symmetry

MDPI

Article

A Novel (2, 3)-Threshold Reversible Secret Image Sharing Scheme Based on Optimized Crystal-Lattice Matrix

Jiang-Yi Lin [1,2], Ji-Hwei Horng [3,*] and Chin-Chen Chang [1,*]

[1] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan; 2011110704@xmut.edu.cn
[2] School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
[3] Department of Electronic Engineering, National Quemoy University, Kinmen 89250, Taiwan
* Correspondence: horng@email.nqu.edu.tw (J.-H.H.); ccc@o365.fcu.edu.tw (C.-C.C.)

Abstract: The (k, n)-threshold reversible secret image sharing (RSIS) is technology that conceals the secret data in a cover image and produces n shadow versions. While k ($k \leq n$) or more shadows are gathered, the embedded secret data and the cover image can be retrieved without any error. This article proposes an optimal (2, 3) RSIS algorithm based on a crystal-lattice matrix. Sized by the assigned embedding capacity, a crystal-lattice model is first generated by simulating the crystal growth phenomenon with a greedy algorithm. A three-dimensional (3D) reference matrix based on translationally symmetric alignment of crystal-lattice models is constructed to guide production of the three secret image shadows. Any two of the three different shares can cooperate to restore the secret data and the cover image. When all three image shares are available, the third share can be applied to authenticate the obtained image shares. Experimental results prove that the proposed scheme can produce secret image shares with a better visual quality than other related works.

Keywords: (2, 3) secret image sharing; reversible data hiding; crystal-lattice matrix; authentication

Citation: Lin, J.-Y.; Horng, J.-H.; Chang, C.-C. A Novel (2, 3)-Threshold Reversible Secret Image Sharing Scheme Based on Optimized Crystal-Lattice Matrix. *Symmetry* **2021**, *13*, 2063. https://doi.org/10.3390/sym13112063

Academic Editor: Yu-Chi Chen

Received: 24 September 2021
Accepted: 26 October 2021
Published: 1 November 2021

1. Introduction

With the rapid development of the information technology, people can transmit data to each other through the internet. However, plaintext transmitted in the network is very easy to access, duplicate, temper, or even destroy by malicious attackers. Thus, the concern of data transmission security emerged. Therefore, image steganography techniques, for instance reversible and irreversible data hiding, have been introduced to conceal the secret data in cover images. Currently we have, according to the technique core, data hiding schemes which can be roughly categorized into the LSB substation [1,2], the difference expansion (DE) [3,4], the histogram shifting (HS) [5,6], the reference matrix-based [7–10], and the pixel-value differencing (PVD) [11,12] approaches. Since the modification of the cover image is very subtle, the constructed marked images cannot be distinguished from the cover one visually. Therefore, these data hiding techniques have significantly enhanced the security level of data transmission.

Instead of hiding secret data in a single cover image, the visual cryptography proposed by Naor and Shamir [13] hides secret data in multiple image shadows. Their method consists of two phases. First, in the dealing phase, a dealer divides the secret data in n image shadows and distributes the data to different participants. Secondly, in the reconstruction phase, k or more than k shadows are gathered and stacked together. The secret data can be retrieved without any error. However, the visual cryptography suffers from two problems. First, each image share looks meaningless and may catch the eavesdroppers' attention during transmission. Secondly, the produced image shadows are larger than the secret image in scale.

Later, many different secret image sharing (SIS) schemes [14–17] have been proposed. The method in [14] preserves the image scale and the secret image can be retrieved di-

rectly by stacking two transparencies. In [15], the binary secret image can be gained by superimposing any k of n meaningful shadows without performing any cryptographic computation. In 2020, Harn et. al. [17] proposed a secret image sharing scheme with a secure secret reconstruction process. In their method, the secret can be protected from both the attacks of insiders and outsiders. More specifically, the outsiders need to intercept all the released shares to recover the secret, which is impossible.

