Sharing a Secret Image in the Cloud Using Two Shadows

Yu Chen[1], Jiang-Yi Lin[2,3], Chin-Chen Chang[3] and Yu-Chen Hu[4]
(Corresponding author: Chin-Chen Chang)

School of Information Science and Engineering, Fujian University of Technology[1]
33 Xuefu South Road, Fuzhou 350118, China
Department of Computer Science, Xiamen University of Technology[2]
600 Ligong Road, Xiamen 361024, China
Department of Information Engineering and Computer Science, Feng Chia University[3]
100 Wenhua Road, Taichung 40724, Taiwan
Department of Computer Science and Information Management, Providence University[4]
200, Section 7, Taiwan Boulevard, Shalu District, Taichung 43301, Taiwan
(Email: alan3c@gmail.com)
(Received Mar. 20, 2018; Revised and Accepted Sept. 7, 2018; First Online Feb. 20, 2020)

Abstract

In this paper, we present a novel (2, 2) reversible secret image sharing scheme. Our scheme permits secret messages to be shared with two participants by splitting the marked encrypted image into two shadows. The secret messages can be reconstructed if two participants collaborate with each other. The proposed scheme chooses suitable binary blocks of a cover image in which to embed the secret message and divides those blocks into two shadow blocks by executing a logical operation with all of the other binary blocks, thereby producing two shadows. In the data extraction procedure, the secret messages and the cover image can be reconstructed by the logical operation of the corresponding binary blocks of the two shadows. A practical application is demonstrated by modeling our scheme as a reversible watermarking scheme in the Cloud. The experimental results indicated that the proposed method is reversible and that it can restore the image and watermark properly.

Keywords: Data Hiding; Reversible Watermarking; Secret Image Sharing

1 Introduction

It is very important to secure information [1,12,14,15,18] in today's information age. Secret sharing is an effective approach to protect the security of information by sharing parts of the data with different holders to avoid leaking useful information. In 1979, Shamir and Blakley independently introduced the concept of secret sharing and proposed two (t, m) threshold schemes [1, 15]. In Shamir's scheme [15], a dealer divides a secret, D, into m pieces, which are kept by a group of m users; Then, t or more users can collaborate to recover D, and less than t users cannot restore D. Inspired by these (t, m) threshold schemes [3, 5, 10, 13, 16], many researchers focused on the study of secret sharing. In 1995, Naor and Shamir extended the (t, m) threshold scheme to secret image sharing and proposed the concept of visual secret sharing (VSS) [13]. Although their scheme was a novel method for sharing secrets, it applies only to binary images and incurs the pixel expansion problem.

In the past decades, as an important direction for secret message delivery applications, some schemes [2,4,6–9,11] have been proposed. Chang *et al.* introduced a secret image sharing scheme [2] in 2008. In their scheme, a magic matrix was used to modify the cover image in order to embed the secret digits. Their scheme can completely restore a cover image after the secret digits are extracted. In 2009, Lee *et al.* proposed a reversible data hiding scheme [9] in which two steganographic images were used. According to their scheme [9], a cover pixel pair is changed at most by one when two secret bits are embedded. Therefore, a high steganographic image quality is provided. In 2015, Lu *et al.* proposed a dual-image-based data hiding method [11]. In their study, the center folding strategy was used to reduce the value of the secret symbols to obtain the folded secret information that was embedded in the two images. Recently, Chang and Liu presented a novel (2, 2) secret sharing method [7]. In their research, the balance between the quality and the payload of the shadows can be achieved easily by adjusting a control parameter. Compared with the methods of Lee *et al.* [9] and Lu *et al.* [11], Chang and Liu's scheme [7] achieved the highest payload and best flexibility.

Inspired by the above schemes, we present a novel (2, 2) reversible secret sharing scheme. The proposed scheme uses the logical operation on bit-planes [17] to embed secret data, such as the watermark, and then splits the data into two shares. Thus, our scheme generates two meaningful images, called shadows. Furthermore, we applied the proposed secret image sharing scheme to a reversible watermarking scheme in the Cloud to demonstrate its validity.

The rest of this paper is organized as follows. Section 2 reviews Chang and Liu's scheme. Section 3 presents the proposed reversible secret sharing scheme. Section 4 provides our experimental results, and Section 5 gives our conclusions.

2 Brief Introduction of Chang and Liu's Scheme

Chang and Liu's scheme [7] is a reversible (2, 2) secret image sharing method. Their scheme consists of two procedures, *i.e.*, the secret sharing procedure and the secret and image reconstructing procedure. Here, the secret image and the cover image are both grayscale images. The two procedures are described in the following subsections.

2.1 Secret Sharing Procedure

In this process, a grayscale secret image is embedded into a grayscale cover image P. Assume that P with size of $H \times W$ is expressed as $P = \{p_i | i = 1,\ 2,\ \ldots,\ (H \times W)\}$, where p_i is the i-th pixel of P and $p_i \in [0, 255]$. Let G be a binary stream representing a secret image, with the length of $|G|$. G is split into k segments, and each segment, c_q, is the size of ω bits whose value is in the range of 0 to $2^\omega - 1$. Here ω is considered as a control parameter. Thus, G can be represented as $G = \{c_q | q = 1,\ 2,\ \ldots,\ k\}$, where $k = \lceil |G| / \omega \rceil$. Two shadow images, P_1 and P_2, which are the same size as P, are generated after G is embedded. The detailed steps are as follows:

Step 1: Set a value of ω.

