Mathematical Biosciences and Engineering

Mathematical Biosciences and Engineering, 16(4): 1914–1932
DOI: 10.3934/mbe.2019093
Received: 07 November 2018
Accepted: 27 December 2018
Published: 07 March 2019

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

Research article

Security protection using two different image shadows with authentication

Yanjun Liu [1], Chin-Chen Chang [1,*] and Peng-Cheng Huang [1, 2]

[1] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan
[2] College of Computer and Information Engineering, Xiamen University of Technology, Xiamen, Fujian, China

* **Correspondence:** Email: alan3c@gmail.com; Tel: +886-4-2451-7250 ext. 3790; Fax: +886-4-2706-6495.

Abstract: Secret sharing is an important technique for security protection for multimedia information communication by dividing a secret message into several shadows that are held among a set of participants. In this paper, we introduce a novel secret image sharing (SIS) scheme using two meaningful digital images with cheating detection. It allows a dealer to share a secret message into two different meaningful images through the guidance of the turtle shell magic matrix. Then, after performing a permutation operation, two meaningful shadow images are generated and distributed to two participants. The secret message can be reconstructed only when both participants cooperate by releasing real shadow images. Honest participant in the proposed scheme can easily detect whether the other participant is cheating via presenting a fake shadow. Experimental results show that this method ensures high quality of shadow images and good embedding capacity. The cheating detection process is also effective and easy to implement.

Keywords: security protection; secret image sharing (SIS); meaningful shadow; turtle shell matrix; cheating detection

1. Introduction

With the expeditious advance of Internet and multimedia processing technology, it is very convenient to transmit a great amount of digital information through the Internet. Unfortunately, if confidential or sensitive information is directly exposed to the public network, it could be placed at serious risk of being intercepted and forged intentionally. Thus, how to effectively protect the transmitted message has emerged as an important and challenging issue.

Data hiding [1], also called steganography, is a widely used technique for secure data delivery in such a way that secret data is first concealed in a cover medium to generate a stego-medium and then transmitted to the receiver. Cover media includes images, videos, voice, etc., and the digital image is regarded as the most commonly employed cover medium. One of the attractive characteristics of data hiding is that the dissimilarity between the original cover image and the stego-image containing secret data is too slight to be visually distinguishable, which will not rouse suspicions among malicious attackers.

In contrast with data hiding, secret sharing is another prevalent data protection mechanism based on a different perspective. In 1979, Shamir [2] and Blakley [3] first introduced the concept of secret sharing and proposed the (*t*, *n*) threshold secret sharing scheme where $t \leq n$, respectively. Such (*t*, *n*) secret sharing should satisfy the following requirements: (1) A dealer divides the secret into *n* parts, each of which is called a shadow; (2) each shadow is held by a participant; (3) any *t* or more involved participants can release their shadows and then cooperate to reveal the secret; and (4) the secret cannot be reconstructed by the collaboration of fewer than *t* shadows. Inspired by (*t*, *n*) secret sharing, Naor and Shamir [4] presented the first secret image sharing (SIS) scheme that shares a secret image into *n* shadow images and then distributed them among *n* authorized participants. The secret image can be reconstructed by just stacking *t* or more shadow images, rather than performing time-consuming computation operations. However, this SIS scheme [4] has three main deficiencies: (1) it can only be applied to binary images; (2) it can result in a pixel expansion problem; and (3) shadow images are user-unfriendly. Consequently, many enhanced SIS schemes were proposed in the past decades to overcome these drawbacks [5–8]. Unfortunately, the shadow images produced in [5,6] were still noise-like, which were very difficult to handle and easily attracted the attention of attackers. By contrast, the shadow images in [7,8] were easier to manage due to the fact that they were meaningful. However, these shadow images appeared just as the degradation versions of the secret image, thereby leading to a serious security problem that the secret information may be leaked from shadow images. Therefore, how to enhance the security of shadow images has become an extremely urgent problem facing us.

At present, the data hiding technique is extensively applied to SIS schemes for further protection of shadow images [9,10]. In those SIS schemes, one or more meaningful cover images are slightly modified to create several shadow images by the data hiding technique for embedding secret information. The shadow images that contain secret information can achieve high visual quality such that it is impossible to visually perceive the existence of the shared secret thanks to the properties of data hiding technique. Therefore, the chance of suspicion on shadow images is significantly reduced. However, these SIS schemes are vulnerable to cheating attacks. A dishonest participant may tamper with his/her shadow image and presents it to other authorized participants. In this scenario, wrong secret information is obtained via the cooperation of authorized participants. Thus, a cheating detection process should be implemented in SIS schemes and many SIS schemes combining data hiding and cheating detection are proposed [11,12]. However, these schemes are complicated with high computational complexity.

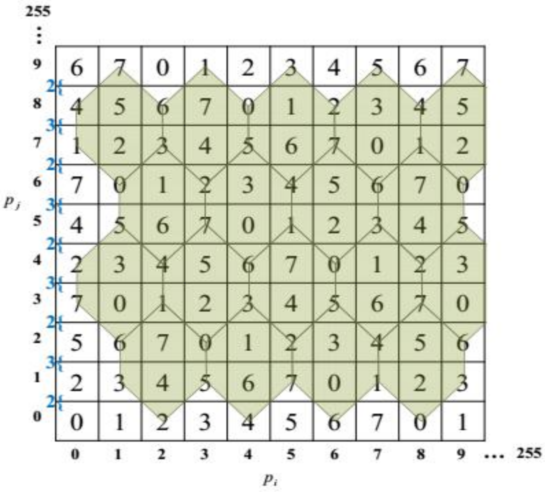
Recently, Chang et al. [13] proposed the first SIS scheme based on magic matrix. Magic matrix is a popular data hiding technique and very easy to implement. Moreover, both the dealer and the participants do not need to pre-store the magic matrix since each of them can construct it independently. This will not cause any auxiliary information and extra storage overhead for secret sharing. Due to these advantages, diverse magic matrices, such as exploiting-modification-direction (EMD) matrix [14] and Sudoku matrix [15], are employed in SISs. Unfortunately, these magic matrix based SIS schemes cannot detect cheaters.

In this paper, we apply a novel magic matrix, called turtle shell matrix [16], in the design of an SIS scheme with cheating detection for the first time. The motivation of using turtle shell matrix is that it is the most favorable option for data hiding due to the merit that both very low distortion of the stego-image and very high embedding capacity can be obtained without time-consuming computations. Under the help of the turtle shell matrix, secret message is shared into two different meaningful shadow images that are constructed by distinct cover images. To ensure high security, shadow images are modified via a permutation operation and then distributed to two participants. The secret message can be losslessly reconstructed only when both participants cooperate by releasing real shadow images. Honest participant in the proposed scheme can easily detect whether the other participant is cheating according to the turtle shell matrix. Experimental results show that the proposed scheme ensures high quality of shadow images and good embedding capacity. Furthermore, the cheating detection process is efficient and easy to implement.