The dual-image-based reversible data hiding (RDH) scheme [18–21] can be regarded as a special case, with $k = n = 2$, of the (k, n) secret image sharing. The first dual-image-based scheme was proposed by Chang et al. [18] in 2007. In their method, each cover pixel pair was used to conceal two 5-base digits along the main and the anti-diagonal direction of the EMD matrix. The embedding capacity (EC) of their method was only about 1 bit per pixel (bpp). Later, an improved version proposed by Chang et al. [19] used the horizontal and the vertical directions of the EMD matrix instead of the main and anti-diagonal directions to embed the two 5-base digits. The peak signal-to-noise ratio (PSNR) raised to 48 dB while maintaining the same EC as [18]. In 2013, Lee and Huang [21] developed a novel reversible data hiding scheme using two shadows, which utilized the combination of the orientations in the corresponding stego pixel pairs to fulfill the reversibility. The EC of their method was 1.07 bpp and the visual quality of the image shadow was improved to 49 dB. In 2021, Chen et al. [20] introduced a dual-image-based RDH scheme using a EMD reference matrix. Each pixel in the cover image is embedded with $(1 + \log_2 5)$ secret bits with the help of a random binary stream. Although the EC of that method is higher as 1.56 bpp, the PSNR of the constructed shadows is less than 42 dB.

The authentication ability has attracted the attention of many scholars, for instance, the original batch verification using summation polynomials [22,23] and batch verification based on blockchain and ECDSA technology [24]. Nevertheless, these methods are realized with the help of a key generation system or a public blockchain center. The authentication in image domain is that a tampered shadow can be detected directly by a legal one without other assistance. The first authenticable secret sharing scheme was proposed by Yang et al. [25] in 2007. However, the authentication ability and image visual quality of their scheme were not satisfactory. To improve the drawback of the method, Liu et al. [26] proposed a novel (2, 2) secret sharing scheme based on the TS reference matrix in 2018. Since the modification of the cover pixel value does not exceed two, good visual quality can be guaranteed in their method. Furthermore, the difference between pairwise generated stego pixels do not exceed two either, so the cheating detection rate based on this property can reach to 95%. Later, Lin et al. [27] proposed a novel (2, 2) secret sharing scheme with the help of the EMD reference matrix in 2019. In comparison with the method in [28], EC and cheating detection ratio are about the same, but the visual quality of image shadows has been greatly improved. Subsequently, different secret image sharing schemes with their authentication mechanisms were proposed [28,29].

The disadvantage of the (2, 2) secret sharing scheme is that it needs both shadows to be gathered to extract the secret data and restore the cover image. In 2020, Gao et al. [30] proposed a (2, 3) reversible secret image sharing scheme based on a fractal matrix. In their method, the secret data is embedded in three shadows of the cover image through the guidance of a fractal matrix. The secret data and the cover image can be retrieved by any two of the three shadows, but image distortion may occur. In this paper, we introduce an optimal (2, 3) reversible secret sharing scheme based on a crystal-lattice matrix. The advantageous features of our method are listed below:

1. Produce image shadows with least distortion;
2. Guarantee the reversibility using any two of the three image shadows;
3. Perform an excellent cheating detection ratio.

The rest of this article is organized as follows. Section 2 introduces the method proposed by Gao et al. in [30]. Section 3 presents the proposed crystal-lattice matrix and the image shadow production process in detail. Our experimental results are illustrated in Section 4. Finally, the conclusions are summarized in Section 5.

2. Review of Gao et al.'s Method

The (2, 3) reversible secret sharing scheme proposed by Gao et al. [30] is composed of three steps, including the fractal matrix construction phase, the image shadow production phase, and the data extraction together with the image recovery phase, as introduced in Sections 2.1–2.3, respectively.

2.1. Fractal Matrix Construction Phase

In the method implemented by Gao et al., they defined two types of fractal groups which are composed of four $2 \times 2 \times 2$ fractal models and nine $3 \times 3 \times 3$ fractal models, respectively, as shown in Figures 1–3. The projections of both fractal models on the three axial planes are a perfect square, as shown in Figures 4 and 5. As shown in the figure, each location at a square projection is occupied by a unique model element. Furthermore, the same conclusion can be found in the projections of the two fractal groups on the three axial planes, shown in Figures 6 and 7.

