, the VSS scheme cannot be completely restored to the original secrets. Using a polynomial approach, some researches [2]-[4] are proposed to completely reconstruct the secrets; however, the generated shadows look like random noises that cause censors to doubt the shadows concealing any secrets. To prevent the noise-like shadows from being perceptible to censors, some researches [7,16] combine the steganography technique with the SS scheme to embed the shared data into the meaningful cover images without being suspected by secret stealers

as it is difficult to distinguish between the original and stego-images using the naked eyes, which the embedded images are also called the stego-images.

Historically, to avoid the participants from providing fake stego-images prior to reconstructing the secret image, an authentication mechanism is imported into an SS scheme to verify the fidelities of the stego-images. Therefore, applying a polynomial, Lin and Tsai [7] provided a secret sharing scheme using steganography and authentication features to embed one shared data into a four-pixel square block of a cover image. Although Lin and Tsai's scheme is based on a good idea in regards to combining secret sharing, steganography, and authentication techniques, Yang et al. [16] argued that Lin and Tsai's scheme has three weaknesseswnamely, hard to detect dishonest participants, large distortions in the stego-images, and non-lossless secret image reconstruction, which will be described in Section 2.

To improve upon these weaknesses, Yang et al. proposed an improved version to provide better visual quality of the stego-image as well as higher security ability. Generally speaking, a smaller shadow size promotes faster transmission on the network. However, Yang et al.'s scheme is inefficient in that a four-pixel square block can only carry one shared data, where one shared data is generated from one pixel in secret image; in other words, the stego-image is extended to four times the size of the secret image. Measuring such performance relies on pixel expansion (PE), which refers to how many times the secret image is expanded in order to present a secret image, where PE is computed as PE = (the size of shadow image/the size of secret image). In essence, the smaller PE values are, the better performance will be.

In this paper, we proposed a simple SS scheme with an authentication mechanism to enhance Yang et al.'s scheme, which provides better visual quality of stego-image and smaller shadows size than that of Yang et al.'s. In the proposed scheme, two pixels in the cover image are treated as a pixel pair; each pixel pair can be used to embed one shared data. To increase the visual quality of the stego-image, an adjustment rule is applied, which defines a better way for embedding shared data into the cover image at a lower cost than the traditional LSB (least significant bit) replacement method.

The rest of this paper is organized as follows. In Section 2, we shall briefly review Yang et al.'s scheme. In Section 3, we shall present our secret sharing and reconstructing procedures. To illustrate the main points of the proposed scheme, an example is provided in Section 3. The experimental results and several evaluations will be presented in Section 4. Finally, conclusions and future works will be drawn in the last section.

2. **Literature Review.** Yang et al. [16] proposed a better scheme to address the three weaknesses found in Lin and Tsai's scheme [7]. The first weakness is called hard to detect dishonest participants, which means fake stego-images can easily pass their proposed authentication mechanism as the check bits are generated by using even or odd parity. In other words, the fake images can be perceived as authentic while illegal participants follow the parity checking rule to make up the stego-images. Instead of parity checking, Yang et al. applied the hash function with the secret key K_r and concatenated the block identification to generate check bits for addressing the problem. The second weakness pointed to the visual quality of the stego-image in Lin and Tsai's scheme, which was not of sufficient quality. According to this weakness, Yang et al. proposed an arrangement ability to enhance the visual quality. Indeed, the visual quality was significantly enhanced in their experiments. In the final weakness, Lin and Tsai's scheme requires the use of more pixels in stego-images than Yang et al.'s to present a secret pixel, while the secret pixel value ranges from 0 to 255.

Although Yang et al.'s scheme performs much better than Lin and Tsai's scheme, however, in both schemes, a four-pixel square block in the cover image can only be used to embed one shared data so that the size of the stego-image is four times that of the secret image. Furthermore, the distortions of the stego-image in their schemes are not minimized as they adopt traditional LSB replacement to directly replace the LSBs (least significant bits) of a pixel with the shared data. In this paper, the proposed scheme can reduce the shadow size and enhance the visual quality of the stego-image. More details of the analysis in which the proposed scheme is compared to Yang et al.'s scheme are provided in Section 4.