The rest of this paper is organized as follows. Section 2 introduces the concept of turtle shell matrix and its unique features. Section 3 presents the proposed SIS scheme that uses turtle shell matrix to achieve cheating detection. The performance of the proposed scheme is tested in Section 4 and the paper is concluded in Section 5.

2. Brief introduction of turtle shell matrix

The turtle shell matrix is first introduced by Chang et al. and used in their data hiding method [16]. As illustrated in Figure 1, the turtle shell matrix, denoted as *M*, is a 256×256 matrix that is composed of a large amount of adjacent hexagons called turtle shells. Values p_i and p_j on horizontal and vertical coordinate axes represent grayscale pixel values ranging from 0 to 255. We denote the element at the location of row p_j and column p_i in *M* as $M(p_i, p_j)$. Each element in *M* indicates a to-be-embedded secret digit in base-8 numeral system. Constructing *M* should comply with two rules as follows. (1) For the two consecutive elements in the same row, the right one is greater than its left neighbor plus 1. (2) For the two consecutive elements in the same column, the upper one is larger than the lower one plus 2 and 3 by turns. Thus, a turtle shell includes eight different digits in the range of [0,7], among which six digits are on the edge and two on the back. As can be seen later in the next section, this turtle shell matrix *M* will be employed as an essential constructional unit in the proposed scheme.



p_j \ p_i	0	1	2	3	4	5	6	7	8	9
9	6	7	0	1	2	3	4	5	6	7
8	4	5	6	7	0	1	2	3	4	5
7	1	2	3	4	5	6	7	0	1	2
6	7	0	1	2	3	4	5	6	7	0
5	4	5	6	7	0	1	2	3	4	5
4	2	3	4	5	6	7	0	1	2	3
3	7	0	1	2	3	4	5	6	7	0
2	5	6	7	0	1	2	3	4	5	6
1	2	3	4	5	6	7	0	1	2	3
0	0	1	2	3	4	5	6	7	0	1

Figure 1. Turtle shell matrix [16]**.**

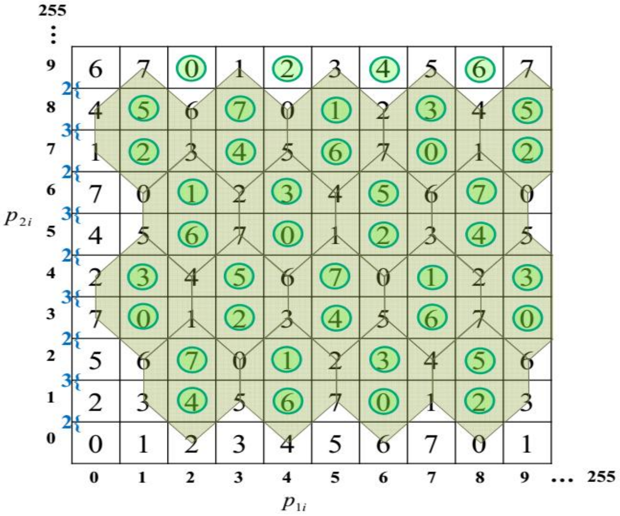
3. The proposed VSS scheme

In this section, we propose an SIS scheme that shares secret message into two different meaningful shadow images. The proposed scheme is based on turtle shell matrix with cheating detection. Assume that the grayscale secret image S is represented by a binary stream which can be divided into a sequence of 3-bit segments. Thus, we denote S as $S=\{sg_k \,|k=1,2,...,n\}$, where sg_k is a 3-bit segment and n is the number of segments. Two distinct grayscale cover images with $H\times W$ pixels are denoted as $C_1=\{p_{1i} \,|i=1,2,...,H\times W\}$ and $C_2=\{p_{2i} \,|i=1,2,...,H\times W\}$, respectively. After embedding secret message, cover images C_1 and C_2 are converted to shadow images S_1 and S_2 with the same size of $H\times W$, respectively, where S_1 is denoted as $S_1=\{p'_{1i} \,|i=1,2,...,H\times W\}$ and S_2 is represented as $S_2=\{p'_{2i} \,|i=1,2,...,H\times W\}$.

3.1. Overview

The proposed scheme consists of two phases: (1) share construction phase and (2) cheating detection and secret extraction phase. In the share construction phase, each 3-bit secret segment sg_k (i.e., a base-8 digit) is embedded into a cover pixel pair (p_{1i}, p_{2i}) in which p_{1i} comes from C_1 and p_{2i} comes from C_2. After embedding, cover pixel pair (p_{1i}, p_{2i}) is modified to its corresponding shadow pixel pair (p'_{1i}, p'_{2i}), where p'_{1i} and p'_{2i} belong to S_1 and S_2, respectively. In our proposed scheme, we have to comply with a main embedding principle that each shadow pixel pair

(p'_{1i}, p'_{2i}) should be mapped on the back of a turtle shell in matrix *M*. In other words, the mapping element $M(p'_{1i}, p'_{2i})$ can only occur in the circle-marked position in Figure 2.



p_{2i} \ p_{1i}	0	1	2	3	4	5	6	7	8	9
9	6	7	0	1	2	3	4	5	6	7
8	4	5	6	7	0	1	2	3	4	5
7	1	2	3	4	5	6	7	0	1	2
6	7	0	1	2	3	4	5	6	7	0
5	4	5	6	7	0	1	2	3	4	5
4	2	3	4	5	6	7	0	1	2	3
3	7	0	1	2	3	4	5	6	7	0
2	5	6	7	0	1	2	3	4	5	6
1	2	3	4	5	6	7	0	1	2	3
0	0	1	2	3	4	5	6	7	0	1

Figure 2. Embedding principle.