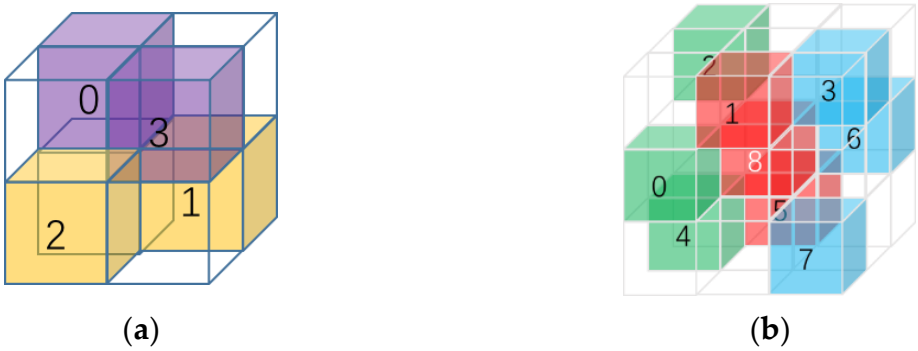


Figure 1. Two types of fractal models. (**a**) Type I: fractal model sized $2 \times 2 \times 2$; (**b**) Type II: fractal model sized $3 \times 3 \times 3$.

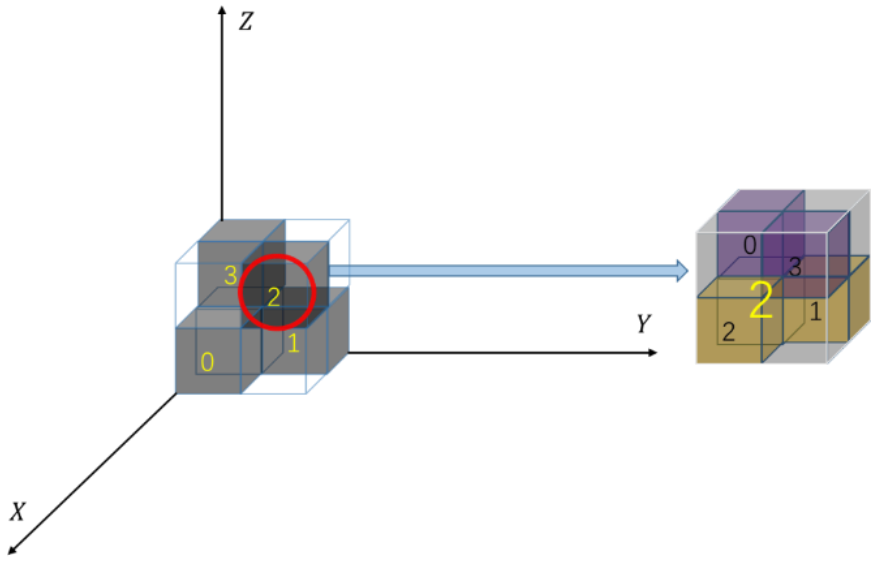


Figure 2. Fractal group constituted by Type I model.

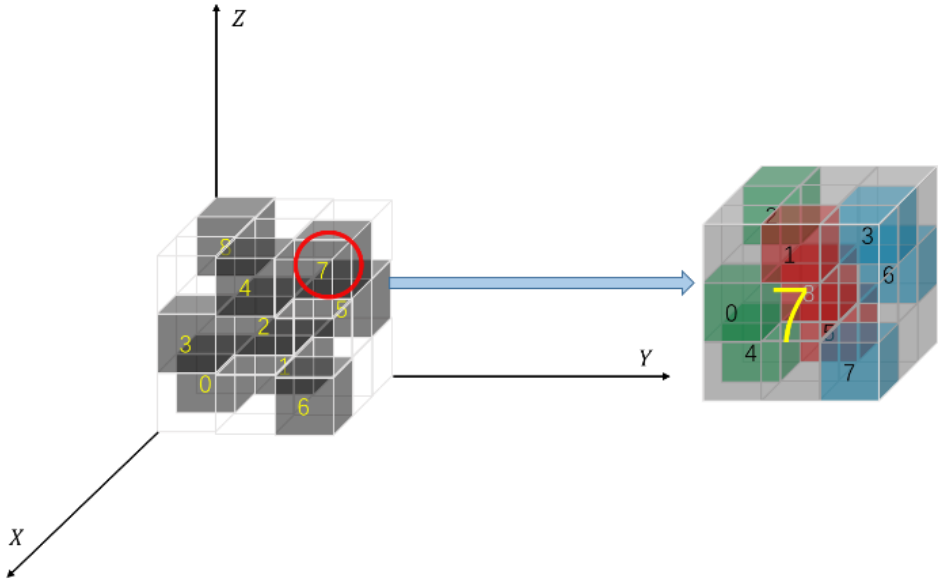


Figure 3. Fractal group constituted by Type II model.