3. **The Proposed Lossless Image Secret Sharing Scheme.** This section presents a novel SS scheme with authentication ability thereby making the visual quality of the stego-image as high as possible. Moreover, the authentication ability can authenticate whether the stego-images provided by participants are legal before reconstructing the secret image. Consequently, any two stego-images can completely recover the original secret image. This section details two procedureswnamely, the secret sharing and reconstructing procedures.

Secret Sharing Procedure

A secret image S is with size $w \times w$, in which the pixel values are depicted as s_i, where s_i ranges from 0 to 255 and $1 \leq i \leq w^2$. In the embedding procedure, the secret image is first permuted with a secret key K, and all pixels in the cover image are divided into several pixel pairs P_j, where$1 \leq j \leq w^2$. The first pixel and its counterpart in the pixel pair P_j are represented as P_j^1 and P_j^2, respectively. The binary representation of the pixel s_i in the secret image is denoted as $(b_1 b_2 \ldots b_8)$, which is subsequently divided into two nibbles comprising the first four MSBs (most significant bit) $(b_1 b_2 b_3 b_4)$, and the last four LSBs $(b_5 b_6 b_7 b_8)$. For example, the value of pixel s_i is 142, and its corresponding binary representation is $(10001110)_2$. The first four MSBs (1000) and the last four LSBs (1110) can be treated as two nibbles.

Subsequently, the two nibbles are transformed into two integers 8 and 14, depicted as s_i^1 and s_i^2, respectively. Next, a formula is generated using the two integers with Equation (1), in which x is the identification of the participant and the computed result is returned to F.

$$F = x \times s_i^1 + s_i^2 \bmod 17, \tag{1}$$

Following the above example, the formula is generated as $F = x \times 8 + 14 \bmod 17$. After inputting the identification of the participant into the formula, the computational results and a check bit, which together are called a shared data, are embedded into a pixel pair in the cover image. The generation of the check bit, denoted as A, is broken into two caseswnamely, Case 1 and Case 2waccording to different F values. By computing Equation (2), if the value of R is equal to 0, it falls into Case 1; otherwise, falls into Case 2. In Equation (3), the values of ℓ, y_1 and y_2 are given according to the chosen case. Then, F value is transformed into an ℓ-bit binary bit stream as $f = (f_1 f_2 \ldots f_\ell)$.

$$R = F \bmod 16. \tag{2}$$

$$\begin{cases} \ell = 5,\ y_1 = 3 \ and \ y_2 = 3,\ if\ Case\ 1, \\ \ell = 4,\ y_1 = 2 \ and \ y_2 = 3,\ Otherwise. \end{cases} \tag{3}$$

Next, the check bit is produced from Equation (4), which concatenates the bit stream f, f', and x and finally runs the XOR operation to return the value to A, where f' is a random

bit stream with the same size as f generated by applying the pseudo random number generator with the secret key K. In addition, x is the identification of the participant.

$$A = \mathrm{XOR}(f, f', x). \tag{4}$$

In the embedding procedure, the check bit A initially concatenates with the bit stream f to become the shared data. Next, the first y_1 bits and the last y_2 bits of the shared data are converted into two integers as SD_1 and SD_2. Additionally, the last y_1 LSBs in P_j^1 and the last y_2 LSBs in P_j^2 are transformed into two integers, PL_1 and PL_2, respectively. SD_1 and SD_2 are subsequently embedded them into the last y_1 LSBs in P_j^1 and the last y_2 LSBs in P_j^2. An adjustment embedding method is proposed as defined in Equation (5), where j is the identification of the pixel pair, d is the difference between SD_g and PL_g (i.e., $d = SD_g - PL_g$), and $g \in \{1, 2\}$ indicates the pixel number in the pixel pair P_j.

$$\begin{array}{cc} P_j^g = P_j^g + d - 2^{y_g}, & if\ d > 2^{y_g-1} and\ (P_j^g + d - 2^{y_g}) \geq 0, \\ P_j^g = P_j^g - |d| + 2^{y_g}, & if\ d < -2^{y_g-1} and\ (P_j^g - |d| + 2^{y_g}) \leq 255, \\ P_j^g = P_j^g + d, & otherwise. \end{array} \tag{5}$$