Since cheating detection is another very important issue to be addressed, now we investigate the following scenario and consider whether the aforementioned embedding principle is feasible. Let participants U_1 and U_2 hold shadow images S_1 and S_2, respectively. To extract the original secret message, both participants have to release their shadow images. Assume that S_1 released by U_1 is a real one while S_2 presented by U_2 has been modified. Thus, there is a high probability that some $M(p'_{1i}, p'_{2i})$'s are no longer on the back of turtle shells. In other words, if we found that there exists an $M(p'_{1i}, p'_{2i})$ on the edge of a turtle shell, we can immediately figure out that S_2 is fake. However, if U_2 knows the embedding principle, U_2 can intentionally modify S_2 to ensure that each $M(p'_{1i}, p'_{2i})$ is shifted to the back of another turtle shell. If it holds, the cheater cannot be detected and wrong secret message is retrieved. Based on the above analysis, the embedding principle should be modified. A permutation operation is conducted on shadow pixels after embedding in such a way that $M(p'_{1i}, p'_{2i})$ can be on either the back or the edge of a turtle shell. As a result, the cheater has no idea of the embedding rule and the possibility of successful deception becomes very low.

In the next two subsections, we will elaborate the detailed steps of the two phases.

3.2. Share Construction Phase

In this phase, firstly, the turtle shell matrix is established and a permutation operation is generated. Then, according to the turtle shell matrix, the secret message is embedded into two different cover images to create two visually distinct shadow images. Finally, in order to ensure high security, the generated permutation operation is performed on shadow pixel values. Thus, the share construction phase is divided into three stages as follows: (1) pre-processing; (2) secret embedding; and (3) permutation.

(1) Pre-processing

There are two objectives of pre-processing: construction of the turtle shell matrix and generation of the permutation. The concrete steps are shown as follows.

Step 1. Establish the turtle shell matrix *M* as illustrated in Figure 1.

Step 2. Randomly select a decimal integer *a*.

Step 3. Use *a* as a seed to create a set *B* that contains 256 pseudo-random numbers ranging from 1 to 1000.

Step 4. Divide *B* into 64 subsets, each of which includes 4 numbers. Let $B_l = \{b_{lj} \,|\, l = 0,1,...,63,\ j = 0,1,2,3\}$ denote the *l*th subset. Sort the numbers in B_l by an ascending order and record the position of each number in the sorted subset as $PS(b_{lj})$, where $PS(b_{lj}) = 0,1,2,3$. Define $ID(b_{lj}) = PS(b_{lj}) + 4l$.

Step 5. Let the set $D = \{0,1,...,255\}$ and divide it into 64 subsets such that each subset is denoted as $D_l = \{4l, 4l+1, 4l+2, 4l+3\}$, where $l = 0,1,...,63$.

Step 6. Generate a permutation σ_l of the subset D_l as

$$\sigma_l = \begin{pmatrix} 4l & 4l+1 & 4l+2 & 4l+3 \\ ID(b_{l0}) & ID(b_{l1}) & ID(b_{l2}) & ID(b_{l3}) \end{pmatrix}. \tag{1}$$

Now we give an example to better understand how to generate the permutation. Assume that we select an integer $a = (153)_{10}$ and use it to generate a set *B* which includes 256 pseudo-random numbers. Let $B = \{3, 45, 93, 42, 88, 47, 39, 12, 65, 125, 60, ...\}$ and we just pick up the first eight numbers in *B* to explain the permutation. In this case, the first four numbers compose the subset B_0 and the next four numbers compose B_1. That is, $B_0 = \{b_{00}, b_{01}, b_{02}, b_{03}\} = \{3, 45, 93, 42\}$ and $B_1 = \{b_{10}, b_{11}, b_{12}, b_{13}\} = \{88, 47, 39, 12\}$.

First, let us take a look at the numbers in B_0. By Step 4, $\{b_{00}, b_{01}, b_{02}, b_{03}\} = \{3, 45, 93, 42\}$ is sorted as $\{b_{00}, b_{03}, b_{01}, b_{02}\} = \{3, 42, 45, 93\}$ in an ascending order. Therefore, the position of each number after sorting is that $PS(b_{00}) = 0$, $PS(b_{01}) = 2$, $PS(b_{02}) = 3$ and $PS(b_{03}) = 1$. On the other hand, the subset $D_0 = \{0,1,2,3\}$. Since $ID(b_{0j}) = PS(b_{0j})$ when $l = 0$, the permutation σ_0 of D_0 is generated as

$$\sigma_0 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ ID(b_{00}) = 0 & ID(b_{01}) = 2 & ID(b_{02}) = 3 & ID(b_{03}) \end{pmatrix}. \tag{2}$$

Similarly, in the sorted set B_1, the position of each number is $PS(b_{10}) = PS(88) = 3$, $PS(b_{11}) = PS(47) = 2$, $PS(b_{12}) = PS(39) = 1$ and $PS(b_{13}) = PS(12) = 0$. Then, we can compute $ID(b_{10}) = PS(b_{10}) + 4 = 7$, $ID(b_{11}) = PS(b_{11}) + 4 = 6$, $ID(b_{12}) = PS(b_{12}) = 5$, $ID(b_{13}) = PS(b_{13}) + 4 = 4$. Consequently, the permutation σ_1 of D_1 ($D_1 = \{4, 5, 6, 7\}$) is created as

$$\sigma_1 = \begin{pmatrix} 4 & 5 & 6 & 7 \\ ID(b_{10}) = 7 & ID(b_{11}) = 6 & ID(b_{12}) = 5 & ID(b_{13}) \end{pmatrix}. \tag{3}$$

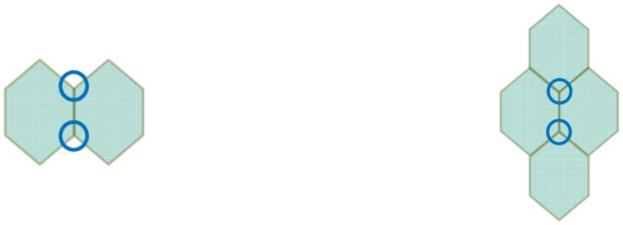
Based on the same construction method as well as Equation (1), σ_2 to σ_{63} can be successfully created.

(2) Secret embedding

The main goal of this stage is to embed secret message by slightly modifying the cover images based on the turtle shell matrix *M*. First of all, for each cover pixel pair (p_{1i}, p_{2i}), where p_{1i} and p_{2i} come from cover images C_1 and C_2, respectively, it is mapped to the element $M(p_{1i}, p_{2i})$ in the turtle shell matrix. Then, we determine the ***candidate turtle shells*** for $M(p_{1i}, p_{2i})$ according to the location of $M(p_{1i}, p_{2i})$ in *M* to check whether the cover pixel pair (p_{1i}, p_{2i}) is embeddable. There are two cases to consider.