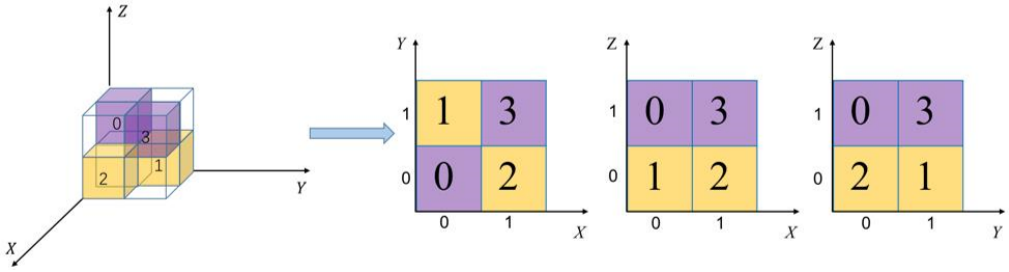


Figure 4. The projections of Type I fractal model on the axial planes.

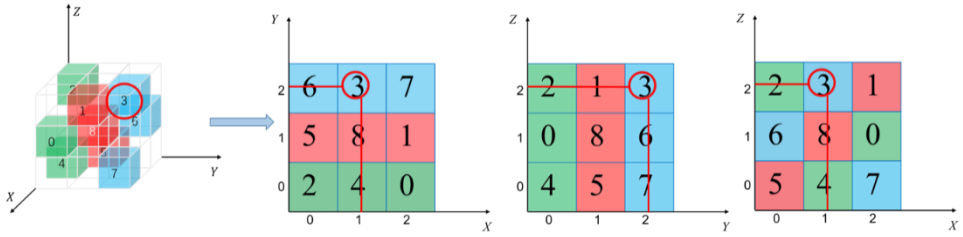


Figure 5. The projections of Type II fractal model on the axial planes.

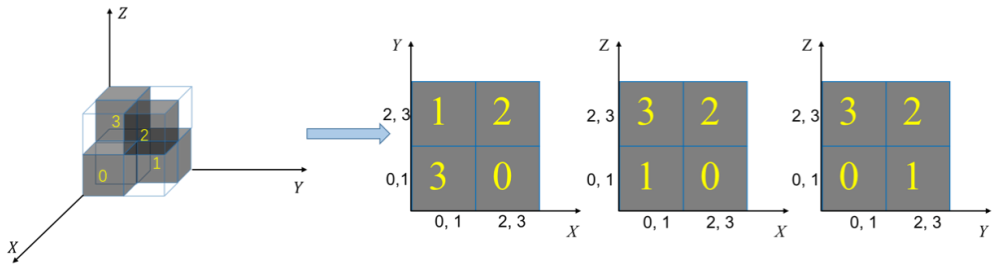


Figure 6. The projections of Type I fractal group on the axial planes.

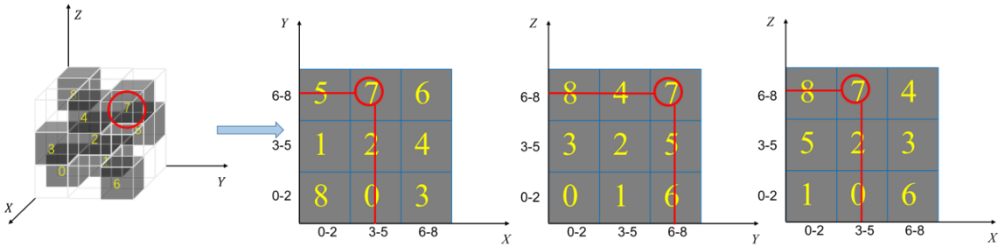


Figure 7. The projections of Type II fractal group on the axial planes.