Except for the "otherwise" case in Equation (5), the adjustment embedding method has the ability to limit the distortion range from 0 to 2^{y_g-1}. In comparison the distortion of traditional LSB replacement with that of the proposed method, an example is taken as follows. When a secret bit stream $SD_1 = (101)$ is hidden into the pixel $P_1^j = 8(8 = (00001000)_2)$, using the traditional approach, the last three LSBs of P_1^j are directly replaced with SD_1 as $(00001101)_2 = 13$. Following the above example by applying the proposed approach, the values g, y_1, PL_1, SD_1, and d are 1, 3, 0, 5, and 5, respectively. As the calculated $P_1^j + d - 2^{y_1}$ is greater than 0, the embedded pixel P_1^j is computed as 5. The distortions between the original pixel P_1^j and the embedded one, adopting the traditional method and the proposed method, are 5 and 3, respectively. Consequently, the distortion of the proposed scheme can be significantly lower than that of the traditional scheme.

To describe the embedding procedure more clearly, the data embedding process of the proposed scheme is described step-by-step. All ten steps are repeated until all secret pixels are concealed into the cover image.

Step 1: Input pixel s_i in secret image S.
Step 2: Input a pixel pair P_j in a cover image.
Step 3: Transform the s_i into two nibbles as integers s_i^1 and s_i^2.
Step 4: Generate a formula using Equation (1).
Step 5: Input the identification of the participant to return the F value.
Step 6: Compute the R value using Equation (2). If the R value is 0, Case 1 is chosen; otherwise, choose Case 2.
Step 7: Determine the values of ℓ, y_1, and y_2 according to Equation (3) and then transform the F value into an ℓ-bit bit stream f. For example, if the value ℓ is 5, the $f = (f_1 f_2 f_3 f_4 f_5)$.
Step 8: Get a check bit A using Equation (4).
Step 9: Concatenate A with f and subsequently calculate the values of SD_1, SD_2, PL_1, and PL_2.
Step 10: Embed SD_1 and SD_2 into pixels P_j^1 and P_j^2 using Equation (5).

Secret Reconstruction Procedure

During the reconstruction procedure, at least two stego-images can losslessly reconstruct the original secret image. First, two pixels are treated as a pixel pair C_j, where $1 \leq j \leq w^2$

and w^2 is the size of the secret image. Pixels C_j^1 and C_j^2 refer to the first and the other pixel in the pixel pair C_j . In the reconstructing procedure, two cases are divided according to the values of C_j^1 and C_j^2. If the last LSB of C_j^1 and last three LSBs of C_j^2 are all zeros, it falls into Case 1; otherwise, it falls into Case 2. In Case 1, the decoder retrieves the third last LSB of C_j^1 to be the check bit as A; otherwise, it obtains the check bit from the second last LSB of C_j^1. Using Equation (6), the values of z_1 and z_2 are retrieved according to the chosen case. Next, the last z_1 LSBs in C_j^1 concatenate with the last z_2 LSBs in C_j^2 to be the concatenated bit stream f'', which is then transformed into an integer I_j. Subsequently, check bit A' is retrieved by calculating Equation (7), where f' is generated with the key K and x

is the identification of the participant. In comparing A with A', if both are the same, the pixel pair is authentic; otherwise, it is inauthentic. To show the authenticated image in experiments, the authentic pixel pair outputs pixels in black; otherwise, it will output pixels in white.

$$\begin{cases} z_1 = 2 \; and \; z_2 = 3, \; if \; Case \; 1, \\ z_1 = 1 \; and \; z_2 = 3, \; Otherwise. \end{cases} \tag{6}$$

$$A' = \mathrm{XOR}(f''||f'||x). \tag{7}$$

When the stego-image is evaluated to be authentic, the formula for the current pixel pair is built as $ax + b \bmod 17 = I_j$ defined in Equation (8), where a and b are unknown values and x is the identification of participant. Two authentic pixel pairs from any two valid stego-images can build two formulas for each secret pixel construction. Two formulas can subsequently cooperate to resolve the values a and b by using Lagrange's interpolation. Finally, the pixel s_i in the secret image can be reconstructed using Equation (9). When all pixels are constructed, the decoder uses the key K to permute the extracted secret pixel to be the original secret image.

$$ax + b \bmod 17 = I_j. \tag{8}$$

$$s_i = a \times 16 + b \tag{9}$$

The brief process for the reconstruction procedure is described as follows. Each of the participants attending to the secret image reconstruction must repeat the following ten steps until all secret pixels are reconstructed.