Step 2: Sequentially read each segment c_q of ω bits from G.

Step 3: Embed c_q into p_q to produce two shadow pixels p_{q1} and p_{q2}. Let $\omega^* = 2^\omega/2$, and then perform different embedding processes for $c_q \le \omega^*$ or $c_q > \omega^*$.

Step 3.1: Embed c_q for $c_q \le \omega^*$. There are three cases to deal with.

Case 1: If $0 \le p_q \pm c_q \le 255$, p_{q1} and p_{q2} are computed as

$$p_{q1} = p_q - c_q, \quad (1)$$
$$p_{q2} = p_q + c_q. \quad (2)$$

Case 2: If $p_q + c_q > 255$, p_{q1} and p_{q2} are computed as

$$p_{q2} = p_q, \quad (3)$$
$$p_{q1} = p_q - (2e - 1), \quad (4)$$

where $e = p_q + c_q - 255$.

Case 3: If $p_q - c_q < 0$, p_{q1} and p_{q2} are computed as

$$p_{q1} = p_q, \quad (5)$$
$$p_{q2} = p_q + (2f - 1), \quad (6)$$

where $f = 0 - (p_q - c_q)$.

Step 3.2: Embed c_q for $c_q > \omega^*$. Firstly, set $c_q = c_q - \omega^*$ to satisfy $c_q < \omega^*$. Secondly, perform Step 3.1 to obtain p_{q1} and p_{q2}. Finally, swap p_{q1} and p_{q2}.

Step 4: Repeat Step 2 and Step 3 until G is fully processed and P_1 and P_2 are generated.

Step 5: Give P_1 to one receiver and give P_2 to another receiver.

In the above steps, ($2e$-1) and ($2f$-1), which appear in Cases 2 and 3, respectively, are set to be odd in order to identify the overflow condition, which will be described in detail in the subsequent extraction process. Setting $c_q = c_q - \omega^*$ in Step 3.2 is to decrease distortion. Then, p_{q1} and p_{q2} are swapped to satisfy $p_{q1} \ge p_{q2}$.

Example 1. *Assume that a binary stream* S= *'00111101' is part of the secret image* G *and that* $p_1 = 80$ *and* $p_2 = 253$ *are two pixels of the cover image* P. *Next is how to embed* S *into* p_1 *and* p_2. *First, we set* $\omega = 4$ *and* $\omega^* = 2^\omega/2 = 8$. *So,* S *is separated into two segments* c_1 *and* c_2, *both of which are 4 bits,* i.e., $c_1 = (0011)_2 = 3$ *and* $c_2 = (1101)_2 = 13$.

1) Embed $c_1 = 3$ into $p_1 = 80$: Since $c_1 < \omega^*$ and $0 \le p_1 \pm c_1 \le 255$, it is Case 1. And we apply Equations (1) and (2) to produce two shadow pixels $p_{11} = p_1 - c_1 = 80 - 3 = 77$ and $p_{12} = p_1 + c_1 = 80 + 3 = 83$.

2) Embed $c_2 = 13$ into $p_2 = 253$: Since $c_2 > \omega^*$, the process turns to Step 3.2 to set $c_2 = c_2 - \omega^* = 13 - 8 = 5$. Then, the process turns to Step 3.1. Since $p_2 + c_2 = 253 + 5 > 255$, it is Case 2, and Equations (3) and (4) are applied to compute p_{21} and p_{22} where $p_{22} = p_2$ and $p_{21} = p_2 - (2e-1) = p_2 - (2 \cdot (p_2 + c_2 - 255) - 1) = 248$. Finally, p_{22} and p_{21} are swapped.

2.2 Secret and Image Reconstructing Procedure

Let p_{j1} and p_{j2} be the corresponding pixels of the two shadow images P_1 and P_2, respectively. The secret image and the cover image can be reconstructed by the following steps:

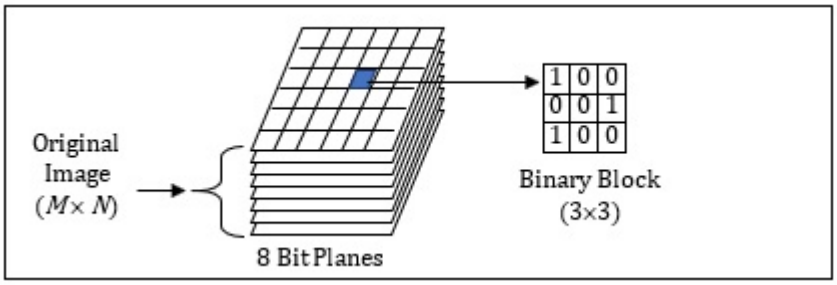

Figure 1: Schematics of bit-planes and a binary block

Step 1: Swap p_{j1} and p_{j2} if $p_{j1} > p_{j2}$.

