


## Article

# Using Two Meaningful Shadows to Share Secret Messages with Reversibility

Lin Li <sup>1,2</sup>, Chia-Chen Lin <sup>3,\*</sup>  and Chin-Chen Chang <sup>2,\*</sup><sup>1</sup> Computer Engineering College, Ji Mei University, Xiamen 361021, China; llinfcu@gmail.com<sup>2</sup> Department of Information Engineering and Computer Science, Feng Chia University, 100 Wenhwa Road, Seatwen, Taichung 40724, Taiwan<sup>3</sup> Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan\* Correspondence: mhlin3@pu.edu.tw (C.-C.L.); alan3c@gmail.com (C.-C.C.);  
Tel.: +86-(04)2451-7250 (ext. 3791) (C.-C.C.)

Received: 2 December 2018; Accepted: 8 January 2019; Published: 11 January 2019



**Abstract:** A subtopic of visual secret sharing (VSS) is information hiding-based VSS (IH-VSS), which embeds secret messages into images using an information hiding technique. In the IH-VSS scheme, stego-images are divided into shadows under the guidance and constraint of some predetermined approaches. In order to achieve the purpose of security and reliability, the hidden information cannot be recovered unless a certain amount or all of the credible shadows work together. In this paper, we propose a (2, 2) IH-VSS scheme with reversibility and friendliness. In the shadow generation phase, two meaningful shadow images are produced and then distributed. In the extraction and restoration phase, the hidden secret information and cover image, respectively, can be reconstructed credibly and correctly. No complex computation of shadow generation is involved, but high security is achieved. Moreover, a satisfying peak-signal-to-noise ratio (PSNR) is obtained with the high embedding capacity of 1.59 bpp in a very simple and effective way.

**Keywords:** IH-VSS; reference matrix; reversibility; friendliness

## 1. Introduction

Nowadays, in the open network environment, effectively managing and protecting digital multimedia content, as well as preventing increasingly serious computer crimes and information from being illegally leaked, deleted, and modified, have become a research hotspot in the field of information security. People first turn to traditional cryptography [1], which ensures the message cannot be overheard or destroyed by using secret keys with different algorithms [2–4]; consequently, the eavesdropper cannot see or understand the secret information. Nevertheless, at the same time, distinctive and meaningless secret key-based encryption results may unintentionally attract intruders' attention. Furthermore, encryption algorithms are always very complicated, and the processes of encryption and decryption requires huge space and time complexity. To solve these problems, people have been looking for new solutions as an effective complement to the traditional cryptographic system. In such efforts, secret sharing and information hiding are significantly concerned.

Secret sharing serves to enhance security features and prevent authority fraud by avoiding the loss, modification, or destruction of important hidden information. The first secret sharing scheme is the threshold secret sharing scheme based on the Lagrange interpolating polynomial and finite geometry, which was proposed in 1979 by Shamir [5] and Blakley [6], independently. In their schemes, a dealer partitions a secret into  $n$  shadows, which are distributed to  $n$  participants. It takes at least  $k$  ( $k \leq n$ ) participants to work together to recover the secret, and less than  $k$  will not release any

valuable information. Thus, the advantages of a secret sharing scheme are as follows: (1) Preventing the excessive concentration of power to lead to abuse due to shared features; (2) ensuring the security and integrity of the secret key as enough shares must be provided to restore the original image and secret; and (3) increasing the reliability of the transmission process of secret information without increasing the risk. With the advent of secret sharing and in-depth research, secret sharing schemes based on different access structures have been increasingly proposed and applied in the last decade.

In 1994, at the European cryptography conference, a new branch of the secret sharing field, the visual secret sharing (VSS) scheme—or visual encryption (VC), as some people call it—was put forward by Naor and Shamir [7]. VSS uses several random-like images called shadows that do not make any sense, instead of the original image, to transfer information over the Internet. By superposing the shadows, secret images can be easily recovered because VSS depends only on the human visual system (HVS) and has no need to apply the complex calculations of the traditional cryptographic system. In fact, it has no need for any cryptology knowledge or computer help in restoring a secret. Hence, VSS has a wide range of practical applications and has been extensively studied in the literature (e.g., References [8–14]). However, this method faces the pixel expansion problem and suffers from the management problem. As we know, a VSS with less pixel expansion will have a smaller probability of making mistakes in reconstructing the secret image. Another problem that should not be ignored is that, during the transmission of shares, the noise-like shares increase transmission risk. Thus, it is important to find a method for eliminating pixel expansion and reducing the transmission risk in VSS schemes, especially for medical, military, or artistic images.

In order to solve the problems mentioned thus far and achieve higher security, secret sharing technology, in combination with information hiding (i.e., information hiding based VSS, or IH-VSS), is valued and has become a hot research topic in recent years. The purpose of information hiding is to cover up the secret using technical means. If human eyes cannot see the existence of the transmission of information from the surface, the information can be forwarded and the detection by third-party recipients can be avoided. Because information hiding is less likely to attract the attention of attackers or regulators, the hidden secret is less likely to be leaked. Information hiding technology consists mainly of two parts: An information-embedding algorithm and a hidden information detection/extraction algorithm (detector). Secure parameters may be required to detect/recover the hidden secret information from a cover carrier. Under the premise that secret parameters are unknown, it is difficult for a third party to get or delete from the carrier—or even find the secret information.

Through the combination of information hiding and visual secret sharing (IH-VSS), the defects of secret sharing and/or drawbacks of information hiding will be significantly weakened, thereby making covert communication more secure. Hence, many studies related to IH-VSS scheme using a meaningful cover image have been conducted, such as [15–20]. However, one common problem of these studies is that they could not restore the shadow image to the cover image. In 2010, a reversible image sharing scheme based on a Sudoku matrix to preserve the fidelity of valuable host images was proposed [21] to successfully camouflage the shadows into the cover image with satisfactory quality, accompanied with distortion limited within a range of [0, 3]. Shortly after [21] was published, an updated version of it was put forward that not only enhanced visual quality, but also had a larger capacity for embedded secret data [22]. Subsequently, a low computational complexity Quadri-Directional Searching Algorithm (QDSA) for secret image sharing was proposed [23], thereby enabling the modification of the original cover pixel values to be limited within a small range according to the value of  $\omega$ . Moreover, the shadow images can achieve excellent visual quality, as demonstrated by [24], followed by a (2, 2)-threshold reversible IH-VSS scheme that achieve high quality and have authentication ability by skillfully employing a turtle shell matrix [25]. In addition to the above papers, some scholars have also designed different secret sharing schemes that feature the enhanced security of hidden secret messages. For example, in 2017, Cheng et al. [26] used meaningful image shadows that are based on gray code to offer fairly high security. In [27], He et al. proposed a high quality and high security image sharing.

Recently, an adaptive method with a greatly enhanced embedding capacity under less execution time was proposed; it is obviously superior to many current algorithms in performance efficiency [28].

The algorithms described thus far are all related to IH-VSS and have distinctive approaches that differ from one another, yet none of them are exempt from considering the trade-off between the embedding capacity and image quality of the generated shadows. In this paper, with no time-consuming mathematical operations involved, we proposed a 2-2 threshold IH-VSS scheme using a meaningful image as the cover image through which the avoidance of the pixel expansion problem caused by the noisy-like shadows of the traditional Shamir's algorithm can be realized. In the algorithm, we first append part of the information from the cover image to the end of the secret and hide the secret using Kim et al.'s  $M$  matrix. Next, pixel pairs of the cover image with partial modification and stego-image are switched according to the sequence of  $\{0, 1\}$  generated by a random number generator. During the extraction and restoration phase, when the two participants collect the valid shadows, they can calculate the random number  $Num$  according to the pre-determined algorithm and then generate the corresponding  $\{0, 1\}$  string to recover the hidden secret information and further obtain the intact cover image. In this way, the embedding rate of 1.59 bpp (bits per pixel) along with PSNR of 54.34 dB are realized. Moreover, a good balance between the high embedding capacity and PSNR is achieved. Furthermore, if one participant provides a fake shadow or maliciously wants to obtain information about another participant, the  $Num$ , which is key information for pixel switching between the two shadows, cannot be derived correctly. In other words, the security of the hidden secret is guaranteed. Experimental results show that, by increasing the embedded payload of secret information, the PSNR value remains above 50 dB, which is quite high; furthermore, the small distortion between shadow1 and shadow2 and the cover image adds a lot of imperceptibility and security to our proposed IH-VSS scheme. It is also noted that our proposed IH-VSS scheme is completely reversible and friendly, with highly execution efficiency, which makes the application of our proposed scheme more extensive.

The rest of this paper is organized as follows. Section 2 discusses Shamir's secret sharing system and Kim et al.'s EMD-2 scheme. The IH-VSS scheme we propose is introduced in Section 3, followed by the experimental results and performance analysis in Section 4. Finally, the conclusions are presented in Section 5.

## 2. Related Works

To give readers enough background knowledge for our proposed scheme, Section 2 briefly reviews Shamir's secret sharing [2], and Kim et al.'s EMD-2 scheme [29].

### 2.1. Shamir's Secret Sharing

Shamir [5] and Blakley [6] independently proposed the first secret sharing scheme in 1979, focusing on the concept of the  $(k, n)$  threshold secret sharing scheme. Some important notions were defined in their schemes. For example, the image held by a participant is called a secret share or a sub-secret and shadow. Secret sharing schemes are usually established by trusted distributors who calculate all the shares and distribute them over secure channels to participants. The threshold  $k$  has the important meaning that the possible values for secret  $S$  seem as likely as with no knowledge when the number of shares does not reach the lower limit  $k$ . In other words, the secret  $S$  cannot be reconstructed with fewer than  $k$  pieces. If  $k = n$ , then every shadow derived from the original secret  $S$  is required to reconstruct the secret.