Step 1: Input two pixels C_j^1 and C_j^2 from the pixel pair C_j.

Step 2: If the last LSB in C_j^1 and last three LSBs in C_j^2 are all zeros, it falls into Case 1 and goes to Step 2.a; otherwise, it falls into Case 2 and goes to Step 2.b.

Step 2.a: Obtain the check bit from the third last LSB in C_j^1 as A.

Step 2.b: Retrieve the check bit from the second last LSB in C_j^1 as A.

Step 3: Retrieve the values of z_1 and z_2 applying Equation (6) according to the chosen case from Step 2.

Step 4: Concatenate the last LSB in C_j^1 with the last z_2 LSBs in C_j^2 as the bit stream f'' and transform it into an integer I_j.

Step 5: Generate the bit stream f' generated using the secret key K and compute the check bit A'S with Equation (7).

Step 6: Compare A with A'; if A is equal to A', the pixel pair is authentic and output pixel values 0; otherwise, it is inauthentic and output pixel values 255.

Step 7: Generate a formula using Equation (8) as $ax + b \bmod 17 = I_j$.

Step 8: Resolve the two unknown values a and b by cooperating with another legal participant by applying Lagrange's interpolation.
Step 9: Calculate the reconstructed pixel s_i using Equation (9).
Step 10: If the reconstructed pixel is the last pixel in the reconstructed image, use the key K to permute the reconstructed image into an original secret image; otherwise, output the constructed pixel s_i and then go to Step 1.

An Example for Sharing and Reconstructing Procedures

An example is presented here to demonstrate the sharing and reconstruction procedures more clearly. The shared data generated from pixel s_i is embedded into two pixel pairs, namely P_1 and P_3 which are from two different cover images. The details of the embedding process are shown in Figures 1(a)-1(c). First, pixel si consists of two nibbles that are transformed into two integers, 9 and 8. As shown in Figure 1(a), a formula is generated as $F = 9x + 8 \bmod 17$ by using Equation (1), where the values of s_i^1 and s_i^2 are 9 and 8, respectively, and x is the identification of the participant.