Step 2: Reconstruct secret segment c_q and cover pixel p_q. The reconstruction process consists of the following three cases:

Case 1: If $(p_{q1} + p_{q2}) \bmod 2 = 0$,

$$p_q = \frac{p_{q1} + p_{q2}}{2}, \tag{7}$$

$$c_q = p_q - p_{q1} \text{ or } c_q = p_{q2} - p_q. \tag{8}$$

Case 2: If $(p_{q1}+p_{q2}) \bmod 2 \neq 0$ and $p_{q2}+\omega^* > 255$,

$$p_q = p_{q2}. \tag{9}$$

According to Equations (3) and (4), c_q can be calculated as

$$c_q = 255 + \frac{p_{q2} - p_{q1} + 1}{2} - p_{q2}. \tag{10}$$

Case 3: If $(p_{q1} + p_{q2}) \bmod 2 \neq 0$ and $p_{q1} - \omega^* < 0$,

$$p_q = p_{q1}.$$

According to Equations (5) and (6), c_q can be calculated as

$$c_q = \frac{p_q + p_{q2} + 1}{2}.$$

Step 3: Set $c_q = c_q + \omega^*$ if Step 1 is executed.

Step 4: Repeat Steps 1-3 until all the pixels of the shadow images are processed.

In this scheme, the parity of a number is used to distinguish whether or not it is overflow. The order of the values of two shadow pixels is employed to determine whether a normal value or a processed value was embedded.

Example 2. *Use the two pairs of shadow pixels generated in Example 1,* $(p_{11} = 77, p_{12} = 83)$ *and* $(p_{21} = 253, p_{22} = 248)$*, to demonstrate the reconstructing procedure for the secret and the cover images.*

1) Use p_{11} and p_{12} to restore c_1 and p_1. Notice that $p_{11} < p_{12}$, we can get $c_1 \leq \omega^*$ based on the method used in the secret sharing procedure. Since $(p_{11} + p_{12}) \bmod 2 = 0$, Equations (7) and (8) are applied to reconstruct p_1 as $p_1 = (p_{11} + p_{12})/2 = (77 + 83)/2 = 80$ and c_1 as $c_1 = p_1 - p_{11} = 80 - 77 = 3 = (0011)_2$.

2) Use p_{21} and p_{22} to restore c_2 and p_2. Here $p_{21} > p_{22}$ means that $c_2 > \omega^*$, and the process turns to Step 1. Also, because $(p_{21} + p_{22}) \bmod 2 \neq 0$ and $p_{22} + \omega^* > 255$, that is Case 2, and Equations (9) and (10) are used to calculate p_2 and c_2. Here $p_2 = p_{22} = 253$ and $c_2 = 255 + \frac{p_{22} - p_{21} + 1}{2} - p_{22} = 255 + \frac{253 - 248 + 1}{2} - 253 = 5$. Finally, adjust c_2 to $c_2 + \omega^* = 5 + 8 = 13 = (1101)_2$. Accordingly, we obtain the binary secret stream $S =$ '00111101' and the cover image pixels are $p_1 = 80$ and $p_2 = 253$.

3 Proposed Secret Image Sharing Scheme

Inspired by Chang and Liu's scheme [7], we propose a novel (2, 2) secret image sharing scheme using a model in which an image is watermarked and shared by the Cloud. The bit-planes-based technique is used for image processing in the proposed scheme, and it is introduced as follows. An original grayscale image of 8-bit resolution can be decomposed into eight bit-planes in such a way that we can perform image processing at the bit-level. Yi *et al.* [17] embedded the bits in the lower bit-planes of an original image into the higher bit-planes, allowing the lower bit-planes to be reserved for hiding secret data later.

The logical operation is performed on the binary blocks in the bit-planes and the binary blocks with certain characteristics are selected to embed the watermark. Then, the embedded bit-planes are combined to form a stego-image. A schematic of the bit-planes of an image and a binary block is shown in Figure 1.

After receiving an image, the Cloud embeds a watermark in it, produces two shadows, S_1 and S_2, of the marked image, and then distributes them to the two recipients. By incorporating S_1 and S_2, we can extract the watermark and recover the original image. The procedures in our proposed scheme are shown schematically in Figure 2.

3.1 Watermark Embedding and Shadows Generating

To prevent revealing any information of an original image I, the proposed method encrypts I by processing the

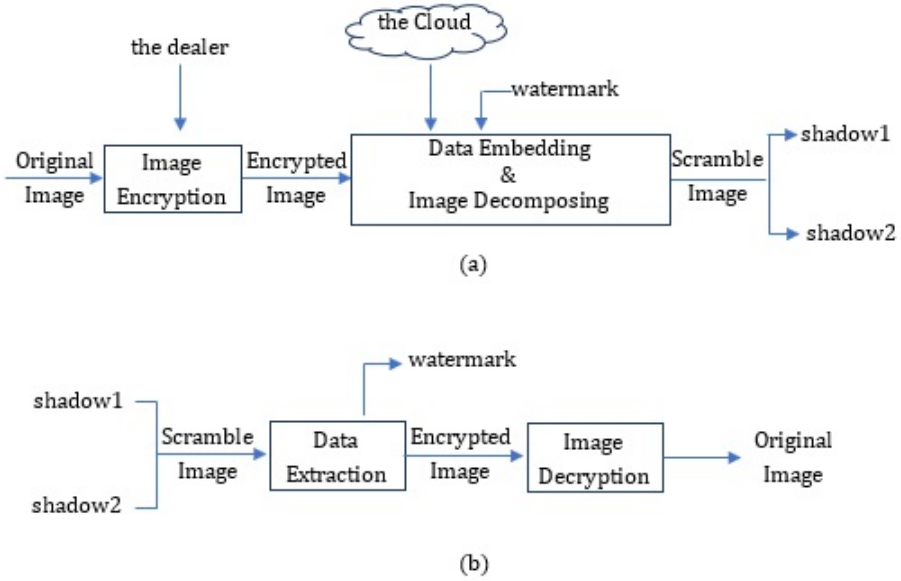

Figure 2: Procedures of (a) Secure image sharing; (b) Secret data retrieving and image restoring