In Shamir's  $(k, n)$  [5] (where  $1 \leq k \leq n$ ,  $k$  and  $n$  are integer) threshold secret sharing scheme, the secret data can be converted to (or can be made) a number, which we call  $I$ . The domain of secrets and shares is the elements of a finite field  $F_p$ , where  $p$  is a prime number ( $p > I$ ). To split  $I$  into  $n$  shares, the dealer determines and constructs a  $(k - 1)$ -degree polynomial as

$$f(x) = (a_0 + a_1x + \dots + a_{k-1}x^{k-1}) \bmod p, \quad (1)$$

then uses it through the following steps:

- Step 1. Make  $a_0 = I$ .
- Step 2. Choose  $(k - 1)$  integer values  $a_1, a_2, \dots, a_{k-1}$  randomly within  $[0, p - 1]$ .
- Step 3. Select freely for the  $i$ th ( $1 \leq i \leq n$ ) share a value of  $x_i$ , and all  $x_i$  must be distinct from one another.
- Step 4. Calculate a corresponding value of  $f(x_i)$  for each chosen  $x_i$ , using Equation (1):

$$y_i = f(x_i). \quad (2)$$

- Step 5. Sent the corresponding  $(x_i, y_i)$ —the generated share  $SH_i$ —to the  $i$ th participant.
- Step 6. Perform Step 3 to Step 5 until each of the  $n$  participants gets the corresponding share  $SH_j$  ( $1 \leq j \leq n$ ).

The following figure is the flow chart of Shamir's secret sharing scheme (Figure 1).

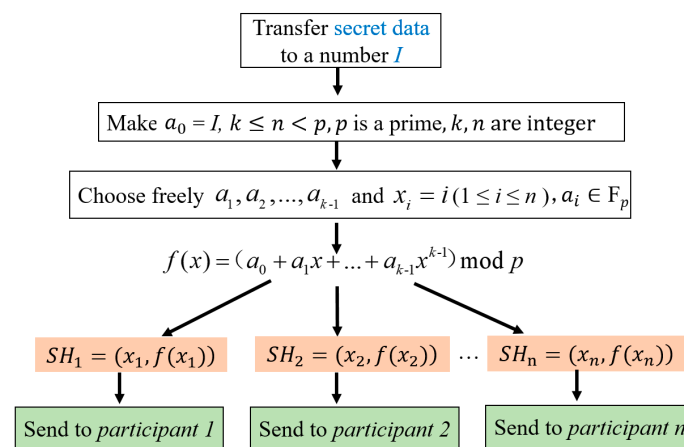


Figure 1. Flowchart of the share generation process of Shamir's scheme.

Thus, the secret share formation and distribution phase is completed. When necessary, the following steps can be performed to extract secret information.

- Step 1. Collect at least  $k$  ( $k \leq n$ ) secret shares  $SH_j$  ( $1 \leq j \leq k$ ) from the  $n$  participants to form the following equations:

$$\begin{aligned}
 f(x_1) &= (a_0 + a_1x_1 + \dots + a_{k-1}x_1^{k-1}) \bmod p, \\
 f(x_2) &= (a_0 + a_1x_2 + \dots + a_{k-1}x_2^{k-1}) \bmod p, \\
 &\vdots \\
 f(x_k) &= (a_0 + a_1x_k + \dots + a_{k-1}x_k^{k-1}) \bmod p.
 \end{aligned} \quad (3)$$

where  $x_i$  and  $f(x_i)$  are  $2k$  known values from the  $k$  secret shares.

- Step 2. Use the Lagrange polynomial interpolation method to solve the  $k$  unknowns values  $a_1, a_2, \dots, a_{k-1}$  and  $a_0$  in the  $k$  equations in Equation (3) and regenerate the  $(k - 1)$ -degree polynomial function  $f(x)$  given by Equation (1) to be:

$$\begin{aligned}
 f(x) &= f(x_1) \frac{(x-x_2)(x-x_3)\dots(x-x_k)}{(x_1-x_2)(x_1-x_3)\dots(x_1-x_k)} \\
 &+ f(x_2) \frac{(x-x_1)(x-x_3)\dots(x-x_k)}{(x_2-x_1)(x_2-x_3)\dots(x_2-x_k)} \\
 &+ \dots + f(x_k) \frac{(x-x_1)(x-x_2)\dots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\dots(x_k-x_{k-1})}.
 \end{aligned} \quad (4)$$

Step 3. Figure out the solution for the secret value  $a_0$  according to Equation (4)

$$a_0 = f(0) = (-1)^{k-1} \left[ f(x_1) \frac{x_2 x_3 \dots x_k}{(x_1 - x_2)(x_1 - x_3) \dots (x_1 - x_k)} + f(x_2) \frac{x_1 x_3 \dots x_k}{(x_2 - x_1)(x_2 - x_3) \dots (x_2 - x_k)} + \dots + f(x_k) \frac{x_1 x_2 \dots x_{k-1}}{(x_k - x_1)(x_k - x_2) \dots (x_k - x_{k-1})} \right]. \quad (5)$$

Hence, the secret message  $I = a_0$  is revealed. According to Shamir's  $(k, n)$  [5], if fewer than  $k$  shares are collected, the  $k$  unknown coefficients cannot be figured out and the desired secret data  $a_0$  cannot be extracted.

Example: Assume that the threshold is  $k = 3$ , we will generate a function  $f(x) = a_0 + a_1x + a_2x^2$  which  $f(0) = I$ , where  $I = 1234$  is the secret number derived from secret data. Therefore,  $a_0 = I = 1234$ . Then at random we obtain  $a_1 = 166$ ,  $a_2 = 94$ . Our polynomial which is used to produce secret shares is therefore:  $f(x) = 1234 + 166x + 94x^2$ . In order to have 6 valuable information of the secret, we select freely for the  $i$ th ( $1 \leq i \leq n$ ) share a value of  $x$ , denoted as  $x_i$ , which is distinct from one another, and calculate 6 values on the function  $f(x)$ . Thus, we get 6 shares as  $SH_1 = (x_1, f(x_1)) = (1, 1494)$ ,  $SH_2 = (x_2, f(x_2)) = (2, 1942)$ ,  $SH_3 = (x_3, f(x_3)) = (3, 2578)$ ,  $SH_4 = (x_4, f(x_4)) = (4, 3402)$ ,  $SH_5 = (x_5, f(x_5)) = (5, 4414)$  and  $SH_6 = (x_6, f(x_6)) = (6, 5614)$ , which will be delivered to  $n = 6$  participants.

The correspondent reconstruction process of secret data is as follows: According to the threshold  $k = 3$ , in order to reconstruct the secret any 3 of  $SH_i$  (here,  $1 \leq i \leq 6$ ) will be enough. Let us consider  $(x_0, y_0) = (2, 1942)$ ,  $(x_1, y_1) = (4, 3402)$ ,  $(x_2, y_2) = (5, 4414)$ . We will compute Lagrange basic polynomials on the basis of Equation (4):

$$f(x) = 1942 \times \frac{(x-4)(x-5)}{(2-4)(2-5)} + 3402 \times \frac{(x-2)(x-5)}{(4-2)(4-5)} + 4414 \times \frac{(x-2)(x-4)}{(5-2)(5-4)} \\ = 1234 + 166x + 94x^2$$

Secret is the free coefficient, which means  $I = 1234$ .

Regarding the characteristics of our proposed IH-VSS, we use the  $(2, 2)$  threshold for example in this paper. It can be extended to an  $(n, n)$  threshold.

## 2.2. Kim et al.'s EMD-2 Scheme

In 2006, Zhang and Wang proposed the first exploiting modification direction (EMD) scheme [30] providing good stego-image quality with the PSNR of more than 52dB and a considerable embedding rate of  $R = (\log_2(2n + 1))/n$  by modifying at most one least-significant bit of  $n$  pixel value. In 2010, Kim et al. developed the novel information hiding method called EMD-2 [29], which improved EMD by allowing at most two pixels to be modified and using different directions of modification to represent different secret data.

In Kim et al.'s EMD-2 scheme, a pixel group is defined as  $P_n = (p_1, p_2, \dots, p_n)$ , where  $n \geq 2$ . Follow Kim et al.'s definition, we think that in the case of  $n = 1$ , the processing is carried out with a pixel as the unit, without forming a pixel group. Under such a condition, it can be reduced to the LSB (least significant bits) [31] information hiding algorithm; and we will not discuss it in detail for we have benefited from a lot of treatises which have made detailed exposition to this problem.  $P_n$  is an  $n$ -dimensional space that can be mapped to a value by function  $f(P_n)$ . The basic elements  $b_i$  where  $i \in [0, 8]$  is used as an input to the function  $f$  as a weighted sum with modulo  $(2\omega + 1)$ .

$$f(p_1, p_2, \dots, p_n) = \left[ \sum_{i=1}^n (p_i b_i) \right] \bmod (2\omega + 1), \quad (6)$$

where

$$B_n = [b_1, b_2, \dots, b_n] = \begin{cases} [1, 3] & n = 2, \\ [1, 2, 3, 11, 16, 21, \dots, 6 + 5(n - 3)] & n > 2. \end{cases} \quad (7)$$

and

$$\omega = \begin{cases} 4 & n = 2, \\ 8 + 5(n - 3) & n > 2. \end{cases} \quad (8)$$

Here  $d$  is the secret data to be hidden, and  $f$  is the value calculated by Equation (6) associated with the pixel group  $P_n$ . The subscript value  $v$  is calculated using Equation (9).

$$v = \begin{cases} d - f & \text{if } d \geq f, \\ (2\omega + 1) - |d - f| & \text{if } d < f \text{ and } n > 2, \\ 4 - |d - f| & \text{if } d < f \text{ and } n = 2. \end{cases} \quad (9)$$

The vector of the treated pixel value  $P'_n$  can be acquired using Equation (10).

$$P'_n = P_n + C_v. \quad (10)$$

where  $C_v$  is the exponential vector when the number  $v$  is associated with the basis vector  $B_n$  which has been defined in Equation (7).

Actually, when  $n = 2$ , the best choice of the basis vector is  $B_2 = [1, 3]$  (see Equation (7)); meanwhile,  $\omega = 4$  (see Equation (8)). We use these parameters in our proposed scheme because we use (2, 2) threshold as an example in this paper. This basis vector can produce numbers from 0 to 8. Table 1 summarizes the details of the nine numbers and the corresponding coefficient vectors named  $C_i$ , where  $i \in [0, 8]$ . Combine the lowest three digits  $C_0$  of 0 is  $[0, 0]$ .  $C_1 = [1, 0]$ .  $C_2 = [-1, 1]$ , etc. Note that the number 0 can be generated by the basis vector as  $0 = C_0 \cdot B_2^T = (0 \times 1 + 0 \times 3) \bmod 9$ . In addition,  $C_1$  is  $[1, 0]$ ; thus, the number 1 is generated by  $1 = (1 \times 1 + 0 \times 3) \bmod 9$ . All nine numbers from 0 to 8 can be generated using Equation (6) by the linear combination of the basic elements with their associated coefficients.