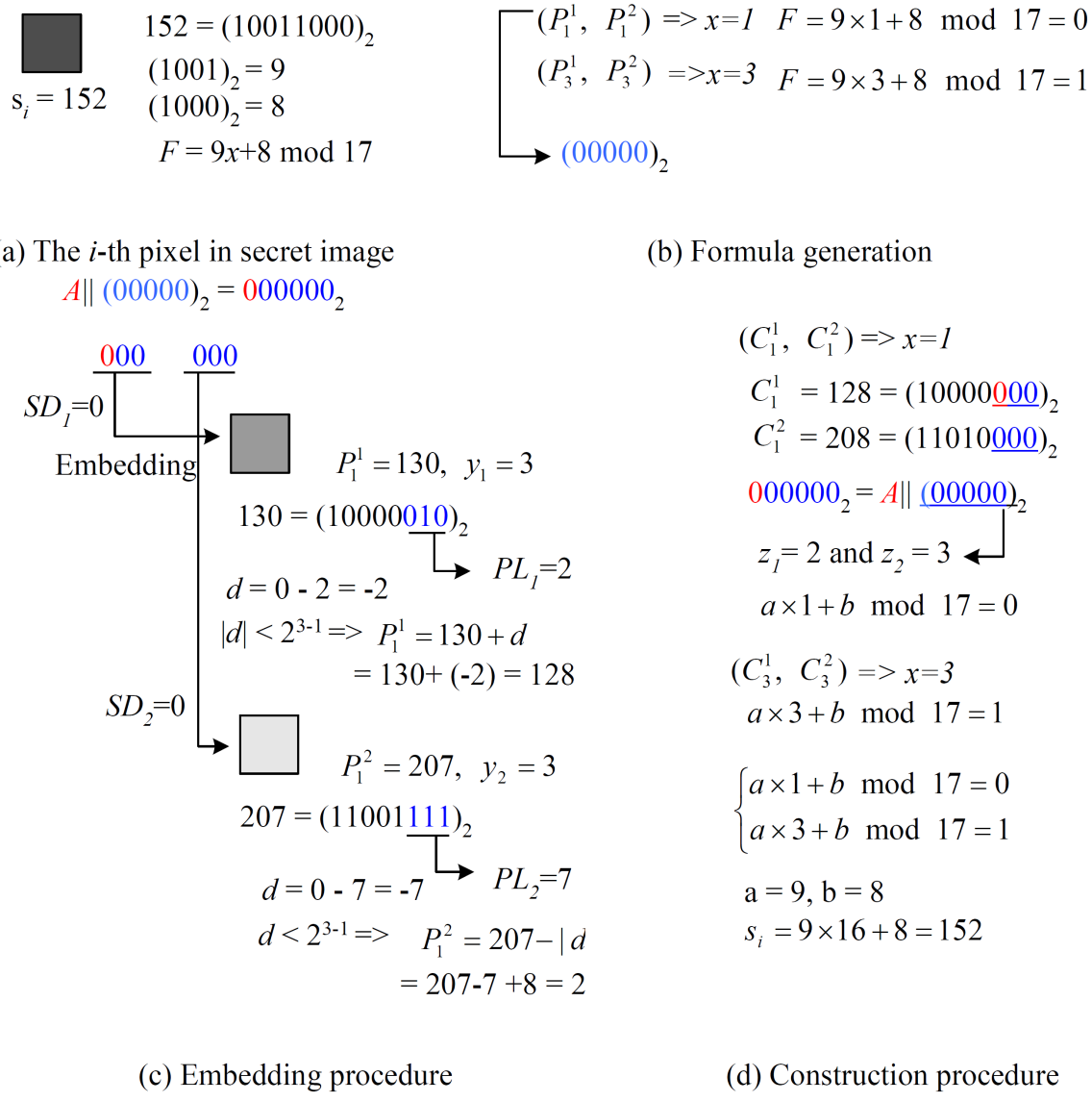

FIGURE 1. Example of embedding and reconstruction procedures

In this example, the identification of the participant for the pixel pairs (P_1^1, P_1^2) and (P_3^1, P_3^2) are 1 and 3, respectively. According to the results, the F values are 0 and 1, as shown in Figure 1(b). Due to R value calculated with Equation (2) is equal to 0; thus, the encoder transforms the F value into a 5-bit bit stream as $(00000)_2$. As the embedding procedures for both pixel pairs (P_1^1, P_1^2) and (P_3^1, P_3^2) are the same, we only take one pixel pair (P_1^1, P_1^2) as an example; its corresponding pixel values are 130 and 207, respectively, as shown in Figure 1(c). The check bit A is then concatenated with the bit stream. First, three bits are embedded into the first pixel P_1^2; the other bits are embedded into P_1^1. The last three LSBs of P_1^1 and P_1^2 are extracted and denoted as PL_1 and PL_2 to be integers as 2 and 7. Owing to the F value is 0, the SD1 and SD_2 are all zeros. Consequently, the differences between SD_1 and PL_1, and SD_2 and PL_2 are -2 and -7, respectively. According to the embedding rule with Equation (5), the embedded pixels are 128 and 208.

In the reconstructing procedure, receivers convert the stego-pixels into bit streams at first as shown in Figure 1(d). As the last three LSBs of pixel C_1^2 and the last LSB of pixel C_1^1 are all zeros, the decoder performs the extraction process using Case 1. In Case 1, the third last LSB of pixel C_1^1 is the check bit as A and the decoder concatenates the five bits retrieved from the last z_1 LSBs in pixels C_1^1 and the last z_2 LSBs in pixel C_1^2, where z_1 and z_2 are 2 and 3, respectively, by applying Equation (6). Next, the concatenated bit stream is converted into an integer I_1. Using Equation (7), a check bit is generated as A' so the decoder can verify whether the pixel pair is valid by comparing A with A'. The identification of the participant and the value I_1 are inputted into Equation (8) to build the formula $a \times 1 + b \bmod 17 = 0$. In the same way, pixels C_3^1 and C_3^2 are able to build another formula ae3+b mod 17 = 1. The two built formulas can be seen as the simultaneous equations $\begin{cases} a \times 1 + b \bmod 17 = 0 \\ a \times 3 + b \bmod 17 = 1 \end{cases}$. Next, the values of a and b can be resolved by using Lagrange's interpolation. Finally, the pixel s_i can be recovered as $s_i = a \times 16 + b = 9 \times 16 + 8 = 152$.