encryption procedure, and only the encrypted image will be sent to the Cloud. Let E be an encrypted image. $E = \{r_l | l = 1, 2, \ldots, (M \times N)\}$, where $r_l \in [0, 255]$. Now we decompose E into eight bit-planes, and every bit-plane then is divided into a set of non-overlapping 3×3 binary-blocks. For each block, let b be the central bit of the block. Let n and n' be the total numbers of the remaining bits in this block excluding b that are equal to b or not, respectively. Based on the result of the comparison of n and n', all the blocks are classified into two categories, *i.e.*, the Nice block where $n > n'$ and the Bad block where $n \leq n'$.

The Nice blocks are selected and the central bits of them will be modified to embed the watermark. Since the Cloud will produce two shadows of E, *i.e.*, S_1 and S_2, for the two receivers, the proposed scheme utilizes different combinations of the values of the central bits of the blocks in the corresponding S_1 and S_2 to denote that 0 or 1 has been embedded. Next, we give an example of the Nice block and the Bad block and the structure of the shadow block in Figure 3.

The Nice blocks will be chosen to embed the watermark. In the proposed scheme, we generate two corresponding shadow blocks for one block of the cover image. The central point bits of a block and its corresponding shadow blocks are designed and shown in Table 1. The column L_1 indicates which number in a block is more. The column L_2 represents the combination of b_1 and b_2, the central bits of the two corresponding shadow blocks. According to the columns L_1 and L_2, the proposed scheme gives four scenarios, respectively, against $b = 0$ or $b = 1$.

Let B be a 3×3 Nice block divided from the bit-planes of the cover image, whose structure is shown in Figure 4(a). Let SB_1 and SB_2 be two corresponding shadow blocks to B, and distribution of their bits is shown in Figures 3 (c) and (d), respectively. The two shadow

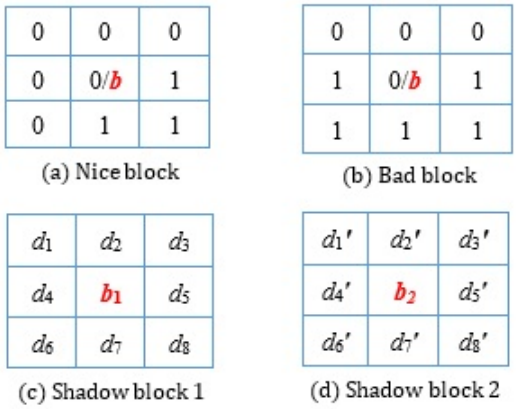

Figure 3: An example of the blocks in encrypted image and shadow image where the value of the central point, b, is 0; (a) The Nice block for $n = 5$ and $n' = 3$; (b) The Bad block with $n = 3$ and $n' = 5$; (c) and (d) The bits structure of shadow blocks corresponding to (a).

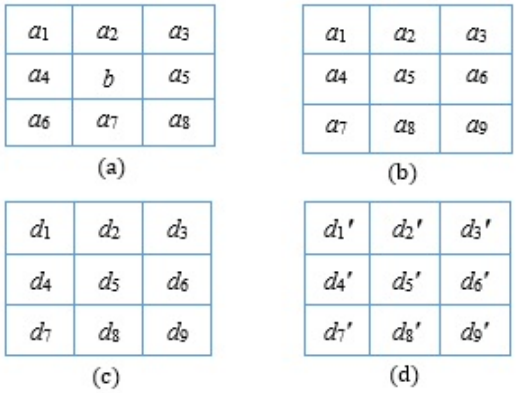

Figure 4: Examples of the structure of the blocks

Table 1: Central point bits of the binary block and its corresponding shadow blocks

$b=0$			$b=1$		
Majority bit L_1	b_1b_2 L_2	Description L_3	Majority bit L_1	b_1b_2 L_2	Description L_3
0	00	embedding 0	0	11	bad block
0	01	embedding 1	1	01	no embedding
0	10	no embedding	1	10	embedding 0
1	00	bad block	1	11	embedding 1