**Table 1.** When  $n = 2$ , the basic numbers produced by the basis vector  $B_2 = [1, 3]$ .

Basic Number	$C_i$
0	$[0, 0]$
1	$[1, 0]$
2	$[-1, 1]$
3	$[0, 1]$
4	$[1, 1]$
5	$[-1, -1]$
6	$[0, -1]$
7	$[1, -1]$
8	$[-1, 0]$

In order to make it more intuitive and convenient to apply in our subsequent processes of the IH-VSS scheme, the calculation process we just mentioned can be projected as the generation of a matrix called  $M$ . In the matrix, under the function of mod 9, the difference value between the two adjacent elements in the same row of the reference matrix  $M$  is set to 1, and the difference value between the two adjacent elements in the same column is set to 3. An example of the reference matrix  $M$  with a size of  $256 \times 256$  is shown in Figure 2. The column value  $p_i$  and the row value  $p_{i+1}$  in  $M$  represent the grayscale values of the pixel pair  $(p_i, p_{i+1})$  in the cover image.

$$M(p_i, p_{i+1}) = f(p_i, p_{i+1}) = P_2 \cdot B_2 \bmod 9 = (p_i \times 1 + p_{i+1} \times 3) \bmod 9. \quad (11)$$



255	0	1	2	3	4	5	6	7	8	0	...	3
⋮	⋮					⋮					⋮	⋮
9	0	1	2	3	4	5	6	7	8	0	⋮	0
8	6	7	8	0	1	2	3	4	5	6		6
7	3	4	5	6	7	8	0	1	2	3		3
6	0	1	2	3	4	5	6	7	8	0		0
$p_{i+1}$ 5	6	7	8	0	1	2	3	4	5	6	...	6
4	3	4	5	6	7	8	0	1	2	3		3
3	0	1	2	3	4	5	6	7	8	0		0
2	6	7	8	0	1	2	3	4	5	6		6
1	3	4	5	6	7	8	0	1	2	3		3
0	0	1	2	3	4	5	6	7	8	0	...	0
	0	1	2	3	4	5	6	7	8	9	...	255
	$p_i$											

Figure 2. Kim et al.'s  $M$  matrix.

According to Equations (6)–(8), each pixel pair  $(p_i, p_{i+1})$  is located in the element function  $M(p_i, p_{i+1})$  in  $M$ . By using Equation (11), we can calculate all the values of the matrix.

Here, two examples are given to demonstrate how  $M$  matrix maps a pixel pair into a value. For pixel pair (3, 5), as illustrated in Figure 2, its derived value is 0 based on Equation (11)  $M(3, 5) = (3 \times 1 + 5 \times 3) \bmod 9 = 0$ . For pixel pair (123, 150), its derived value is 6 based on Equation (11)  $M(123, 150) = (123 \times 1 + 150 \times 3) \bmod 9 = 6$ .

### 3. Proposed Scheme

This section presents a (2, 2)-threshold IH-VSS scheme based on Kim et al.'s  $M$  matrix, which has the feature of high embedding capacity, without distortion and low computational complexity. In our scheme, three entities are involved: *dealer D* and two participants (i.e., *participant 1* and *participant 2*). Given a secret message and a meaningful cover image, two meaningful shadows are created and distributed to participants by *dealer D*. At the time of need, the two shadows are given by the designated legal participants, and their secret information can be extracted using our proposed extracting algorithm; at the same time, the cover image is restored. The concealment and sharing process will be introduced in Sections 3.2 and 3.3, respectively. The hidden secret extraction procedure will proceed when the involved participants gather all shadows together. Meanwhile, the cover image can be reconstructed without loss. The corresponding process is discussed in Section 3.4.

#### 3.1. Preliminary Phase

Here, a secret data  $S$ , which is (or can be transferred into) a binary string, will be concealed. An  $H \times W$  grayscale image, denoted as  $O = \{o_i \mid i = 1, 2, \dots, H \times W\}$  is used as the cover image, where  $o_i \in [0, 255]$ ,  $H$  is the height and  $W$  is the width of the image. Our proposed IH-VSS scheme applies a reversible data hiding scheme to embed secret data  $S$  along with the lowest three digits of the first four pixels in the original cover image into two meaningful grayscale images. Once the concealment phase is complete, each pixel pair in the stego-image carries one secret digit in the 9-base system; hence, the maximum hiding capacity using the proposed IH-VSS scheme is  $n = 415,498 (512 \times 512 \times \log_2 9 / 2)$  bits when the cover grayscale image has  $512 \times 512$  pixels and the embedding rate is  $1.59 = 0.5 \times \log_2 9$  bpp.

In the proposed (2, 2) IH-VSS scheme, the pre-process is listed as follows.

**Step 1:** Design a pixel value random diffusion function  $Rand3LSB(p_i)$ . Here  $3LSB$  means the three least significant bits of a pixel value, and  $Rand$  means that the  $3LSBs$  are given a value between '000' and '111' in a binary system with equal probability, as shown in Figure 3. This function makes the three LSBs of a pixel value  $p_i$  of a cover image slightly altered with randomness on a small scale. Such an arrangement can keep the pixel value still in the range of  $[0, 255]$ ,

and the underflow and overflow problems do not exist in our IH-VSS scheme. Thus, the least significant three bits of the binary number are randomly changed to a new value that differs from the original value in the range  $[0, 7]$ . As a result, the severe pixel distortion problem can also be avoided.

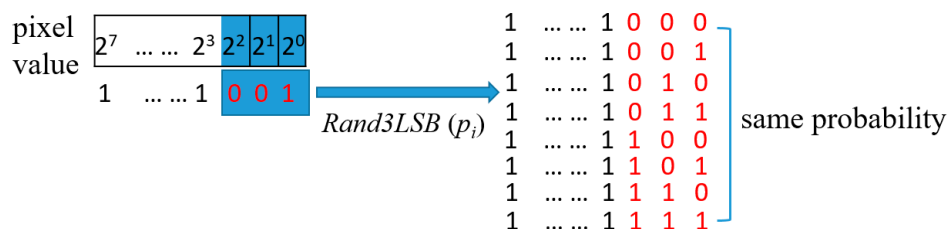


Figure 3.  $\text{Rand3LSB}(p_i)$  function.

**Step 2:** Process the first four pixels of the cover image  $O$  with the random diffusion function  $\text{Rand3LSB}(p_i)$ . After being handled, collect the four processed pixels and the rest of the original pixels of the cover image  $O$  and denote them as image  $O'$ . The first four pixels of image  $O'$  are called  $o'_1, o'_2, o'_3$  and  $o'_4$ . The relevant preprocessing is illustrated in Figure 4.

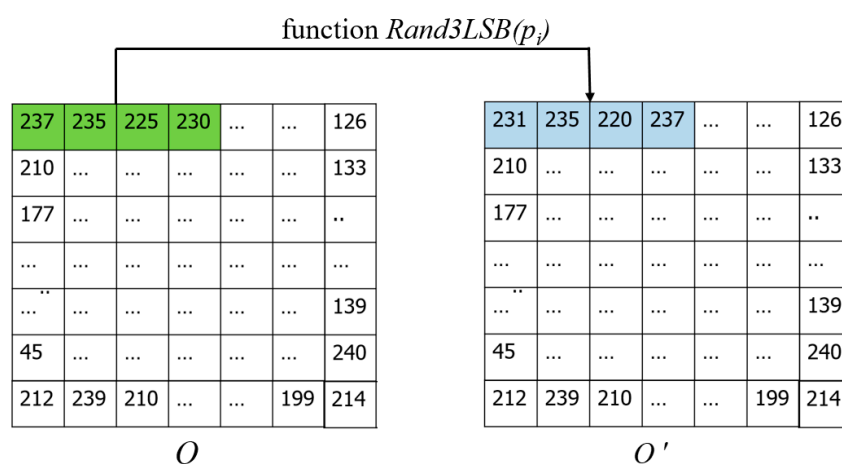


Figure 4. The preprocessing procedure of the cover image  $O$ .

**Step 3:** Consider the pixel values of a grayscale image, which range between  $[0, 255]$ , as a one-dimensional array.  $p_1$  is the first pixel value and  $p_2$  is the second pixel value. The subscript of  $p$  increases continuously until the maximum value is  $H \times W$ . Define pixel pair  $(p_i, p_{i+1})$  that is constituted by the grayscale values of two adjacent pixels  $p_i$  and  $p_{i+1}$ . For example, pixel  $p_i$  has a pixel grayscale value of 42 and pixel  $p_{i+1}$  has grayscale value of 41. The resulting pixel pair is  $(p_i, p_{i+1}) = (42, 41)$ , as shown in Figure 5.