4. **Experiments and Security Analysis.** In our experiments, five test cover images with size 512 × 512w"Baboon", "Jet", "Sailboat", "Lena" and "Pepper"wwere selected from the USC-SIPI image database, as shown in Figures 2(c)-6(g). In this section, three critical metricswnamely, visual quality, pixel expansion, and security and authentication issuesware considered in measuring the performances of schemes [7], [16], and ours.

(a) Secret image

(b) Secret image

(c) Baboon

(d) F16

(e) Sailboat (f) Lena (g) Pepper

FIGURE 2. Secret images and five cover images

To compare the performances of schemes [7] and [16] with ours, we use the same sized cover images but a different sized secret image for schemes [7] and [16], and ours, as shown in Figures 2(a) and 6(b), respectively. A four-pixel block in schemes [7] and [16] can be used to embed one shared data. Moreover, a pixel pair can embed one shared data in our scheme. Therefore, using the same sized cover image, our hiding capacity is twice of that of scheme [7] and [16]. Consequently, the PE value in our scheme is only half theirs because a pixel pair is used to embed one shared data instead of a four-pixel square block.

After embedding the secret image (Figures 2(b) and (a)) behind the same cover image using our proposed method and schemes [7] and [16], respectively, the performances of the proposed scheme and the others were compared (see Table 1). In regards to visual quality (PSNR), the PSNR value of the proposed scheme is higher than those of schemes [7] and [16] even though our capacity is twice theirs because the proposed embedding policy can provide a better embedding method for minimizing more distortions in the proposed scheme than in schemes [7] and [16].

TABLE 1. Comparison of performances between proposed scheme and schemes [7] and [16]

Performances / Images	Scheme [7]		Scheme [16]		Proposed Scheme	
	PE	PSNR (dB)	PE	PSNR (dB)	PE	PSNR (dB)
Baboon	4	37.71	4	40.06	2	42.36
Jet	4	38.35	4	40.15	2	42.42
Lake	4	38.49	4	40.97	2	42.39
Lena	4	38.60	4	41.10	2	42.41
Pepper	4	38.29	4	40.66	2	42.40
Average	4	38.288	4	40.588	2	42.396

The outcomes after authenticating are divided into three partswnamely, stego-images, tampered images, and authenticated images. In the first part (see Figures 3(a1)-7(d1)), it is difficult to distinguish the differences between the stego-images and the original ones using the naked eyes. To highlight the performance of authentication ability, different tampered images are presented in Figures 3(a2)-7(d2). During authentication, the inauthentic pixels are detected and drawn with white; conversely, the authentic pixel is presented in block, as shown in Figures 3(a3)-7(d3).