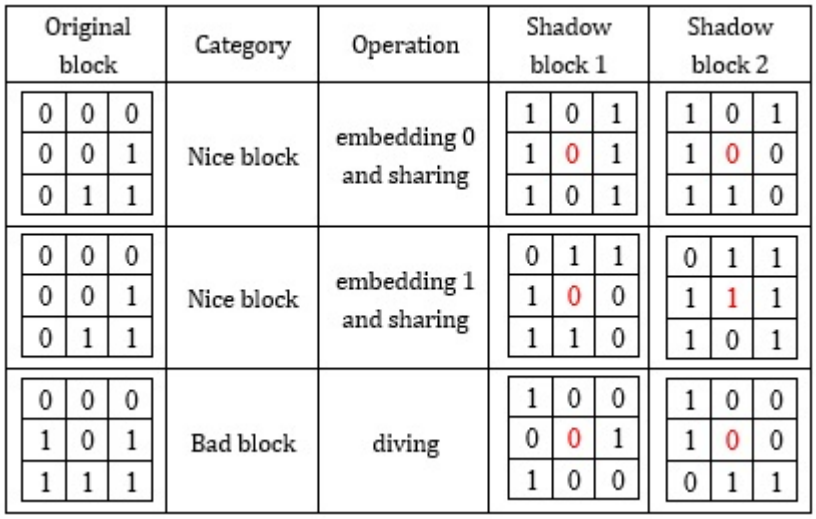

Figure 5: Examples of an original block being divided into two shadow blocks

blocks SB_1 and SB_2 must satisfy $d_i \oplus d'_i = a_i$, where $i = 1, 2, \ldots, 8$, $\oplus$ represents the XOR operation, and (b_1, b_2) must satisfy the conditions in Table 1 for embedding zero and one. Similarly, for the Bad block B is equal to Figures 4 (b). Let the two shadow blocks SB_1 and SB_2 be equal to Figures 4 (c) and (d), respectively. And SB_1 and SB_2 must satisfy $d_i \oplus d'_i = a_i$, where $i = 1, 2, \ldots, 9$.

As we can see, d_i can be replaced by 0 or 1, so can d'_i. Thus, d_i and d'_i are unfixed for a shadow block. After all the binary-blocks are processed as shown above, the Cloud produces two shadows of the encrypted image and delivers them to the two receivers. However, they only can obtain the original block of the encrypted image by the cooperation of the two shadow blocks, since neither of the single shadow blocks can give the original block. Therefore, the proposed scheme allows the secure sharing of secret images.

Example 3. *The following example illustrates how to generate two shadow blocks for embedding a single bit in a given Nice block with the situation that the central bit equal zero. Figure 5 lists three cases of the process. Similarly, for the case of the central bit being one, we can also list like this according to the conditions in Table 1.*

3.2 Watermark Extraction and Image Recovering

After embedding the watermark in the encrypted image, the Cloud delivers two shadows, S_1 and S_2 of the encrypted image to two different receivers, R_1 and R_2. After being permitted, R_1 and R_2 can extract the watermark and restore the image to the state that the Cloud accepted, and they can restore the image to original image by decryption. The process is as follows. First, two shadows will be decomposed into 8 bit-planes, and, then, each bit-plane will be divided into a set of non-overlapping 3×3 binary blocks. Second, we perform the XOR operation on a pair of blocks from corresponding bit-plane pairs from the two shadows. Finally, according to the state of the two central point values and the operating result blocks, we can obtain the original block without any embedded secret bits.

Example 4. *Use two shadow blocks to perform the restoring and extracting process. We can use the two shadow blocks generated in Example 3. Three examples of obtaining the original blocks and secret bits are shown in Figure 6.*

First, the XOR operation is performed on two shadow blocks to get a result block, where their central bits are labeled as b_1 and b_2. Second, determine the number, d,

Shadow block 1	Shadow block 2	XORing result	Majority bit	Original block	Secret bit
1 0 1 1 0 1 1 0 1	1 0 1 1 0 0 1 1 0	0 0 0 0 0 1 0 1 1	0	0 0 0 0 0 1 0 1 1	0
0 1 1 1 0 0 1 1 0	0 1 1 1 1 1 1 0 1	0 0 0 0 1 1 0 1 1	0	0 0 0 0 0 1 0 1 1	1
1 0 0 0 0 1 1 0 0	1 0 0 1 0 0 0 1 1	0 0 0 1 0 1 1 1 1	1	0 0 0 1 0 1 1 1 1	null

Figure 6: Examples of the restoration of the original block and the extraction of secret bits

that appears more often except for the central point. Finally, according to columns L_1 and L_2 in Table 1, we can get the original block and the secret bit by comparing the relationships between L_1 and d and between L_2 and the connection string of b_1 and b_2. Therefore, in the first row of Figure 6, the central bit of the original block and the secret bit are both 0 when $d = 0$ and the concatenation of b_1 and b_2 is '00'. By proceeding with all of the blocks in the two shadows in this way, the encrypted image that was sent to the Cloud is restored, and the watermark is extracted.

4 Experimental Results

In this section, some experiments were conducted on some test images to evaluate the correctness of the proposed scheme. Our experiments were conducted by using software MATLAB R2012b running on a personal computer whose operation system is Windows 10. The CPU of the computer is Intel Xeon E3-1225 v5, 3.3GHz, and the memory is 8GB. Focusing on the method related to secret sharing rather than encrypting images, the Henon map, an extensively-used, easily-implemented method, was used as the simulation to encrypt our test image. The Henon map is a two-dimensional, non-linear map, and it is defined as follows:

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{cases} \quad (11)$$

In our experiments, we set $x_1 = -0.4$, $y_1 = -0.4$, $a = 1.2$, $b = 0.3$, and these four values represent the secret key. By using Equation (11), two one-dimensional chaotic maps can be produced for use in creating a transform matrix to change the pixel locations of the original image and to obtain the shuffled image, *i.e.*, the encrypted image. Two test grayscale images with sizes of 512×512 were encrypted by using a Henon map with the above parameters to generate two encrypted images, while two grayscale watermark images were embedded into the encrypted images, respectively. The two watermark images we used are shown in Figure 7.