The fractal matrix sized $256 \times 256 \times 256$ is constructed by arranging fractal groups consecutively along the main diagonal direction. Since the sizes of Type I and Type II fractal groups are $4 \times 4 \times 4$ and $9 \times 9 \times 9$, respectively, the fractal matrix contains $\lfloor 256/4 \rfloor = 64$ adjacent Type I fractal groups or $\lfloor 256/9 \rfloor = 28$ adjacent Type II fractal groups, where $\lfloor \cdot \rfloor$ denotes the floor function. Without the loss of generality, the fractal matrix with Type II fractal model sized $3 \times 3 \times 3$ is applied in the following description.

2.2. Image Shadow Production Phase

Based on a cover image I, their scheme produces three image shadows, S_1, S_2, and S_3, with the guidance of the fractal matrix. The given cover image I sized $W \times H$ is first rearranged into a pixel sequence $I_V = \{p_i, i = 1, 2, \ldots, W \times H\}$ in the raster scan order. Then, the pixels in the sequence are consecutively processed. Each time, a pixel p_i is duplicated into a triplet (p_i, p_i, p_i) and the triplet is modified into (p_{i1}, p_{i2}, p_{i3}) according to the given secret digit q_k and the fractal matrix. The pixel values of the modified triplet are then separately recorded in the three image shadows of the corresponding spatial location.

The rules of modification are as follows. First, the triplet (p_i, p_i, p_i) is treated as the 3D coordinates of an element in the fractal matrix. Since the coordinates of the three axes are identical, the located element lays on the main diagonal line of the fractal matrix. Recall that if the main diagonal line is consecutively arranged with fractal groups, the located element must be within a fractal group. The index of the located fractal group can be determined by $n_G = \lfloor p_i/9 \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor operation. The $9 \times 9 \times 9$ space occupied by a fractal group contains nine main diagonal elements as well as nine fractal models. To ensure the reversibility, a one-to-one mapping between the nine main diagonal elements and the nine fractal models is preassigned. An example of numbering the fractal models is shown in Figure 7. Thus, the target fractal model $F_{n_M}(x, y, z)$ sized $3 \times 3 \times 3$ for data embedding can be determined by $n_M = p_i \bmod 9$, where mod is the modulo operation. The exact target element of embedding is determined by the 9-based secret digit q_k, which satisfies $F_{n_M}(x_t, y_t, z_t) = q_k$. Finally, the shadow pixels (p_{i1}, p_{i2}, p_{i3}) can be obtained by

$$\begin{cases} p_{i1} = 9 \times n_G + x_t, \\ p_{i2} = 9 \times n_G + y_t, \\ p_{i3} = 9 \times n_G + z_t. \end{cases} \tag{1}$$

A simple example is elaborated for a better understanding of the embedding phase of Gao et al.'s method. Suppose the cover pixel $p_i = 16$ and the 9-based secret digit $q_k = 3$. The index of the target fractal group is $n_G = \lfloor p_i/9 \rfloor = 1$, and the number of the fractal model used for data embedding is determined by $n_M = 16 \bmod 9 = 7$. According to Figure 7, the target fractal model located by $n_M = 7$ is $F_{n_M}(3:5, 6:8, 6:8)$, as circled in red in 3D and 2D projected versions. To embed the secret digit $q_k = 3$, the exact matched element is $F_{n_M}(4, 8, 8) = 3$, as shown in Figure 5. Thus, the shadow pixels (p_{i1}, p_{i2}, p_{i3}) can be obtained by

$$\begin{cases} p_{i1} = 9 \times 1 + 4 = 13, \\ p_{i2} = 9 \times 1 + 8 = 17, \\ p_{i3} = 9 \times 1 + 8 = 17. \end{cases} \tag{2}$$

Finally, the shadow pixels are recorded into shadow images S_1, S_2, and S_3. Notice that 9 is not a factor of 256, so four pixel values are not covered by any fractal group. In their method, the pixel values 0, 1, 254, and 255 are left intact. To simplify explanation, the pixel values 0 and 1 are not excluded from fractal groups in our demonstration.