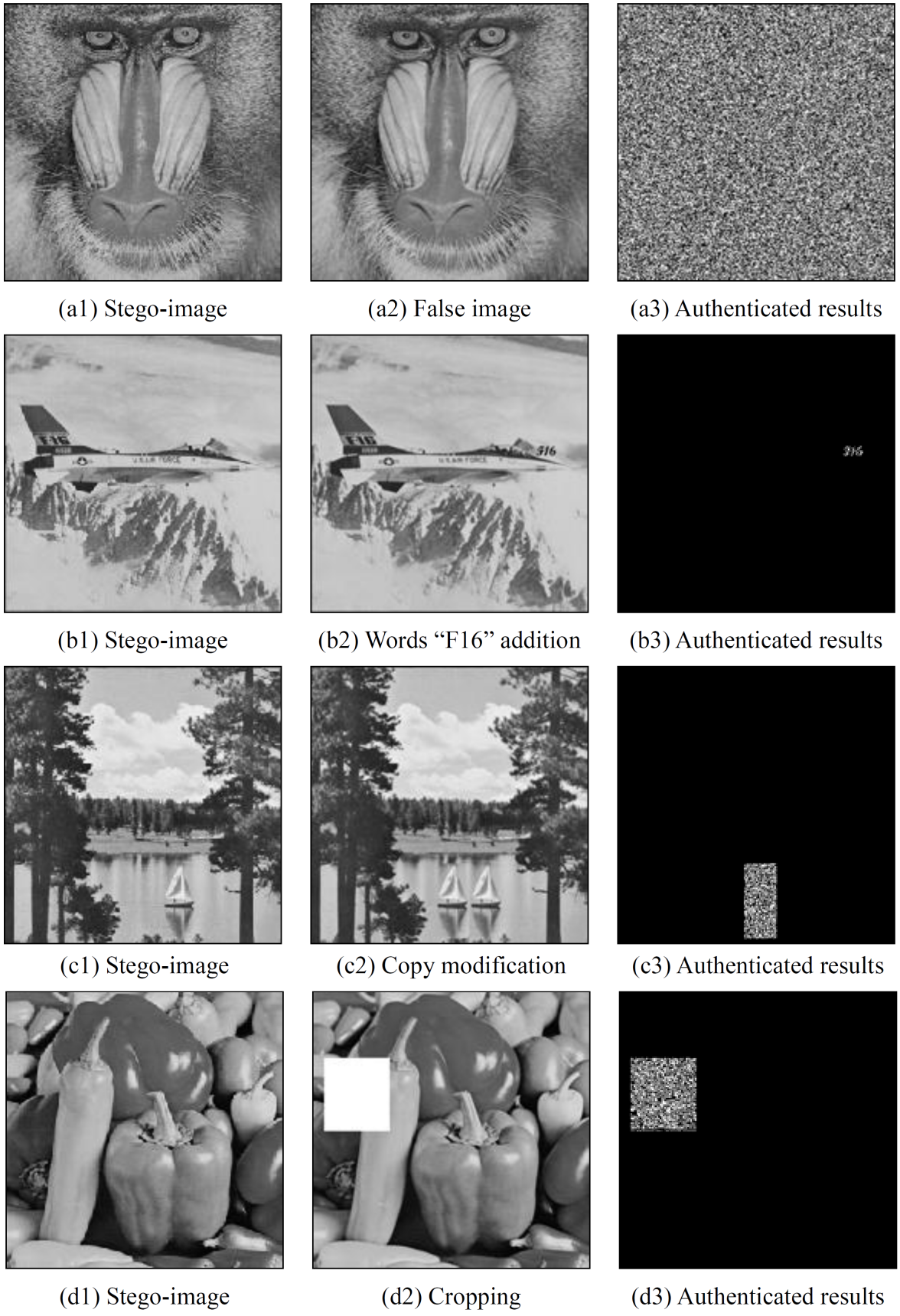

FIGURE 3. Four authenticated examples

In the first authenticated experiment, a fake image similar to the stego-images is generated as shown in Figure 3(a2). After authentication, the authenticated image as random noises (shown in Figure 3(a3)) mean the image is inauthentic. In comparing the proposed scheme with schemes [7] and [16], two pixels (a pixel pair) in the proposed scheme use one check bit to determine whether the pixels are authentic, providing a fine-grained ability to detect a smaller tampered area. To demonstrate this fine-grained ability, the word F16 has been added to Figure 3(b1), as shown in Figure 3(b2). After authentication, the faked area is localized as shown in Figure 3(b3).

In the third experiment, the copied boat is taken from blocks of the stego-image itself, as shown in Figure 3(c2). The location of the fake boat is also localized using the proposed authentication ability, as demonstrated in Figure 3(c3). Finally, a square block is cropped from Figure 3(d1), as shown in Figure 3(d2). The cropped area is localized using our authentication ability as illustrated in Figure 3(d3). These four experiments demonstrate that the authentication ability of proposed scheme can successfully note which area is authentic and which is inauthentic.

In the security analysis of our proposed scheme, the secret image is permuted by the secret key while the check bit generation depends on the secret key and the identification of participant. As such, attackers cannot reconstruct the secret image and generate the correct check bit without the secret key. The attackers can correctly guess each check bit with a 1/2 probability, meaning they only have a $(1/2)^{\text{size}/4\times\text{size}}$ probability of successfully reconstructing the original secret image, where size is the height or width size of the stego-image. In comparison with schemes [7] and [16], the four-pixel square block only gets one check bit so that the attackers have a $(1/2)^{\text{size}/4\times\text{size}}$ probability of uccessfully passing the authentication mechanism. Thus, the probability of attackers successfully reconstructing the image in the proposed scheme is much smaller than that in schemes [7] and [16].