(a) watermark image 1 with size 57×57

(b) watermark image 2 with size 102×102

Figure 7: Two test watermark images

Figure 8 shows the test image "Lena" and its encrypted image and the corresponding two sets of shadow images. Figure 9 shows the case in which the test image was "Baboon". Then, in the reconstruction process, we used the two cooperative shadow images generated above to extract the logos and reconstruct the encrypted images successfully. We also used Equation (11) to restore the encrypted image exactly to its original state after obtaining the secret key used for encryption. We also used additional images to test the payload of the proposed scheme.

To illustrate the security of the proposed scheme, we will explain it from two different perspectives. One is the computational cost, and the other is the most common quantities, *i.e.*, the number of pixels change rate (*NPCR*)

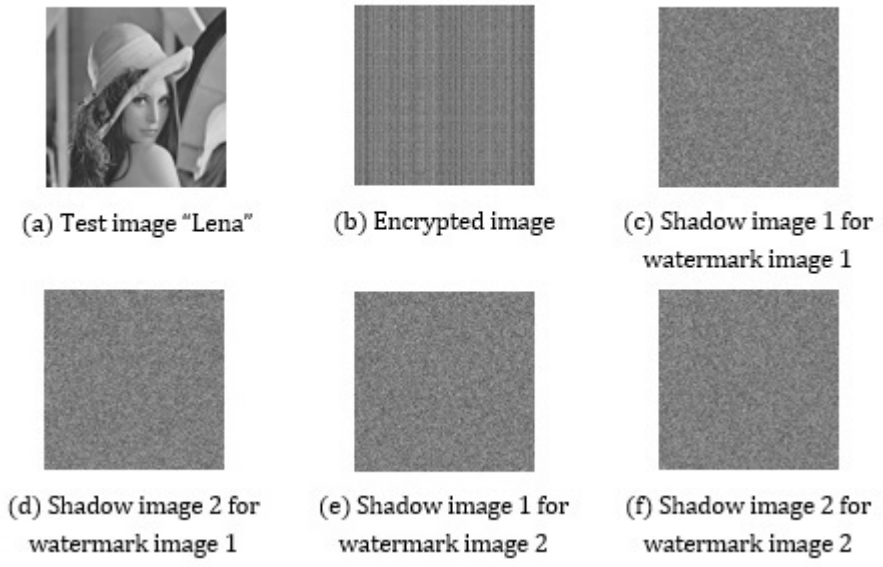

Figure 8: Example of the proposed scheme using the test image "Lena"

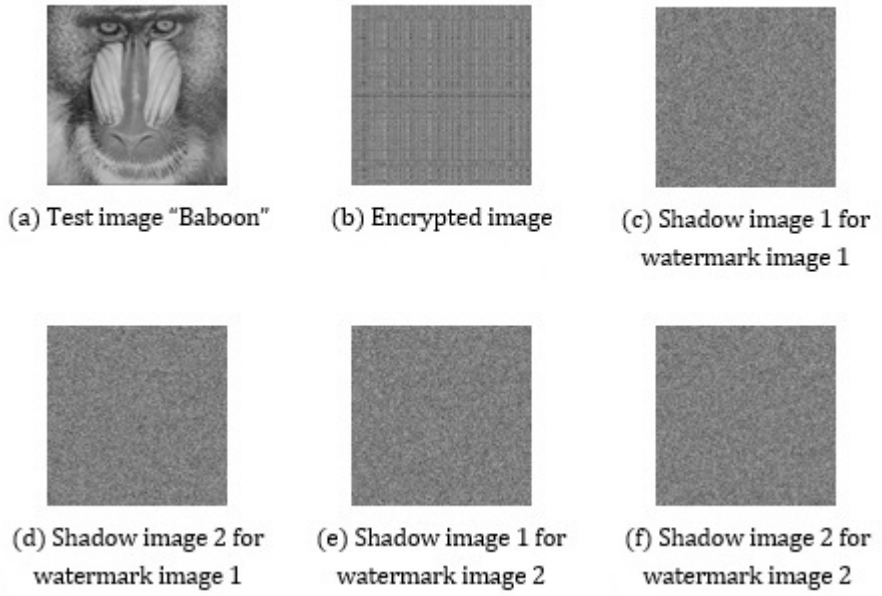

Figure 9: Example of the proposed scheme using the test image "Baboon"

and the unified average changing intensity ($UACI$). Assume that an attacker gets one shadow image, which won't leak any secret information. As described in Section 3 of the shadow images generation process, if he/she applies a brute-force attack to obtain the other shadow image, the possibility of success can be calculated as:

$$pb = \frac{1}{2^{M\times N\times 8}} = \frac{1}{256^{M\times N}}.$$

Obviously, this is very hard to accomplish. The computational cost of our method is mainly the XOR operation of the bit matrix in the image, which is linear, so the computational complexity of the proposed scheme is low. Next, we look at the second perspective, $NPCR$ and $UACI$. Generally, a high $NPCR/UACI$ score is interpreted as a stronger anti-attack performance, and they are the most common standardized tests for the security of an image. These two quantities were used in our experiment to test the two shadow images. Let us assume that we tested the two shadow images A^1 and A^2, respectively; the pixel value at corresponding positions were denoted as $A^1(i,j)$ and $A^2(i,j)$, and an array F is defined in Equation (12). Then, the $NPCR$ and $UACI$ are defined by Equations (13) and (14), respectively, where M and N are the width and height of the shadow images, L is the maximum pixel value, which is 255 for a grayscale image.