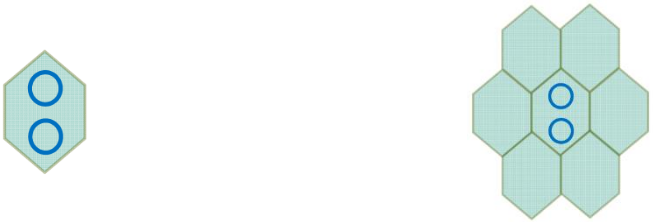
Case 1: $M(p_{1i}, p_{2i})$ is on the common edge of two adjacent turtle shells in the horizontal direction as marked by the circle in Figure 3 (a). If the rocket-shaped candidate turtle shells shown in Figure 3 (b) can be found, (p_{1i}, p_{2i}) is embeddable; otherwise, (p_{1i}, p_{2i}) is unembeddable.

Case 2: $M(p_{1i}, p_{2i})$ is on the back of a turtle as marked by the circle in Figure 4 (a). If the flower-shaped candidate turtle shells shown in Figure 4 (b) can be found, (p_{1i}, p_{2i}) is embeddable; otherwise, (p_{1i}, p_{2i}) is unembeddable.



(a) Two possible locations of $M(p_{1i}, p_{2i})$ (b) Candidate turtle shells

Figure 3. Case 1.



(a) Two possible locations of $M(p_{1i}, p_{2i})$ (b) Candidate turtle shells

Figure 4. Case 2.

Before embedding the secret message, the integer *a* selected in the pre-processing process must be embedded. The purpose of doing this is that the pseudo-random numbers created by *a* are essential to recover the original secret message. If *a* is regarded as part of the embedded message, *a* can be directly derived by the participants from the shadow images to create the same pseudo-random numbers as that in the pre-processing stage. Therefore, it is unnecessary for the dealer to send the pseudo-random numbers as auxiliary information to the participants for extracting the secret message. Embedding is composed of two steps as follows.

(a) Embed the integer *a*. Convert *a* to a group of three base-8 digits d_1, d_2, d_3 and conceal each digit into an embeddable cover pixel pair.

(b) Embed the secret message $S = \{sg_k \,|k = 1, 2, ..., n\}$. In particular, each 3-bit secret segment sg_k equal to a base-8 digit is concealed into an embeddable cover pixel pair.

Since the rules of embedding d_1, d_2, d_3 and sg_k are the same, we now let h represent d_1, d_2, d_3 or sg_k. The rule for embedding a base-8 digit h into a cover pixel pair (p_{1i}, p_{2i}) is described as follows.

(a) Check which case the pair (p_{1i}, p_{2i}) is in. If it is an embeddable pair in Case 1, obtain the rocket-shaped candidate turtle shells; if it is an embeddable pair in Case 2, get the flower-shaped candidate turtle shells.

(b) Find an element $M(x, y)$ on the back of candidate turtle shells which satisfies two requirements: 1) $M(x, y) = h$ and 2) $M(x, y) = h$ has the shortest distance with $M(p_{1i}, p_{2i})$.

(c) Set the shadow pixel pair $(p'_{1i}, p'_{2i}) = (x, y)$, where p'_{1i} and p'_{2i} belong to shadow images S_1 and S_2, respectively.

When the secret embedding stage is completed, all shadow pixel pairs are mapped on the back of turtle shells in matrix *M*. Moreover, the distortion of shadow images is very low due to the fact that the range of the modification to the cover images is limited within the candidate turtle shells.

We will take the following example to illustrate the secret embedding more clearly. Suppose the selected integer *a* is still 153 and the secret message $S = \{sg_1 = 111, sg_2 = 010\}$. Cover images C_1 and C_2 are shown in Figures 5 (a) and 5 (b), respectively.

2	5	5	2	6

(a) Cover image C_1

4	3	4	3	5

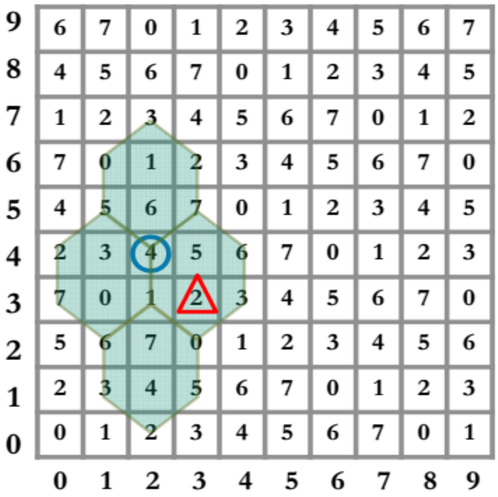
(b) Cover image C_2

Figure 5. Cover images in an example.

In the following, we will use C_1 and C_2 to first embed *a* and then embed *S*. To embed *a*, we must convert it to a group of three base-8 digits such that $a = (153)_{10} = (010\ 011\ 001)_2 = (2\ 3\ 1)_8$. Each of the three digits are embedded into an cover pixel pair.

(i) Embed the first digit “2” of *a*

Take out the first pixel “2” from C_1 and the first pixel “4” from C_2 to compose a pair (2, 4) and map it onto the element $M(2,4)$ in the turtle shell matrix *M*, as shown in the circle-marked value in Figure 6. After checking, we confirm that (2, 4) is an embeddable pair in Case 1, thus we obtain the rocket-shaped candidate turtle shells. Since there is only one back element $M(3,3)$ equaling the to-be-embedded digit “2”, Thus, the cover pixel pair (2, 4) is modified to its corresponding shadow pixel pair (3, 3) as shown in the triangle-marked value in Figure 6, where the first pixel “3” and the second pixel “3” belong to the shadow image S_1 and S_2, respectively.



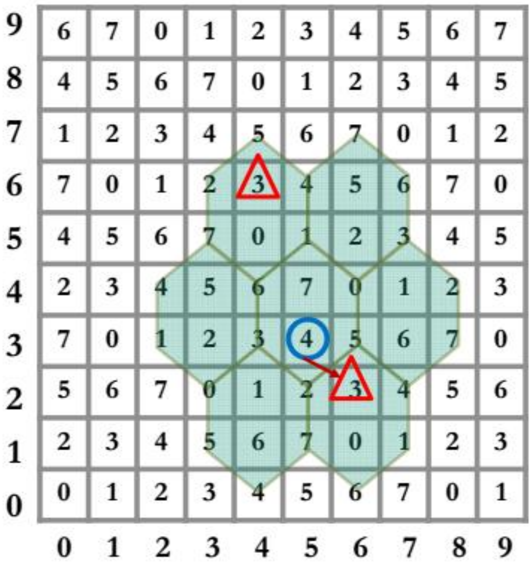
9	6	7	0	1	2	3	4	5	6	7
8	4	5	6	7	0	1	2	3	4	5
7	1	2	3	4	5	6	7	0	1	2
6	7	0	1	2	3	4	5	6	7	0
5	4	5	6	7	0	1	2	3	4	5
4	2	3	4	5	6	7	0	1	2	3
3	7	0	1	2	3	4	5	6	7	0
2	5	6	7	0	1	2	3	4	5	6
1	2	3	4	5	6	7	0	1	2	3
0	0	1	2	3	4	5	6	7	0	1
	0	1	2	3	4	5	6	7	8	9

Figure 6. Embedding example 1.