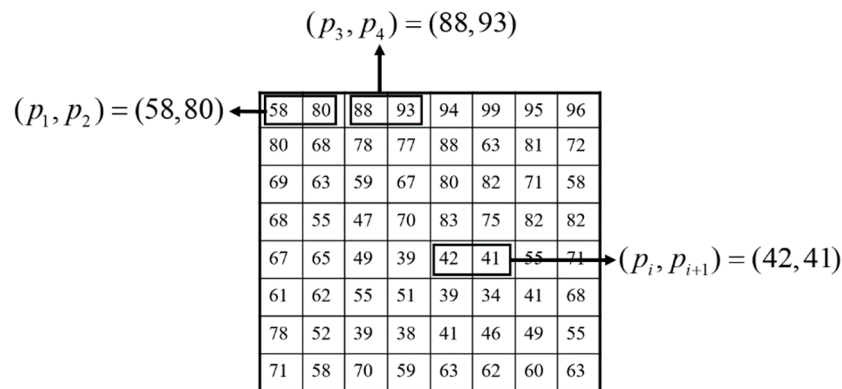
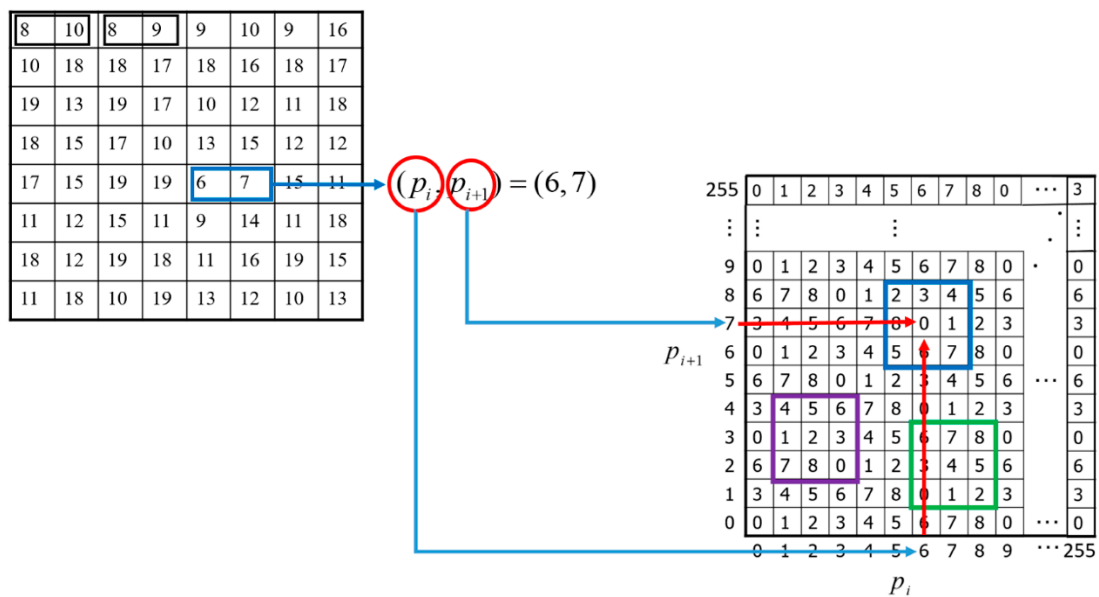


Figure 5. The definition of pixel pair.

**Step 4:** Define a matrix  $M$  with the size of  $256 \times 256$  according to Kim et al.'s EMD-2 algorithm. In the matrix  $M$ , any  $3 \times 3$  square, for example, the green, blue and purple square demonstrated in Figure 6, has 9 numbers that can be used to conceal a secret value ranging from 0 to 8, as shown in Table 2 according to Figure 6.  $(p'_i, p'_{i+1})$  is pixel pair of  $(p_i, p_{i+1})$  after being hidden. Since each pixel pair we processed is targeted at the pixel pairs with the same position in the cover image and stego-image, their subscripts are one-to-one correspondence.

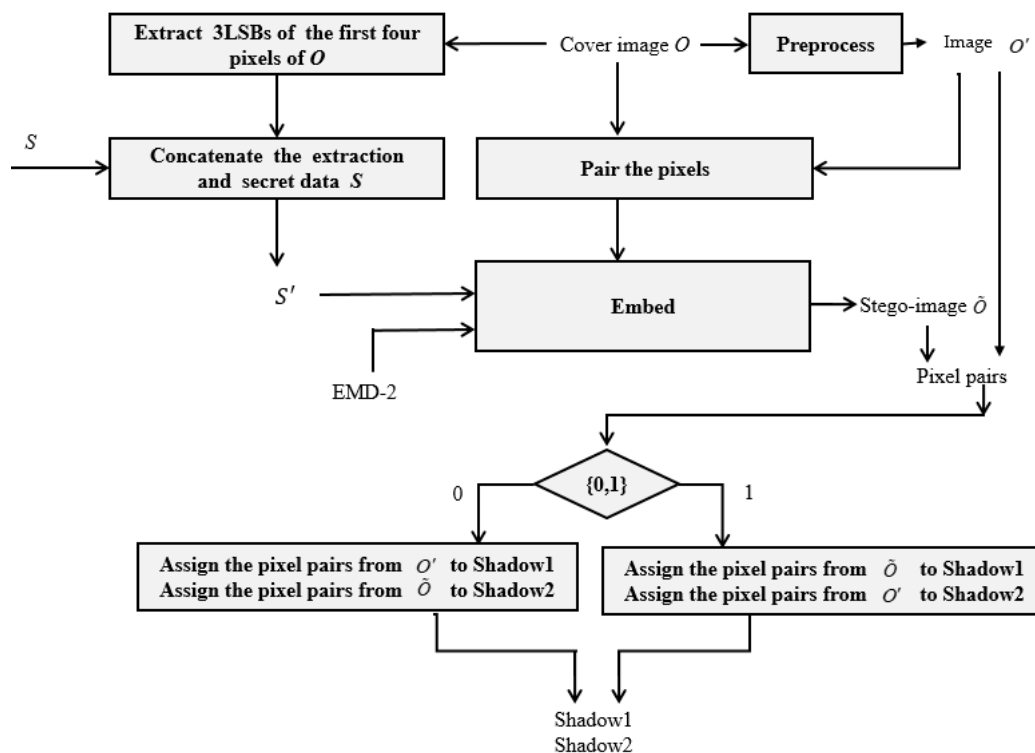
Figure 6. Matrix  $M$  and the  $3 \times 3$  square.

**Table 2.** Hidden data value and  $3 \times 3$  square of  $(p_i, p_{i+1})$ .

Hidden Data Value	$(p'_i, p'_{i+1})$ (Pixel Pair after Being Hidden)		
	$(p_i, p_{i+1}) = (6, 7)$	$(p_i, p_{i+1}) = (8, 3)$	$(p_i, p_{i+1}) = (1, 1)$
0	(6, 7)	(9, 3)	(0, 0)
1	(7, 7)	(7, 4)	(1, 0)
2	(5, 8)	(8, 4)	(2, 0)
3	(6, 8)	(9, 4)	(0, 1)
4	(7, 8)	(7, 2)	(1, 1)
5	(5, 6)	(8, 2)	(2, 1)
6	(6, 6)	(9, 2)	(0, 2)
7	(7, 6)	(7, 3)	(1, 7)
8	(5, 7)	(8, 3)	(2, 2)

### 3.2. Concealment Phase

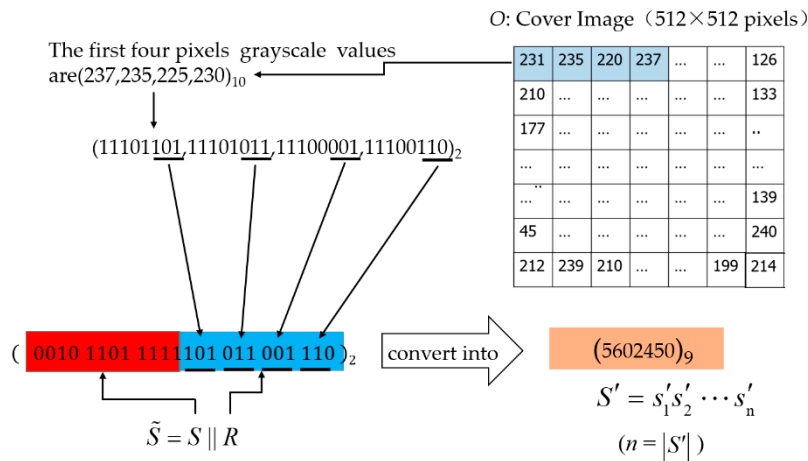
A secret data  $S$  and an  $H \times W$  grayscale cover image are given, then the concealment phase can be executed as the flowchart of Figure 7.

**Figure 7.** Flowchart of the embedding phase.

**Step 1:** Name the first four pixels of cover image  $O$  as  $o_1, o_2, o_3$ , and  $o_4$ .

**Step 2:** Combine the lowest three digits of  $o_1, o_2, o_3$ , and  $o_4$  to form  $R = (101\ 011\ 001\ 110)_2$ . Define the sign of operation  $' \parallel '$  as a concatenate function. Following the previous example, if  $S = 001011011111$ , then  $\tilde{S} = S \parallel R = (001011011111101011001110)_2$ , and the final number  $S'$  represented by a number in 9-base system is  $(5602450)_9$ . We specify that  $\tilde{S}$  is in the binary number,  $S'$  is the 9-base number corresponding to  $\tilde{S}$ , and actually, they have the same value. In addition, we express  $S'$  in this form  $S' = s'_1 s'_2 \dots s'_n$ , where  $n = |S'|$  is the length of the number string  $S'$ , and  $s'_k (k \in [1, n])$  is every digit in this number. This preprocessing procedure is illustrated in Figure 8.

- Step 3:** Conduct the function  $RNG(Num)$  to generate the seed of the  $randi()$  function, where  $RNG(Num)$  function controls random number generation so that the same string of  $\{0, 1\}$  will be generated by  $randi()$  on the secret distributor and receiver, which allows encrypt and decrypt participants to use the same  $\{0, 1\}$  string.
- Step 4:** Pair each of the remaining pixels in image  $O'$  with their adjacent pixels to form non-overlapping pixel pairs, such as  $(o'_i, o'_{i+1})$ . Thus, there will be  $\lfloor H \times W \rfloor / 2 - 2$  pixel pairs that can be used to embed secret information  $S'$ . As the cover image can be selected at will, the number of pixels of its height and width do not have to be even, so their product may be odd. Thus, we get the maximum number of the pixel pairs through floor function on  $H \times W$ .
- Step 5:** Embed secret information  $S'$  into each pixel pair with the concealment algorithm according to the following four sub-steps.
- Step 5.1:** Calculate the value  $m = M(o'_i, o'_{i+1})$  to obtain the value  $m$  by mapping  $o'_i$  and  $o'_{i+1}$  to matrix  $M$  defined in Figure 2 and Equation (11).
- Step 5.2:** Compare  $m$  and the  $s'_k$ . If they are equal, no change is required,  $(o'_i, o'_{i+1})$  in cover image will be copied to  $(\tilde{o}_i, \tilde{o}_{i+1})$  in stego-image. Otherwise, set  $m$  as the center  $(o'_i, o'_{i+1})$  and form a  $3 \times 3$  search area. In the  $3 \times 3$  square search area, find the corresponding  $\tilde{o}_i$  and  $\tilde{o}_{i+1}$  whose intersection is  $s'_k$  in the  $M$  matrix. Here, a new pixel pair  $(\tilde{o}_i, \tilde{o}_{i+1}) = M^{-1}(s'_k)$  of the stego-image is derived to carry the secret digit  $s'_k$ . The subscripts of  $(o'_i, o'_{i+1})$  and  $(\tilde{o}_i, \tilde{o}_{i+1})$  are one-to-one correspondence, as discussed in Step 4 of preliminary phase. Thus, a pixel pairs  $(o'_i, o'_{i+1})$  of  $O'$  conceals a base-9 digit  $s'_k$  by set values of pixel pairs  $(\tilde{o}_i, \tilde{o}_{i+1})$  of stego image  $\tilde{O}$ .
- Step 5.3:** Repeat the preceding steps until all pixel pairs of  $O'$  except the first four pixels have been processed. Hence, the stego-image  $\tilde{O}$  is obtained.