$$F(i,j) = \begin{cases} 0, \text{ if } A^1(i,j) = A^2(i,j) \\ 1, \text{ if } A^1(i,j) \neq A^2(i,j). \end{cases} \quad (12)$$

$$NPCR(A^1,A^2) = \sum\nolimits_{i,j} \frac{F(i,j)}{M\times N} \times 100\%. \quad (13)$$

$$UACI(A^1,A^2) = \sum\nolimits_{i,j} \frac{|A^1(i,j) - A^2(i,j)|}{M\times N\times L} \times 100\%. \quad (14)$$

The ranges of $NPCR$ and $UACI$ are all [0, 1]. If $NPCR(A^1,A^2) = 0$, it means that all of the pixels in A^2 have the same value as in A^1, and, if $NPCR(A^1,A^2) = 1$, then all of the corresponding pixels have different values in A^1 and A^2. Obviously, the ideal value for $NPCR$ is close to 1, but, for $UACI$, it is not obvious that the better value also is close to 1. However, it generally is believed that the expected $UACI$ value of a grayscale image is about 33%. Based on Eqs. 12- 14, we calculate the $NPCR$ and $UACI$ scores of the four sets of shadow images in the previous examples, and the data are shown in Table 2. It can be observed that the $NPCR$ scores of the shadow images are greater than 99.8%. It indicates that the two shadow images in each pair differ greatly. In addition, $UACI$ scores of the shadow images are higher than 34%. It is obvious that the average changing intensity of the corresponding pixels in the shadow images is very strong.

In terms of image embedding capacity, in addition to the previous two test images, we also tested several other images in our experiment. Figure 10 shows the other four test images and their encrypted images. Table 3 shows the payloads of all six encrypted images, from which it can be seen that the payload of the encrypted aircraft image reaches the maximum of 117805 bits and the encrypted baboon image has the smallest payload of 87272 bits. Thus, the experimental results show that proposed scheme can successfully share a secret image. It is a good method for the watermarking in the Cloud and achieving the secure sharing of secret information.

5 Conclusions

In this paper, a novel (2, 2) reversible secret sharing scheme is proposed, and the scheme was demonstrated by an application model of reversible watermarking in the Cloud. In this scheme, first, the original image is encrypted before uploading to the Cloud so that it cannot be leaked to any third party. Second, two shadows of the encrypted image with watermark information are generated. The test results in Table 2 indicate that $NPCR$ score and $UACI$ score are ideal. And from the experimental results in Table 3, we find that the average payload of these six encrypted images is 99185 bits. Finally, only through the collaboration of the two recipients can the exact watermark be determined and the encrypted image be obtained. The original image can be recovered without any damage, if desired.

References

[1] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of National Computer Conference, American Federation of Information Processing Societies*, pp. 313–317, June 1979.

[2] C. C. Chang, Y. C. Chou, and T. D. Kieu, "An information hiding scheme using sudoku," in *Proceedings of The Third International Conference on Innovative Computing Information and Control (ICICIC'08)*, pp. 17–21, June 2008.

[3] C. C. Chang, Y. P. Hsieh, and C. H. Lin, "Sharing secrets in stego images with authentication," *Pattern Recognition*, vol. 41, no. 10, pp. 3130–3137, 2008.

[4] C. C. Chang, T. D. Kieu, and Y. C. Chou, "Reversible data hiding scheme using two steganographic images," in *Proceedings of IEEE (TENCON'07)*, pp. 1–4, Oct. 2007.

[5] C. C. Chang, C. C. Lin, and N. T. Huynh, "Safeguarding visual information using (t, n) verifiable secret shares," *Journal of Computers*, vol. 22, no. 2, pp. 72–88, 2011.

[6] C. C. Chang, C. C. Lin, T. H. N. Le, and H. B. Le, "Sharing a verifiable secret image using two shadows," *Pattern Recognition*, vol. 42, no. 11, pp. 3097–3114, 2009.

[7] C. C. Chang, Y. J. Liu, and H. L. Wu, "Distortion-free secret image sharing method with two meaningful shadows," *IET Image Processing*, vol. 10, no. 8, pp. 590–597, 2016.