(ii) Embed the second digit “3” of *a*

Take out the second pixel “5” from C_1 and the second pixel “3” from C_2 to compose a pair (5, 3) and map it onto the element $M(5,3)$ in the turtle shell matrix *M*, as shown in the circle-marked value in Figure 7. Because the pair (5, 3) is embeddable according to Case 1, we obtain the flower-shaped candidate turtle shells. As shown in the triangle-marked values in Figure 7, two back elements, $M(4,6)$ and $M(6,2)$, that are equal to the to-be-embedded digit “3” are found. Obviously, $M(6,2)$ is nearer to $M(5,3)$ such that $M(6,2)$ is selected and the cover pixel pair (5, 3) is modified to its corresponding shadow pixel pair (6, 2), where the pixels “6” and “2” belong to the shadow images S_1 and S_2, respectively.

Here, we just display how to embed the first two digits of a. The third digit of a and the secret message S can be embedded by adhering to the same rule. When the embedding is completed, the shadow images S_1 and S_2 are produced and shown in Figure 8.



9	6	7	0	1	2	3	4	5	6	7
8	4	5	6	7	0	1	2	3	4	5
7	1	2	3	4	5	6	7	0	1	2
6	7	0	1	2	3	4	5	6	7	0
5	4	5	6	7	0	1	2	3	4	5
4	2	3	4	5	6	7	0	1	2	3
3	7	0	1	2	3	4	5	6	7	0
2	5	6	7	0	1	2	3	4	5	6
1	2	3	4	5	6	7	0	1	2	3
0	0	1	2	3	4	5	6	7	0	1
	0	1	2	3	4	5	6	7	8	9

Figure 7. Embedding example 2.

3	6	7	2	6

(a) Shadow image S_1

3	2	4	2	5

(b) Shadow image S_2

Figure 8. Shadow images in an example.

(3) Permutation

As discussed in Subsection 3.1, both shadow images need to be modified by the permutation generated in the pre-processing stage to enhance security. It is worth noting that the first three pixels in the shadow images are not allowed to be modified since they are used to directly recover the integer a when extracting the secret message. Therefore, the modification is conducted from the fourth shadow pixel as follows:

Step 1. Read a shadow pixel from S_1 or S_2 and assume its value is q.

Step 2. Determine the set D_l $(l = 0,1,...,63)$ where the value q exists.

Step 3. Obtain the permutation σ_l of D_l by Equation (1).

Step 4. Map q to a value according to σ_l.

After all the shadow pixels are processed, the mapping values of some shadow pixel pairs are changed onto the edge of turtle shells in matrix *M*. Then, the dealer distributes the modified shadow images to two participants. Here we take the fourth and fifth pixels in the shadow image S_1 in Figure 8 for example to show the modification through permutation. The fourth pixel "2" in S_1 exists in the set $D_0 = \{0,1,2,3\}$, thus "2" is mapped to "3" via the permutation σ_0 as shown in Equation (2). By the similar way, considering that the fifth pixel "6" in S_1 is a component of the set $D_1 = \{4,5,6,7\}$, "6" is mapped to "5" by the permutation σ_1 as shown in Equation (3). The shadow images S_1 and S_2 after modification are illustrated in Figure 9. In this example it is worth noting that the original fifth shadow pixel pair (6, 5) is mapped on the back of a turtle shell. However, it is changed onto the edge after being modified.

3	6	7	3	5

(a) Modified shadow image S_1

3	2	4	3	6

(b) Modified shadow image S_2

Figure 9. Modified shadow images in an example.

3.3. Cheating Detection and Secret Extraction Phase

The participants work together with their shadow images to extract the secret message. Unfortunately, the scenario that one of the participants is dishonest and he/she forges his/her shadow image may happen. In the proposed scheme, once the honest participant detects that the other participant is cheating and releasing a fake shadow, the extraction of the secret message is immediately stopped.

Firstly, the participants get the first three shadow pixel pairs from S_1 and S_2 and obtain their mapping values on the turtle shell matrix *M* to retrieve the integer *a*. Then, they use *a* to create the permutations σ_0 to σ_{63} complying with the same method in the share construction phase and immediately obtain the inverse permutations.

After that, the participants handle the remainder of shadow pixel pairs. Suppose the currently selected shadow pixel pair is (*t*, *r*). The values *t* and *r* are mapped back to t' and r', respectively, according to their corresponding inverse permutation. If the mapping element $M(t',r')$ of the pair (t',r') is located on the back of a turtle shell in matrix *M*, the embedded base-8 secret digit is equal to the value of $M(t',r')$. Otherwise, if the mapping element is located on the edge of a turtle shell, it indicates that a cheater is detected and thus the extraction of the secret message is stopped right away.

4. Experimental results

This section will evaluate the performance of the proposed SIS scheme. In our experiments, we share the grayscale secret image "Airplane" (see Figure 10) into two different grayscale cover images. Here, eight groups of 512×512 cover images shown in Figure 11 are used. 1) Group 1: "Lena" and "Baboon"; 2) Group 2: "Peppers" and "Barbara"; 3) Group 3: "Boat" and "Goldhill"; 4) Group 4: "Washsat" and "Zelda"; 5) Group 5: "Elaine" and "Family"; 6) Group 6: "Girl" and "Office"; 7) Group 7: "Portofino" and "Sail"; and 8) Group 8: "Sailboat" and "Bridge".

Figure 10. Secret image "Airplane".