2.3. Data Extraction and Cover Image Restoration Phase

By using any two of the three shadows, Gao et al.'s method can extract secret data and restore the cover image. Without a loss of generality, suppose shadows S_1 and S_2 are applied to decrypt secret data and cover image. The pixels in both images are rearranged into vector sequences $S_{V1} = \{p_{1i}, i = 1, 2, \ldots, W \times H\}$ and $S_{V2} = \{p_{2i}, i = 1, 2, \ldots, W \times H\}$ first. Then, consecutively process the pixel pair (p_{i1}, p_{i2}) to decrypt data. Take the pixel pair $(p_{i1}, p_{i2}) = (13, 17)$ as an example. Its corresponding fractal group is located by $n_G = \lfloor p_{1i}/9 \rfloor = 1$. Then, its projected coordinates in the fractal group can be obtained by $(R_x, R_y) = (13 \bmod 9,\ 17 \bmod 9) = (4, 8)$, which belongs to the fractal model $F_{n_M}(3:5, 6:8, 6:8)$. Note that the range of the z-coordinate is unique by referring to Figure 8. The coordinates $(4, 8)$ map to model index $n_M = 7$, whose range of z-coordinate can be further determined by referring to yz or xz projection. The secret digit can be extracted by applying the modulo operation $(R_{x2}, R_{y2}) = (13 \bmod 3,\ 17 \bmod 3)$=$(1, 2)$ and mapping to the fractal model, as shown in Figure 9. The mapped value at $(1,\ 2)$ of xy-projection is $q_k = 3$. Finally, the cover pixel value can be restored by $p_i = 9 \times n_G + n_M = 9 \times 1 + 7 = 16$.

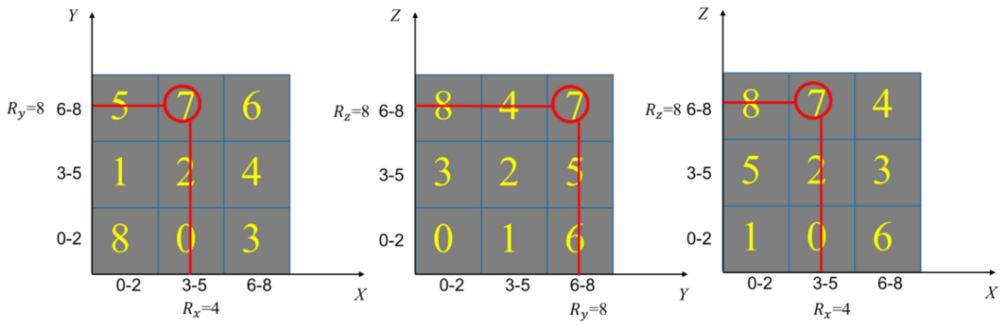


Figure 8. The xy-, yz-, and xz-projections of a fractal group.

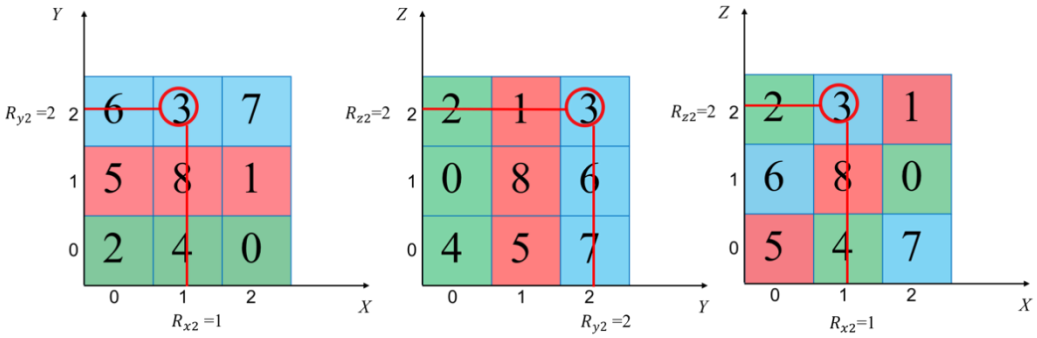


Figure 9. The xy-, yz-, and xz-projections of a fractal model.