5. **Conclusions.** This paper has presented a novel secret sharing scheme that can effectively embed a secret image into a cover image and subsequently completely reconstructed. Comparing the proposed scheme to other schemes, the proposed scheme achieves a better image quality, smaller pixel expansion, and higher security mechanism than that of Lin and Tsai's and Yang et al.'s schemes. Additionally, the computation complexity of the proposed scheme is low and does not require complex computation. Furthermore, the fake or tampered image has a low probability $(1/2)^{\text{size}\times\text{size}}$ of correctly guessing the authentication, making it difficult to pass the authentication mechanism without the secret key. On average, the image quality after embedding is above 42 dB, meaning it is imperceptible to attackers using the naked eyes. Although the proposed scheme outperforms Lin and Tsai's and Yang et al.'s schemes, the shadows size is still a significant issue. In the future, we will investigate another SS scheme that has smaller shadow sizes and higher visual quality.

REFERENCES

[1] Y. F. Chen, Y. K. Chan, C. C. Huang, M. H. Tsai, and Y. P. Chu, A multiple-level visual secret-sharing scheme without image size expansion *Information Sciences*, vol. 177, no. 21, pp. 4696-4710, 2007.

[2] C. C. Chang, C. C. Lin, C. H. Lin, and Y. H. Chen, A novel secret image sharing scheme in color images using small shadow images, *Information Sciences*, vol. 178, no. 11, pp. 2433-2447, 2008.

[3] C. C. Chang, C. Y. Lin, and C. S. Tseng, Secret image hiding and sharing based on the (t, n)-threshold, *Fundamenta Informaticae*, vol. 76, no. 4, pp. 399-411, 2007.

[4] J. B. Feng, H. C. Wu, C. S. Tsai, and Y. P. Chu, A new multi-secret images sharing scheme using Lagrange's interpolation, *Journal of Systems and Software*, vol. 76, no. 3, pp. 237-339, 2005.

[5] W. P. Fang, Friendly progressive visual secret sharing, *Pattern Recognition*, vol. 41, no. 4, pp. 1410-1414, 2008.
[6] Y. C. Hou, and Z. Y. Quan, Progressive visual cryptography with unexpanded shares, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 21, no. 11, pp. 1760-1764, 2011.
[7] C. C. Lin, and W. H. Tsai, Secret image sharing with steganography and authentication, *Journal of Systems and Software*, vol. 79, no. 3, pp. 405-414, 2004.
[8] D. C. Lou, H. K. Tso, and J. L. Liu, A copyright protection scheme for digital images using visual cryptography technique, *Computer Standards & Interfaces*, vol. 29, no. 1, pp. 125-131, 2007.
[9] R. Lukac, and K. N. Plataniotis, Bit-level based secret sharing for image encryption, *Pattern Recognition*, vol. 38, no. 5, pp. 767-772, 2005.
[10] M. Naor, and A. Shamir, Visual cryptography, *Proc. of Advances in Cryptology-EUROCRYPT'94, Workshop on the Theory and Application of Cryptographic Techniques*, LNCS 950, Springer, pp. 1-12, 1995.
[11] A. Shamir, How to share a secret, *Communications of the Association for Computing Machinery*, vol. 22, no. 11, pp. 612-613, 1979.
[12] S. J. Shyu, S. Y. Huang, Y. K. Lee, R. Z. Wang, and K. Chen, Sharing multiple secrets in visual cryptography *Pattern Recognition*, vol.40, no. 12, pp. 3633-3651, 2007.
[13] C. C. Thien, and J. C. Lin, Secret image sharing, *Computers and Graphics*, vol. 26, no. 1, pp. 765-770, 2002.
[14] C. S. Tsai, C. C. Chang, and T. S. Chen, Sharing multiple secrets in digital images, *Journal of Systems and Software*, vol. 64, no. 2, pp. 162-170, 2002.
[15] X. Wu, and W. Sun, Random grid-based visual secret sharing with abilities of OR and XOR decryptions, *Journal of Visual Communication and Image Representation*, vol. 24, no. 1, pp. 48-62, 2013.
[16] C. N. Yang, T. S. Chen, K. H. Yu, and C. C. Wang, Improvements of image sharing with steganography and authentication, *Journal of Systems and Software*, vol. 80, no. 7, pp. 1070-1076, 2007.