**Figure 8.** The preprocessing procedure of the secret information  $S'$

Example: To give a clear demonstration of our concealment phase, here, two pixel pair  $(o'_i, o'_{i+1})$  of  $O'$ , set as  $(5, 8)$  and  $(8, 3)$ , the secret digit  $s'_1 = 6$  and  $s'_2 = 7$  are used. Since  $(o'_5, o'_6) = (5, 8)$ , 5 is mapped to column  $x$  and 8 is mapped to row  $y$  of matrix  $M$ . In other words, a base-9 digit '2' can be found by using  $M(5, 8) = 2$ , which is demonstrated as the center of the red rectangle in Figure 9. But the base-9 digit '2' is not the same as  $s'_1 = 6$  which is the number we need to hide at present, therefore, a  $3 \times 3$  sized square is set as the search area around the center  $(5, 8)$ , and a new pair  $(6, 9) = M^{-1}(6)$  can be found because its intersection of column 6 and row 9 in matrix  $M$  is 6. Thus, the pixel pair  $(o'_5, o'_6) = (5, 8)$  in  $O'$  is changed to  $(\tilde{o}_5, \tilde{o}_6) = (6, 9)$  in  $\tilde{O}$  to carry secret digit  $s'_1 = 6$ . For the next pixel pair  $(8, 3)$ , we can get the center of blue square  $M(8, 3) = 8$  in Figure 9, which will be used to hide  $s'_2 = 7$ . Thus, we have to find 7 in the  $3 \times 3$  search area around the center  $M(8, 3) = 8$ . Then a new pair  $(\tilde{o}_7, \tilde{o}_8)$

$= (7, 3) = M^{-1}(7)$  can be found and set as the pixel pair to represent  $(o'_7, o'_8) = (8, 3)$  to carry secret digit  $s'_2 = 7$ . The detail is illustrated in Figure 9.

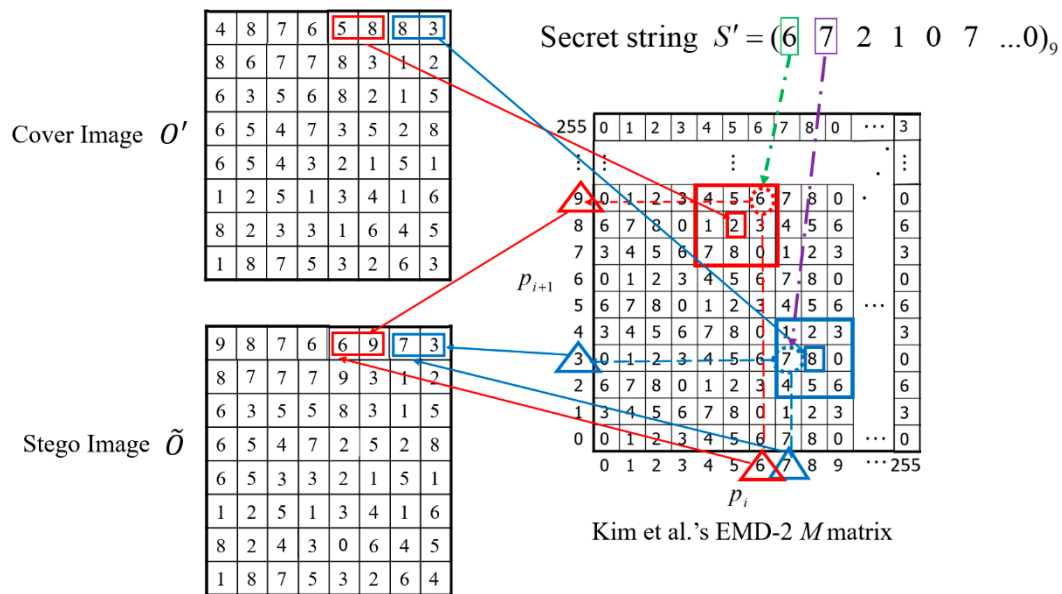


Figure 9. Example of the concealment phase with matrix  $M$ .

### 3.3. Shadow Generation Phase

Once stego-image  $\tilde{O}$  is constructed, it can cooperate with image  $O'$  to generate two meaningful shadows by conducting the following six steps.

- Step 1:** Combine the three LSBs of the first and second pixels of  $\tilde{O}$  to form a 6-bits of binary string. Then transform the 6-bits binary string into the decimal number system and name it  $Num1$ . Next, combine the three LSBs of the third and fourth pixels of  $\tilde{O}$  to form  $Num2$ . Finally, a decimal value named  $Num$ , which is the sum of  $Num1$  and  $Num2$ , is derived.
- Step 2:** Set  $RNum = randi([0, 1], \lfloor H \times W \rfloor / 2 - 1, 1)$  to get a binary string, such as  $(100010100110, 1001 \dots \dots 00101110)_2$ , of which the length is not greater than  $\lfloor H \times W \rfloor / 2 - 1$ .
- Step 3:** Generate the two shadows  $\tilde{O}_1$  and  $\tilde{O}_2$ . For shadow  $\tilde{O}_1$ , the first two pixels are the same as those of image  $\tilde{O}$ , but the following two pixels are derived from the  $Rand3LSB(pi)$  function with the third and the fourth pixels of image  $\tilde{O}$ . On the contrary, the third and the fourth pixels of shadow  $\tilde{O}_2$  are the same as those of image  $\tilde{O}$ . But the first two pixels of shadow  $\tilde{O}_2$  are derived from the  $Rand3LSB(pi)$  function with the first two pixels of image  $\tilde{O}$ .
- Step 4:** Pair the adjacent pixels of image  $\tilde{O}$  and image  $O'$ , except the first four pixels, to form non-overlapping pixel pairs. Use  $RNum$  as symbolic bits to decide to copy pixel pairs of  $O'$  and  $\tilde{O}$  to shadow1  $\tilde{O}_1$  or shadow2  $\tilde{O}_2$ . When the current bit of  $RNum$  is '0', the pixel pair of image  $O'$  is copied to the corresponding position of shadow1  $\tilde{O}_1$ , and the pixel pair of image  $\tilde{O}$  is copied to the corresponding position of shadow2  $\tilde{O}_2$ . When the symbolic bit of  $RNum$  is '1', the pixel pair of image  $O'$  is copied to the corresponding position of shadow2  $\tilde{O}_2$ , and the pixel pair of image  $\tilde{O}$  is replicated to the corresponding position of shadow1  $\tilde{O}_1$ . The detailed shadow generation procedure is shown in Figure 10. Step 4 will be repeated until all pixel pairs of images  $O'$  and  $\tilde{O}$  are processed.

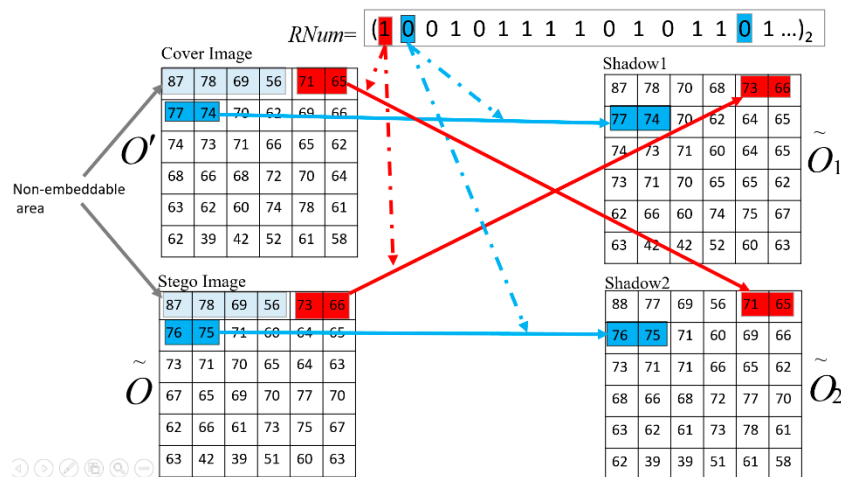


Figure 10. Example of shadow generation.

Example: We execute sequential processing of pixel pairs according to the ascending order of the subscript in the image and  $Rnum$ , thus  $(o'_5, o'_6) = (71, 65)$  in  $O'$ ,  $(\tilde{o}_5, \tilde{o}_6) = (73, 66)$  in  $\tilde{O}$ , '1' in  $Rnum$  (all colored in red) are the first two pixel pairs and the first random bit we use to do with. As the symbolic bit is '1', for  $\tilde{O}_1$ , we make  $(\tilde{o}_{15}, \tilde{o}_{16}) = (\tilde{o}_5, \tilde{o}_6) = (73, 66)$ ; for  $\tilde{O}_2$ ,  $(\tilde{o}_{25}, \tilde{o}_{26}) = (o'_5, o'_6) = (71, 65)$  (see the red arrow). And then, the next bit is '0' of  $Rnum$  (colored in blue), for  $\tilde{O}_1$ , we make  $(\tilde{o}_{17}, \tilde{o}_{18}) = (o'_7, o'_8) = (77, 74)$ ; for  $\tilde{O}_2$ ,  $(\tilde{o}_{27}, \tilde{o}_{28}) = (\tilde{o}_7, \tilde{o}_8) = (76, 65)$  (see the blue arrow). And next,  $(\tilde{o}_{19}, \tilde{o}_{20}) = (o'_{10}, o'_{10}) = (70, 62)$  for  $\tilde{O}_1$ ,  $(\tilde{o}_{29}, \tilde{o}_{30}) = (\tilde{o}_9, \tilde{o}_{10}) = (71, 60)$  for  $\tilde{O}_2$ , and so on.