Table 2: Quantitative data for each pair of shadow images that were tested

Factor	Lenna with watermark image 1	Lenna with watermark image 2	Baboon with watermark image 1	Baboon with watermark image 2
$NPCR(\%)$	99.9722	99.8539	99.9626	99.8169
$UACI(\%)$	34.1083	34.1208	34.1164	34.1255

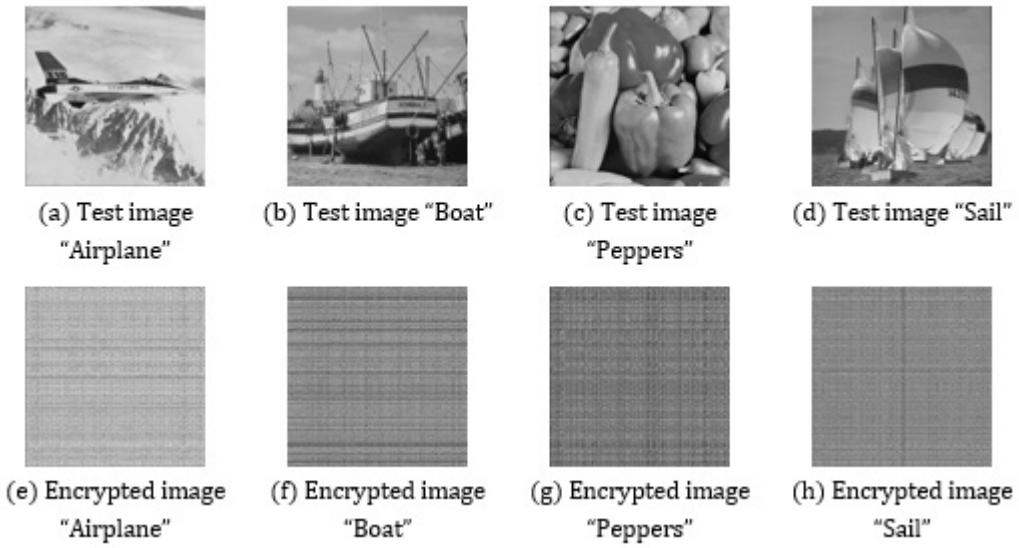

Figure 10: Examples of the test images and the corresponding encrypted images

Table 3: Performances of the test images

Factor	Encrypted Airplane	Encrypted Baboon	Encrypted Boat	Encrypted Lena	Encrypted Peppers	Encrypted Sail
Payloads(bits)	117805	87272	109139	90758	89269	100863

[8] C. F. Lee and Y. L. Huang, "Reversible data hiding scheme based on dual stegano-images using orientation combinations," *Telecommunications Systems*, vol. 52, no. 4, pp. 2237–2247, 2013.

[9] C. F. Lee, K. H. Wang, C. C. Chang, and Y. L. Huang, "A reversible data hiding scheme based on dual steganographic images," in *Proceedings of The Third International Conference on Ubiquitous Information Management and Communication*, pp. 228–237, Jan. 2009.

[10] C. C. Lin and W. H. Tsai, "Secret image sharing with steganography and authentication," *The Journal of Systems and Software*, vol. 73, no. 3, pp. 405–414, 2004.

[11] T. C. Lu, J. H. Wu, and C. C. Huang, "Dual-image-based reversible data hiding method using center folding strategy," *Signal Processing*, vol. 115, pp. 195–213, 2015.

[12] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

[13] M. Naor and A. Shamir, "Visual cryptography," *Lecture Notes in Computer Science*, vol. 950, pp. 1–12, 1995.

[14] Z. Qian and X. Zhang, "Reversible data hiding in encrypted images with distributed source encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636–646, 2016.

[15] A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, no. 11, pp. 612–613, 1979.

[16] C. C. Thien and J. C. Lin, "An image-sharing method with user-friendly shadow images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13 no. 12, pp. 1161–1169, 2003.

[17] S. Yi and Y. Zhou, "Binary-block embedding for reversible data hiding in encrypted images," *Signal Processing*, vol. 133, pp. 40–51, 2017.

[18] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

Biography

Yu Chen received the B.S. degree in Computer and Application from Hunan University, Hunan, China in 1993, and M.S. degree in Software Engineering from Fuzhou University, Fujian, China, in 2006. Currently, he is an associate professor in the School of Information Science and Engineering, Fujian University of Technology(FJUT), China. His current research interests include information retrieval, data mining, and digital image processing.

Jiang-Yi Lin received the B.S. and M.S. degrees in Computer science and Technology from FuZhou Uniersity, FuJian, China, in 2005 and 2008, repectively. He is currently pursuing the Ph.D degree with the Multimedia and Secure Networking Laboratory (MSN lab), the Department of Information Engineering and Computer Science of Feng Chia University, Taichung, Taiwan. His research interests include image processing, secret sharing and steganography.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from February 2005. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And, since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression, and data structures.

Yu-Chen Hu received his PhD. degree in computer science and information engineering from the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan in 1999. Currently, Dr. Hu is a professor in the Department of Computer Science and Information Management, Providence University, Sha-Lu, Taiwan. He is a senior member of IEEE. He is also a member of Computer Vision, Graphics, and Image Processing (CVGIP), Chinese Cryptology and Information Security Association (CCISA), Computer Science and Information Management (CSIM) and Phi Tau Phi Society of the Republic of China. He servers as the Editor-in-Chief of International Journal of Image Processing from June 2009 to May 2015. In addition, he is the managing editor of Journal of Information Assurance & Security since March 2009. He is the associated editor of Human-centric Computing and Information Sciences since Feb. 2011. He joints the editorial boards of several other journals. His research interests include digital forensics, information hiding, image and signal processing, data compression, information security, and data engineering.