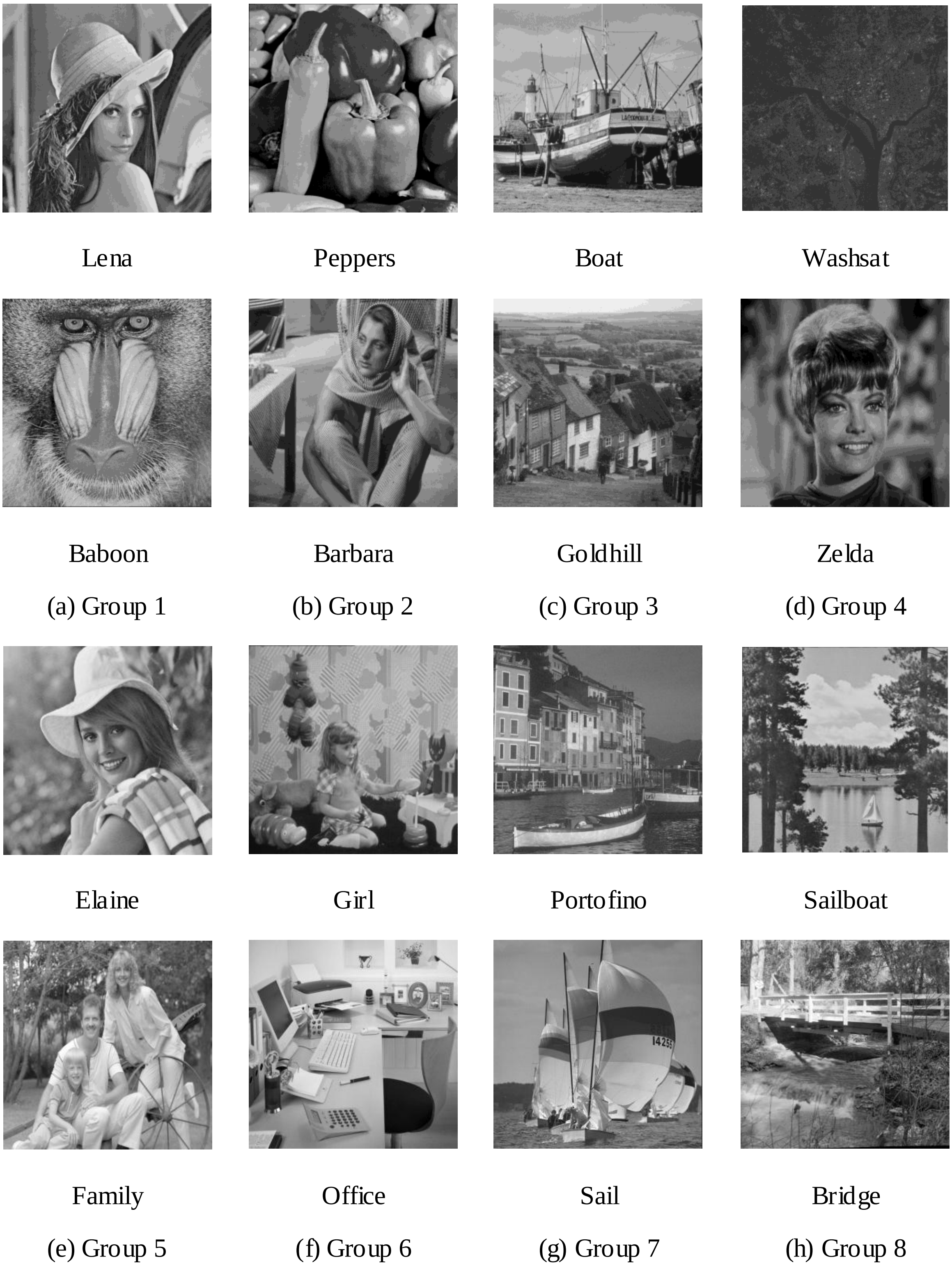
The visual quality of shadow images is the most important criterion for evaluating the performance of a SIS scheme. Without loss of generality, the peak-signal-to-noise ratio (PSNR) is adopted to measure the visual quality of the shadow image after embedding secret message. PSNR is defined as follows:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)(dB), \tag{4}$$

where *MSE* is the mean square error between the original cover image and the shadow image. *MSE* is defined as:

$$MSE = \frac{1}{H\times W}\sum_{i=1}^{H\times W}(p_i - p_i')^2 \tag{5}$$

where $H\times W$ represents the size of a grayscale cover image while p_i and p_i' denote the pixel values of the original cover image and the shadow image, respectively. The larger the *PSNR* value is, the better the visual quality is. We say the visual quality is good when the *PSNR* value is greater than 30 dB since the distortion on the shadow images cannot perceived by human visual system.



Lena Peppers Boat Washsat

Baboon Barbara Goldhill Zelda

(a) Group 1 (b) Group 2 (c) Group 3 (d) Group 4

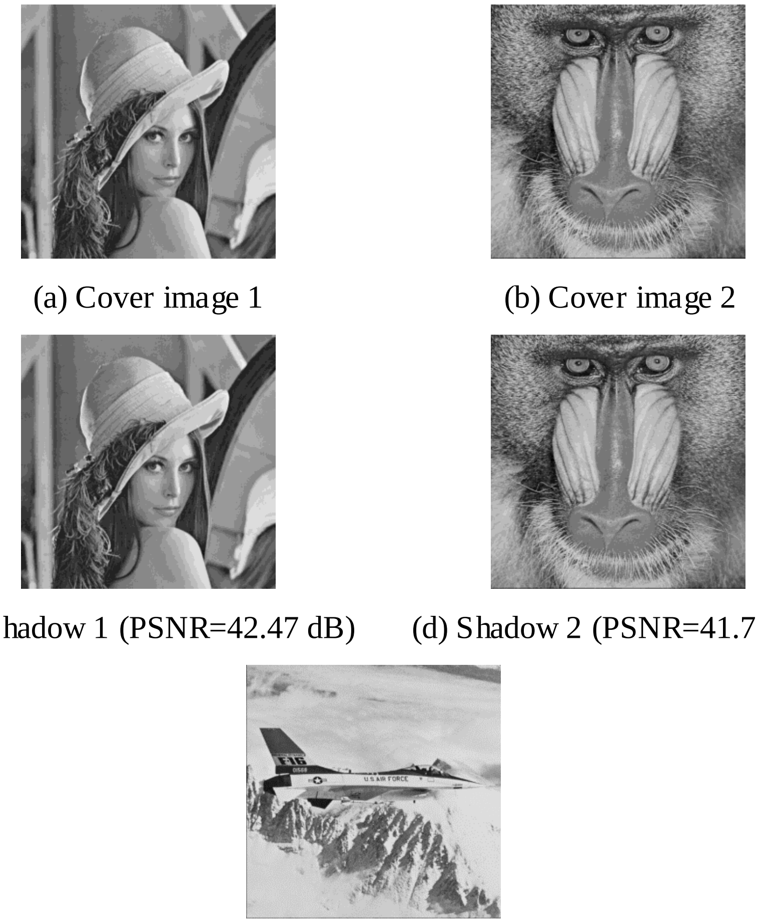
Elaine Girl Portofino Sailboat

Family Office Sail Bridge

(e) Group 5 (f) Group 6 (g) Group 7 (h) Group 8

Figure 11. Eight groups of cover images.