**Step 5:** Set the LSB of the last pixel of shadow  $\tilde{O}_1$  to 0 and name the final result shadow1. Set the LSB of the last pixel of shadow  $\tilde{O}_2$  to 1 and name the final result shadow2. Send shadow1 to *participant 1* and shadow2 to *participant 2*.

### 3.4. Extraction and Restoration Phase

**Step 1.** Collect shadows  $\tilde{O}_1$  and  $\tilde{O}_2$  from participants 1 and 2. Here, we assume two participants are credible and the received shadows are also reliable. According to the LSB of the past pixel of received shadows, shadows  $\tilde{O}_1$  and  $\tilde{O}_2$  can be determined, respectively. From the first pixel pair of  $\tilde{O}_1$  and the second pixel pair of  $\tilde{O}_2$ , the random number seed  $Num$  can be derived. The  $RNG(Num)$  function can then be performed, and images  $O'$  and  $\tilde{O}$  can be restored according to the generated strings of 0 and 1. The restoration procedure is the inverse operation of Figure 10.

**Step 2.** Search in Kim et al.'s  $M$  matrix according to images  $O'$ ,  $\tilde{O}$ , and the function  $M^{-1}$ ; the embedded information in each pixel pair can then be derived sequentially.

**Step 3.** Copy the last 12 bits of the secret information  $S'$  back to image  $O'$  derived in **Step 1**. Thus, the original image  $O$  is restored exactly.

If the participant provides a forged image, the  $Num$  and the string of 0 and 1 based on the received forged image will be incorrect. Ultimately, the hidden secret message cannot be extracted.

## 4. Experimental Results and Analysis

Kim et al.'s data hiding strategy is employed in our proposed IH-VSS scheme to guarantee that two pixel pairs of shadows can carry a base-9 digit according to Equation (6). Hence, the embedding rate is  $1.59 = 0.5 \times \log_2 9$  bpp. In other words, a  $512 \times 512$ -pixels grayscale image can embed  $\lfloor \log_2 9 \times (H \times W) / 2 \rfloor = \lfloor \log_2 9 \times (512 \times 512) / 2 \rfloor = 415,498$  secret bits.

In addition to the hiding capacity, the security and visual quality of shadows are crucial criteria used to evaluate the performance of VSS. The following subsections discuss the security analysis and

visual quality. Comparisons among the four representative VSS schemes and our proposed scheme are demonstrated in Section 4.3 to confirm the performance of our proposed IH-VSS on both hiding capacity and visual quality.

#### 4.1. Security Analysis

This subsection examines the security capability of our proposed scheme. As previously discussed, the proposed IH-VSS scheme is a (2, 2)-threshold secret sharing scheme, in which the core idea is clearly defined that no single share can release any knowledge of the hidden secret message. However, it is possible that malicious users conduct a brute-force attack by randomly selecting pixel pairs from received shadows (i.e.,  $\tilde{O}_1, \tilde{O}_2$ ) to try to reconstruct stego-image  $\tilde{O}$ . To this end, the user must get the correct  $\{0, 1\}$  string. In a brute-force attack, the probability that a user will obtain the exact  $\{0, 1\}$  string is the reciprocal of  $2^{(M \times N)/2} = 2^{(512 \times 512)/2} = 2^{131072}$ , which is quite small and almost impossible to solve in real time. Thus, the security of the hidden secret message with our proposed IH-VSS scheme is guaranteed.

#### 4.2. Visual Quality

In general, the *PSNR* is used to measure the visual quality of the image after secret information is embedded into the cover image. *PSNR* is defined as:

$$PSNR = 10 \times \log\left(\frac{255^2}{MSE}\right). \quad (12)$$

where *MSE* is the mean square error between the cover image and the shadow. For an  $H \times W$  size grayscale cover image, *MSE* is defined as:

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (p_{ij} - \bar{p}_{ij})^2. \quad (13)$$

Here  $p_{ij}$  denotes the pixel value of the original image, and  $\bar{p}_{ij}$  denotes the shadow's pixel value at position  $(i, j)$ . Without a loss of generality, we believe that the difference between the cover image and shares is invisible to the naked eye when the *PSNR* value exceeds 30 dB, and we conclude that the visual quality of shadows reaches an ideal level [32].

The *PSNR* is often discussed with the embeddable secret capacity because they are the two major measurements of the performance of the VSS using the information hiding approach. The following figures show the contrasted *PSNR* value of images involved. Figure 11a shows the original Lena. The two shadows shown in Figure 11c,d have *PSNR* values of 55.98 dB and 55.91 dB, respectively, which are both meaningful images of size  $512 \times 512$ . As it is an accepted fact that when the value of *PSNR* is greater than 30, no difference can be detected directly through the human visual system. However, in the proposed scheme, *PSNR* value is often close to or more than 50 that makes the imperceptibility all the more salient. Furthermore, as is known to all, the pixel expansion is caused by shadows smaller than the cover image, which obviously does not exist in our scheme because shadows are the same size as the cover. Even when a large amount of the secret information is hidden, shares are visually no different from the original.





**Figure 11.** (a) Cover image; (b) Reconstructed cover image; (c) Shadow1 (PSNR = 55.98); (d) Shadow2 (PSNR = 55.91).

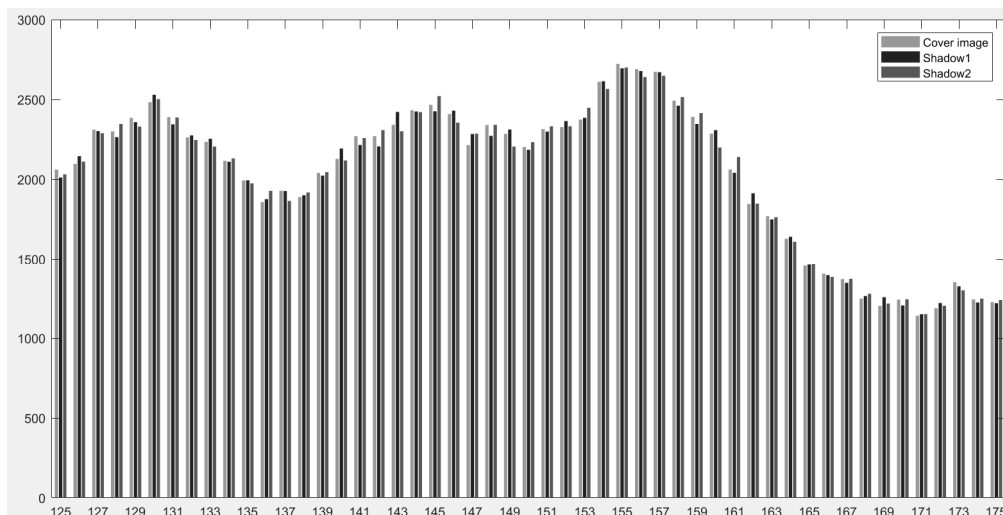
Another topic worth discussing is improvements to the produced shadow size. As we know, if the size of generated shadows is similar to or smaller than the size of the cover image, it is easier to manage and transmit the generated shadows. However, a tradeoff must be made between the size of the shadows and the size of the hidden secret. Smaller generated shadows may limit the size of hidden secret data even when relatively easier to manage and transmit. Thus, in the proposed IH-VSS scheme, the share size we define is the same as the cover image to keep the balance between the transmission efficiency and the size of the hidden secret message. Moreover, to enhance the security of the hidden secret message, the generated shadows are meaningful with the proposed IH-VSS scheme.

To enhance the security of the hidden secret message, the pixel pairs of the two generated intermediated shadows are switched with each other according to a binary random number string, except for the first four pixels, as shown in *Step 5* of shadow generation phase and Figure 10. To further prove that such an arrangement does not affect the visual quality of the final generated shadows, Table 3 shows the PSNRs of intermediate shadows and final shadows under different ECs, which increase from 13,000 to 415,485 bits. The column named image  $O'$  (before exchange) shows the PSNR of image  $O'$  before the exchange method, and it is as high as 84.99 dB when the EC is 415,485 bits because only four pixel values have changed, as we demonstrated in Section 3.3. The column named image  $\tilde{O}_1$  (after exchange) indicates the PSNR of the stego-image, which carries the secret using Kim et al.'s EMD-2 algorithm. The PSNR is larger than 52 dB when the EC is over 400,000 bits, which is significantly higher than 30 dB because the difference between the two pixel values before and after the hidden information is controlled between  $[0, 7]$  using Kim et al.'s EMD-2 algorithm. We also found that the PSNR of image  $\tilde{O}_2$  increased with the happening of the exchange because the exchange of the original image pixel pairs making image  $\tilde{O}_2$  contains not only the stego-pixel pairs, but also some original pixel pairs. Therefore, the PSNR of image  $\tilde{O}_2$  remains at a very high level, as shown in Table 3.

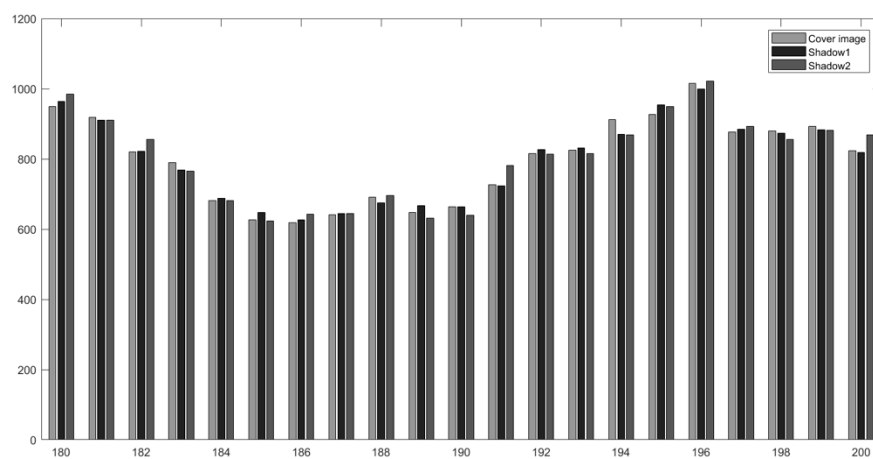
**Table 3.** PSNRs of two generated shadows with the increasing EC (EC: Embedding capacity; BE: Before Exchange; AE: After Exchange).