In the proposed scheme, the distortion of shadow images is very low due to the fact that the range of the modification to the cover images is limited within the candidate turtle shells. Moreover, the permutation operation used to further modify shadow images will not enlarge the distortion. Figures 12 (c) and (d) show the shadow images after the secret image “Airplane” is shared into images “Lena” and “Baboon” in Group 1. Since the size of the shadow is the same as that of the cover image, the proposed scheme does not cause the pixel expansion problem. Both shadow images obtain very good visual quality which are greater than 41 dB. As shown in Figure 12 (e), the secret image can be recovered without distortion when no cheater exists. Figure 13 displays the experimental results when the secret image “Airplane” is shared into images “Peppers” and “Barbara” in Group 2. Table 1 summaries the visual quality and the embedding capacity for all of eight test image groups. Here, the embedding capacity is defined as the total secret bits that can be concealed. From Table 1, we can see that the proposed scheme can achieve both great visual quality and high embedding capacity.



(a) Cover image 1

(b) Cover image 2

(c) Shadow 1 (PSNR=42.47 dB)

(d) Shadow 2 (PSNR=41.72 dB)

(e) Recovered secret image

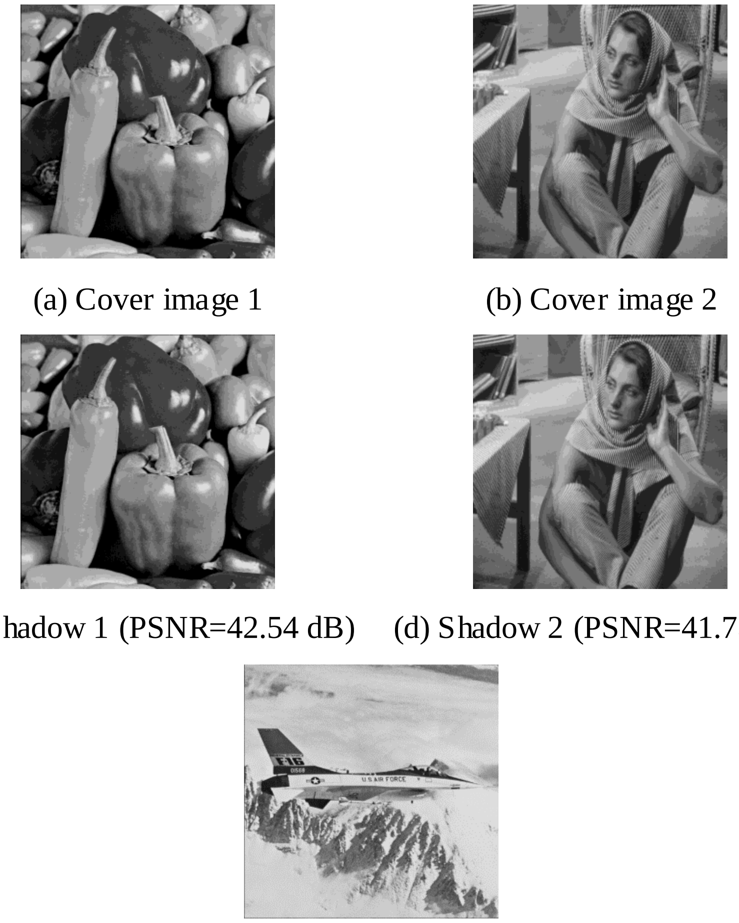
Figure 12. Experimental results 1.

In the following, we show that the proposed scheme can easily achieve cheating detection. Assume that only one of the two participants is a cheater and he/she has tampered with his/her shadow image. By the cooperation of both shadow images, the honest participant can detect whether the other participant is a cheater. In our experiments, the tampered region in a shadow image is an image

"Tiffany". Figures 14–17 illustrate the detection results by the collaboration of a real shadow and a fake shadow.

Table 1. Visual quality and embedding capacity of the proposed SIS scheme.

	Cover image 1	Cover image 2	PSNR (dB)		Embedding capacity (bits)
			Shadow image 1	Shadow image 2	
Group 1	Lena	Baboon	42.47	41.72	786,166
Group 2	Peppers	Barbara	42.54	41.73	783,217
Group 3	Boat	Goldhill	42.87	41.65	786,421
Group 4	Washsat	Zelda	41.80	41.57	786,286
Group 5	Elaine	Family	41.37	41.16	784,537
Group 6	Girl	Office	41.55	41.14	786,286
Group 7	Portofino	Sail	41.45	41.20	784,606
Group 8	Sailboat	Bridge	41.44	41.62	780,946



(a) Cover image 1

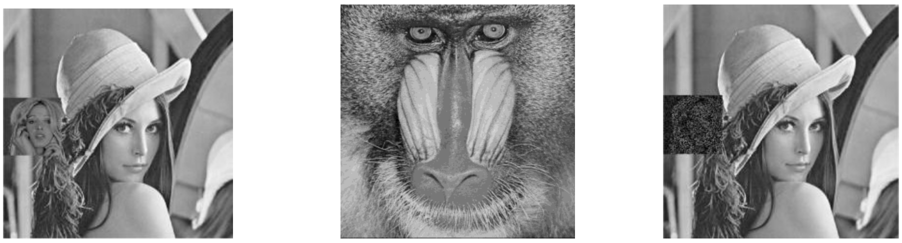
(b) Cover image 2

(c) Shadow 1 (PSNR=42.54 dB)

(d) Shadow 2 (PSNR=41.73 dB)

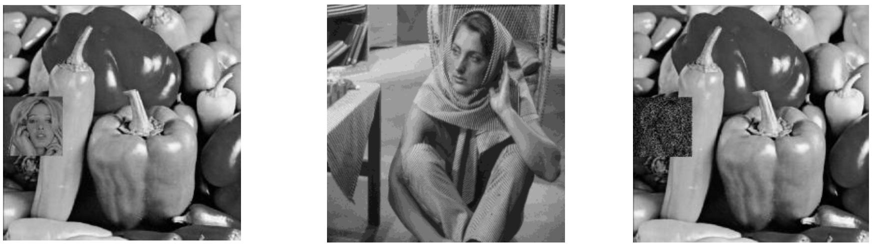
(e) Recovered secret image

Figure 13. Experimental results 2.



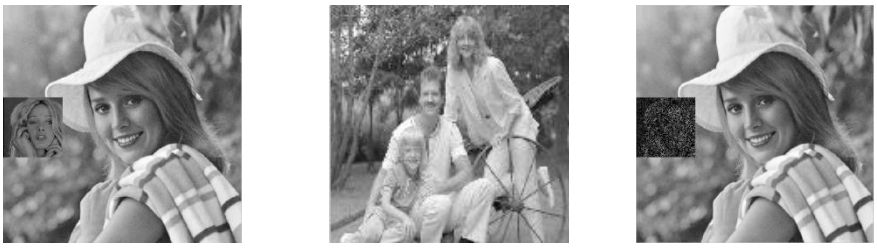
(a) Tampered shadow 1 (b) Real shadow 2 (c) Detection result

Figure 14. Cheating detection result 1.



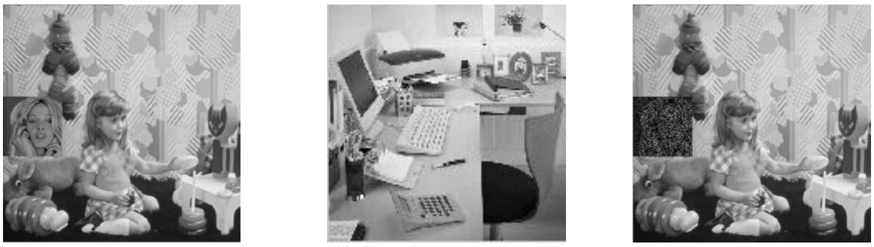
(a) Tampered shadow 1 (b) Real shadow 2 (c) Detection result

Figure 15. Cheating detection result 2.



(a) Tampered shadow 1 (b) Real shadow 2 (c) Detection result

Figure 16. Cheating detection result 3.



(a) Tampered shadow 1 (b) Real shadow 2 (c) Detection result

Figure 17. Cheating detection result 4.

We can use detection ratio (DR) for the tampered region to estimate the possibility of a successful cheating. DR is defined below:

$$DR = NTP/D \quad N, \tag{6}$$

where *NTP* is the total number of the tampered pixels and *NTPD* is the number of the tampered pixels that are detected. Table 2 lists DR values for different tampered shadows by the proposed scheme, which implies that it is very difficult for the cheater to deceive the honest participant because the probability of evading the cheating detection process is only about 50%.

In addition, we have to emphasize the fact that the proposed scheme is the first magic matrix-based SIS scheme that can realize cheating detection through the turtle shell matrix. As a consequence, we just show the experimental outcome of the proposed scheme since there can be no universal criteria to compare the proposed scheme and other existing magic matrix-based SIS schemes.

Table 2. DR values for different tampered shadows.

Tampered shadow	*DR*
Baboon	0.51
Peppers	0.53
Boat	0.51
Washsat	0.50
Elaine	0.50
Girl	0.50
Portofino	0.51
Sailboat	0.50

5. Conclusions

Recently, diverse magic matrices, such as EMD matrix and Sudoku matrix, are widely employed in secret image sharing (SIS). Unfortunately, these SIS schemes cannot detect cheaters. In this paper, to the best of our knowledge, the magic matrix-based SIS scheme is the first proposed method which can realize cheating detection. The characteristics of the proposed scheme include: (1) a turtle shell matrix is used to guide the secret sharing; (2) the secret message is shared into two different meaningful shadow images; (3) the secret message can be recovered losslessly by the cooperation of both shadow images; and (4) the cheating detection process is very easy to implement. Experimental results show that the proposed scheme can achieve great visual quality while maintaining high embedding capacity. By the cooperation of both shadow images based on the turtle shell matrix, the honest participant can detect whether the other participant is a cheater.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 61872436.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. X. Cao, L. Du and X. Wei, et al., High capacity reversible data hiding in encrypted images by patch-level sparse representation, *IEEE Trans. Cybern.*, **46** (2016), 1132–1143.
2. A. Shamir, How to share a secret, *Commun. ACM*, **22** (1979), 612–613.
3. G. R. Blakley, Safeguarding cryptographic keys, *Proceedings of American Federation of Information Processing Societies National Computer Conference*, New York, USA, **48** (1979), 313–317.
4. M. Naor and A. Shamir, Visual cryptography, *Proceedings of Advances in Cryptology-Eurocrypt'94*, Perugia, Italy, 1995, 1–12.
5. S. J. Shyu, Efficient visual secret sharing scheme for color images, *Pattern Recognit.*, **39** (2006), 866–880.
6. C. N. Yang and T. S. Chen, Reduce shadow size in aspect ratio invariant visual secret sharing schemes using a square block-wise operation, *Pattern Recognit.*, **39** (2006), 1300–1314.
7. C. N. Yang, K. H. Yu and R. Lukac, User-friendly image sharing using polynomials with different primes, *Int. J. Imaging Syst. Technol.*, **17** (2007), 40–47.
8. C. C. Thien and J. C. Lin, An image-sharing method with user-friendly shadow images, *IEEE Trans. Circuits Syst.*, **13** (2003), 1161–1169.
9. C. C. Chang, C.Y. Lin and C. S. Tseng, Secret image hiding and sharing based on the (*t*, *n*)-threshold, *Fundam. Inform.*, **76** (2007), 399–411.
10. X. T. Wu and W. Sun, Generalized random grid and its applications in visual cryptography, *IEEE Trans. Inf. Forensics Secur.*, **8** (2013), 1541–1553.
11. C. C. Chang, Y. P. Hsieh and C. H. Lin, Sharing secrets in stego images with authentication, *Pattern Recognit.*, **41**(2008), 3130–3137.
12. P. Y. Lin, J. S. Lee and C. C. Chang, Distortion-free secret image sharing mechanism using modulus operator, *Pattern Recognit.*, **42** (2009), 886–895.
13. C. C. Chang, T. D. Kieu and Y. C. Chou, Reversible data hiding scheme using two steganographic images, *Proceedings of IEEE Region 10 International Conference (TENCON)*, Taipei, Taiwan, 2007, 1–4.
14. C. Qin, C. C. Chang and T. J. Hsu, Reversible data hiding scheme based on exploiting modification direction with two steganographic images, *Multimed. Tools Appl.*, **74** (2015), 5861–5872.
15. N. T. Huynh, K. Bharanitharan and C. C. Chang, Quadri-directional searching algorithm for secret image sharing using meaningful shadows, *J. Vis. Commun. Image Represent.*, **28** (2015), 105–112.
16. C. C. Chang, Y. Liu and T. S. Nguyen, A novel turtle shell based scheme for data hiding, *The Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2014)*, Kitakyushu, Japan, 2014, 89–93.