EC (Bits)	Image $O'$ (BE)	Image $\tilde{O}_1$ (AE)	Image $\tilde{O}$ (BE)	Image $\tilde{O}_2$ (AE)
	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
13,000	84.99	59.94	62.97	62.91
65,786	84.65	55.98	57.90	55.91
130,096	84.96	52.94	49.92	52.94
415,485	88.89	52.90	49.88	52.89

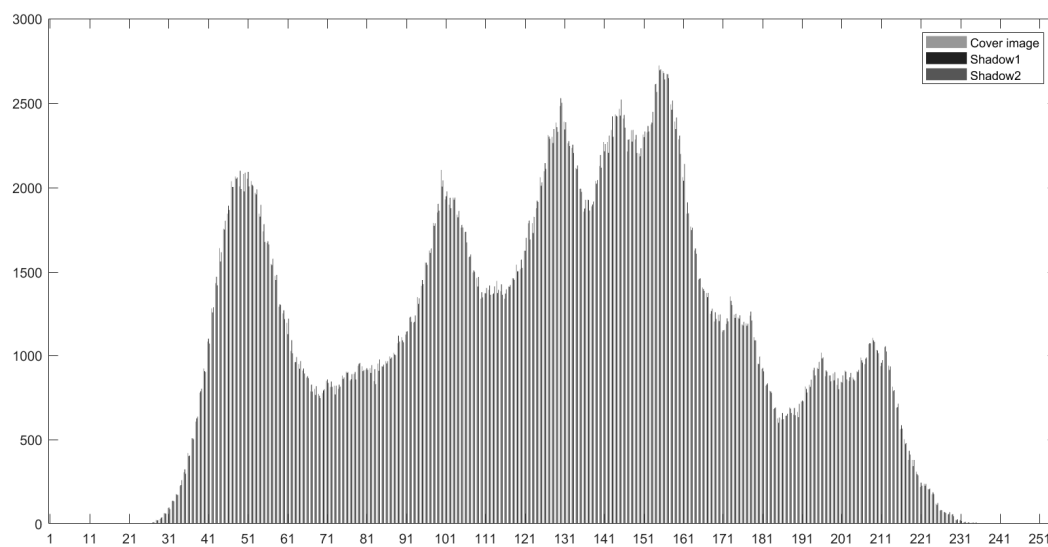
To demonstrate that the histograms of the original image “Lena” and its two shadows are similar, even when the size of our hidden secret message reaches the upper bound with our proposed IH-VSS, pixel values ranging from 125 to 175 is selected and demonstrated in Figure 12. The reason we selected such pixel range because it is the middle range of the pixel value [0, 255] for a given grayscale image, and most of the pixels of the image are in this range, which makes the detailed phenomenon of the frequencies of the cover image and shadows can be shown. In general, the stego-pixel values could be different from the cover image after shadows carried the secret message. However, Figure 12 shows that the difference of the frequencies is considerably small. This conclusion is further supported by Figures 13 and 14 which show the pixel values ranging from 180 to 200 and 0 to 255 (all possible pixel values of an image), respectively.



**Figure 12.** The histogram of the cover image and the two shadows (pixel values: 125–175).



**Figure 13.** The histogram of the cover image and the two shadows (pixel values: 180–200).



**Figure 14.** The histogram of the cover image and the two shadows (pixel values: 0–255).

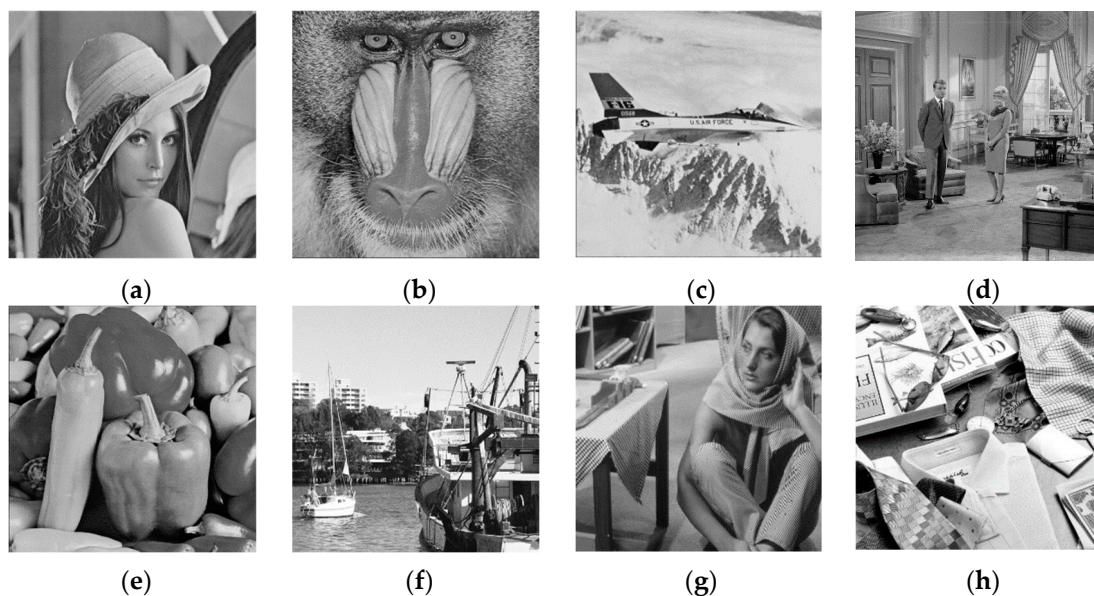
Table 4 also demonstrates the pixel values with the top eight highest number of pixels of the cover image and the two shadows. The data of first line, colored in blue, tells that the pixel value 155 has the maximum number of 2723 in the cover image, correspondingly, the number in shadow1 and shadow2 is 2711 and 2714. These three numbers have similar values. On the other hand, the pixel value 130 has the 5th largest number of pixels (colored in light brown), 2493, 2572 and 2519 are the numbers in the cover image, shadow1 and shadow2, respectively. These three numbers are very close too. The same thing happens to the top eight, as shown in Figure 14, in fact, to all the pixel values. Such results indicate that the proposed IH-VSS scheme is not only visually imperceptible, but also quantitatively negligible.

**Table 4.** The top eight pixel values with the highest number of pixels.

Top 8 (Pixel Value)	The Number of Pixels		
	Cover Image	Shadow1	Shadow2
1(155)	2723	2711	2714
2(156)	2699	2688	2642
3(157)	2683	2671	2659
4(154)	2611	2615	2589
5(130)	2493	2572	2519
6(158)	2483	2410	2521
7(145)	2471	2469	2481
8(143)	2433	2461	2407

#### 4.3. Comparisons with Four Existing Schemes

To evaluate the performance of the proposed VSS scheme, eight standard  $512 \times 512$  grayscale images, as shown in Figure 15, are used as test images. Here, PSNR comparisons between the proposed scheme and four state-of-the-art (2, 2) IH-VSS schemes are listed in Table 5.



**Figure 15.** Eight test images with size of  $512 \times 512$  pixels. (a) Lena; (b) Baboon; (c) Airplane; (d) Couple; (e) Peppers; (f) Harbor; (g) Barbara; (h) Fashion.

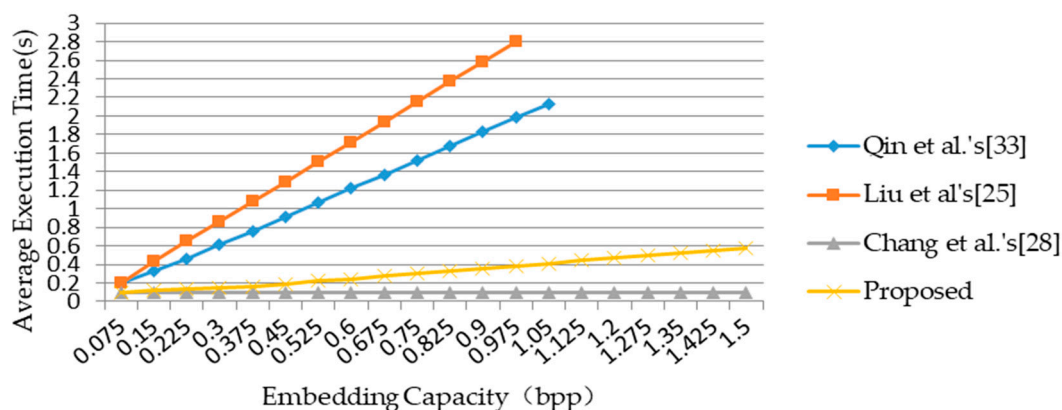
Consider the bpp value of Cheng et al.'s [26] is 0.5, and the maximum storage capacity of He et al.'s scheme [27] can only reach a value of 69398 bits, which is far less from our algorithm, we do not regard them as comparison data here. Therefore, in Table 5 only three existing schemes are used to compare with our proposed scheme. Among them, Chang et al.'s scheme [24] applies a control parameter  $\omega$  to determine the payload. Qin et al. [33] use the matrix proposed by Zhang and Wang to embed a secret. Liu et al.'s scheme [25] applies the turtle shell reference matrix during the shadow generation procedure. Chang et al.'s scheme ( $Th = 4$ ) [28] does not employ any reference matrix, but embeds the secret messages in the four LSB planes by combining LSB substitution and the exclusive or (XOR) operation to produce two shadow images. The comparisons show that the PSNR of the proposed scheme can be maintained at a high level-larger than 52 dB—when the embedding capacity reaches 415,485 bits.

**Table 5.** Comparison of PSNR among different VSS schemes when the EC is 415,484 bits (all the data except the proposed scheme come from [28]).

Cover Image	PSNR (dB)							
	Qin et al.'s Scheme [33]		Liu et al.'s Scheme [25]		Chang et al.'s Scheme [28]		The Proposed Scheme	
	Shadow1	Shadow2	Shadow1	Shadow2	Shadow1	Shadow2	Shadow1	Shadow2
(a) Lena	52.12	41.60	51.72	45.70	37.71	37.92	52.90	52.89
(b) Baboon	52.04	41.57	51.73	45.71	36.46	37.92	52.91	52.89
(c) Airplane	52.11	41.57	51.67	45.72	37.45	37.96	52.95	52.85
(d) Couple	51.83	41.55	51.75	45.68	37.52	37.92	52.92	52.89
(e) Peppers	52.09	41.59	51.75	45.70	37.59	37.93	52.87	52.89
(f) Harbour	51.42	41.48	51.73	45.73	36.46	37.92	52.89	52.90
(g) Barbara	52.11	41.57	51.72	45.71	36.87	37.92	52.89	52.90
(h) Fashion	46.84	40.66	51.94	45.92	36.87	37.92	52.97	52.94

Figure 15a–h uses the cover image to generate the corresponding shadow1 and shadow2 with different schemes. Again, it is proved that the proposed IH-VSS offers the highest PSNRs for two shadows no matter which cover image is adopted. Two reasons can explain this result. First, a finite change of pixel values is caused during the embedding procedure of Kim et al.'s EMD-2 algorithm. Second, the stego-pixel pairs of the two generated intermediate shadows are further switched by using the uniformly distributed pseudo-random string of  $\{0, 1\}$  generated by the function *randi()* at the end of the shadow generation phase. Thus, no matter which image is served as the cover image, PSNRs of the two shadows can be guaranteed to be greater than 50 and be very close.

Figure 16 shows the average execution time of four schemes. Compared with schemes of Qin et al. [33] and Liu et al. [25], our proposed scheme requires less execution time, but takes more time than scheme of Chang et al. [28]. Consider the average PSNR of generated shadows with our proposed scheme is 1.3 times more than that of Chang et al.'s scheme. We believe the execution time of our proposed scheme is acceptable.

**Figure 16.** Average execution time of schemes.

Combined with Table 5, it can be concluded that Chang et al.'s scheme sacrifices PSNR to achieve higher execution efficiency, while Qin et al.'s and Liu et al.'s work has the disadvantage of relatively smaller storage volume but higher PSNR. The proposed scheme, which makes a compromise between the amount of EC and PSNR, makes the PSNR close to or in most cases greater than 50 dB, while at the same time, the EC reaches more than 400,000 bits, and the execution efficiency is quite good.

## 5. Conclusions and Future Work

This paper presents a distortion-free friendly IH-VSS scheme. In the shadow generation phase, using Kim et al.'s  $M$  matrix and a random number, two meaningful shadows are produced and distributed to participants. In the extraction and restoration phase, the hidden secrecy information and cover image can be credibly and correctly extracted and reconstructed, respectively.

Experimental results confirm that no valuable secret information can be obtained with invalid shadows. The switching strategy conducted on pixel pairs between two intermediate shadows during the shadow generation phase makes the hidden and non-hidden pixel pairs of the two final shadows appear to be randomly distributed, making the difference between the two final shadows quite small.

The shadow images are friendly and meaningful, and the reversibility can be achieved. A relatively good balance can be achieved between the high embedding capacity and the satisfying PSNR of the meaningful shadows. The shadow generation phase is quite efficient; therefore, the proposed IH-VSS scheme is quite suitable for real-time applications. To enhance the practicality of our proposed scheme, our future work will focus on designing the hiding strategy so that the tampered areas of the shadows can be identified at the extraction and restoration phase.

**Author Contributions:** Methodology, C.-C.C.; Software, L.L.; Writing—original draft, L.L.; Writing—review & editing, C.-C.L.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Diffie, W.; Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
2. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
3. American National Standards Institute. *American National Standard Data Encryption Algorithm*; American National Standards Institute: Washington, DC, USA, 1981.
4. Daemen, J.; Rijmen, V. AES Proposal: Rijndael. 1999. Available online: [http://www.cs.miami.edu/home/burt/learning/Csc688.012/rijndael/rijndael\\_doc\\_V2.pdf](http://www.cs.miami.edu/home/burt/learning/Csc688.012/rijndael/rijndael_doc_V2.pdf) (accessed on 1 December 2018).
5. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
6. Blakley, G.R. Safeguarding cryptographic keys. Managing Requirements Knowledge. *Int. Workshop (AFIPS)* **1979**, 313–317. [CrossRef]
7. Naor, M.; Shamir, A. Visual cryptography. In *Advances in Cryptology—EUROCRYPT*; De Santis, A., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; Volume 950, pp. 1–12.
8. McEliece, R.J.; Sarwate, D.V. On sharing secrets and Reed-Solomon codes. *Commun. ACM* **1981**, *24*, 583–584. [CrossRef]
9. Karnin, E.; Greene, J.; Hellman, M. On secret sharing systems. *IEEE Trans. Inf. Theory* **1983**, *29*, 35–41. [CrossRef]
10. Brickell, E.F. Some ideal secret sharing schemes. In *Advances in Cryptology—EUROCRYPT*; Quisquater, J.-J., Vandewalle, J., Eds.; Springer: Berlin/Heidelberg, Germany, 1989; Volume 434, pp. 468–475.
11. Benaloh, J.; Leichter, J. Generalized secret sharing and monotone functions. In *Advances in Cryptology—CRYPTO*; Springer: Berlin/Heidelberg, Germany, 1988; Volume 403, pp. 27–35.
12. Ateniese, G.; Blundo, C.; Santis, A.; Stinson, D.R. Constructions and bounds for visual cryptography. *Autom. Lang. Program.* **1996**, *1099*, 416–428.
13. Lai, C.P.; Ding, C. Several generalizations of shamir’s secret sharing scheme. *Int. J. Found. Comput. Sci.* **2004**, *15*, 445–458. [CrossRef]
14. Ciamato, S.; De Prisco, R.; De Santis, A. Probabilistic visual cryptography schemes. *Comput. J.* **2006**, *49*, 97–107. [CrossRef]
15. Thien, C.C.; Lin, J.C. Secret image sharing. *Comput. Graph.* **2002**, *26*, 765–770. [CrossRef]
16. Tsai, C.S.; Chang, C.C.; Chen, T.S. Sharing multiple secrets in digital images. *J. Syst. Softw.* **2002**, *64*, 163–170. [CrossRef]
17. Lin, C.C.; Tsai, W.H. Secret image sharing with steganography and authentication. *J. Syst. Softw.* **2004**, *73*, 405–414. [CrossRef]
18. Wu, Y.S.; Thien, C.C.; Lin, J.C. Sharing and hiding secret images with size constraint. *Pattern Recognit.* **2004**, *37*, 1377–1385. [CrossRef]
19. Yang, C.N.; Chen, T.S.; Yu, K.H.; Wang, C.C. Improvements of image sharing with steganography and authentication. *J. Syst. Softw.* **2007**, *80*, 1070–1076. [CrossRef]



20. Chang, C.C.; Hsieh, Y.P.; Lin, C.H. Sharing secrets in stego images with authentication. *Pattern Recognit.* **2008**, *41*, 3130–3137. [[CrossRef](#)]
21. Chang, C.C.; Lin, P.Y.; Wang, Z.H.; Li, M.C. A Sudoku-based secret image sharing scheme with reversibility (invited paper). *J. Commun.* **2010**, *5*. [[CrossRef](#)]
22. Wang, Z.H.; Guo, C.; Chang, C.C. A novel (n, n) secret image sharing scheme based on Sudoku. *J. Electr. Sci.* **2013**, *11*, 44–50.
23. Huynh, N.T.; Bharanitharan, K.; Chang, C.C. Quadri-directional searching algorithm for secret image sharing using meaningful shadows. *J. Vis. Commun. Image Represent.* **2015**, *28*, 105–112. [[CrossRef](#)]
24. Chang, C.C.; Liu, Y.; Wu, H.L. Distortion-free secret image sharing method with two meaningful shadows. *IET Image Process* **2016**, *10*, 590–597. [[CrossRef](#)]
25. Liu, Y.; Chang, C.C. A turtle shell-based visual secret sharing scheme with reversibility and authentication. *Multimed. Tools Appl.* **2018**, *77*, 25295–25310. [[CrossRef](#)]
26. Cheng, T.F.; Chang, C.C.; Liu, L. Secret sharing: using meaningful image shadows based on Gray code. *Multimed. Tools Appl.* **2017**, *76*, 9337–9362. [[CrossRef](#)]
27. He, J.H.; Lan, W.Q.; Tang, S.H. A secure image sharing scheme with high quality stego-images based on steganography. *Multimed. Tools Appl.* **2017**, *76*, 7677–7698. [[CrossRef](#)]
28. Chang, C.C.; Liu, Y.J.; Chen, K.M. Real-time adaptive visual secret sharing with reversibility and high capacity. *J. Real. Time Image Process.* **2018**, 1–11. [[CrossRef](#)]
29. Kim, H.J.; Kim, C.; Choi, Y.; Wang, S.; Zhang, X. Improved modification direction methods. *Comput. Math. Appl.* **2010**, *60*, 319–325. [[CrossRef](#)]
30. Zhang, X.; Wang, S. Efficient steganographic embedding by exploiting modification direction. *IEEE Commun. Lett.* **2006**, *10*, 781–783. [[CrossRef](#)]
31. Chan, C.K.; Cheng, L.M. Hiding data in images by simple LSB substitution. *Pattern Recognit.* **2004**, *37*, 469–474. [[CrossRef](#)]
32. Chen, K.M.; Chang, C.C. High-capacity reversible data hiding in encrypted images based on extended Run-Length coding and block-based MSB plane rearrangement. *J. Vis. Commun. Image Represent.* **2019**, in press. [[CrossRef](#)]
33. Qin, C.; Chang, C.C.; Hsu, T.J. Reversible data hiding scheme based on exploiting modification direction with two steganographic images. *Multimed. Tools Appl.* **2015**, *74*, 5861–5872. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).