Based on the $9 \times 9 \times 9$ fractal group, a reversible (2, 3) secret image sharing scheme can be realized. Two $9 \times 9 \times 9$ fractal groups and their projections on the xy, yz, and zx-planes are plotted in Figure 10, where each group contains 9 fractal models displayed with different colors. Recall that the original cover pixel triplet (p_i, p_i, p_i) lays on the main diagonal line. To embed secret data, the pixel values are modified into the space occupied by the fractal groups. The deviation of the target element from the main diagonal line directly influences the distortion of pixel values in the image shadows. To produce image shadows with a minimum distortion, the target elements should be arranged to the surroundings of the main diagonal line. Observe that the vicinity of the conjunction

points between fractal groups are not fully exploited to embed data. It indicates that further improvement of shadow image quality is possible.

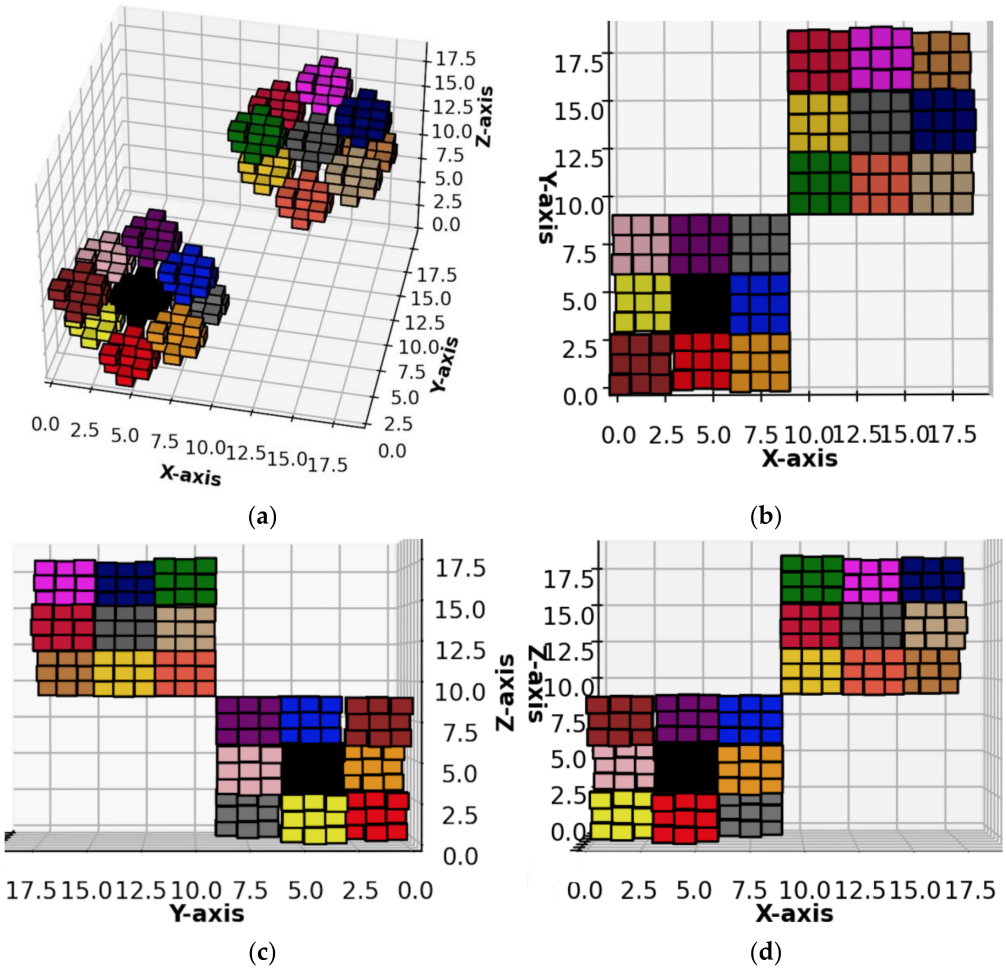


Figure 10. Two fractal groups displayed in the 3D space: (**a**) a 3D view, (**b**) the xy-projection, (**c**) the yz -projection, (**d**) the zx -projection.

3. Proposed Scheme

The proposed reversible (2, 3) threshold secret image sharing scheme is based on the same frame structure as the fractal matrix-based scheme. The crystal-lattice matrix is proposed to address the weakness of the fractal matrix. Construction of the crystal-lattice matrix is firstly introduced in Section 3.1, where the growth of a crystal lattice and the construction of a crystal-lattice matrix are also presented. The shadow image generation phase and the decryption phase are explained in Sections 3.2 and 3.3, respectively.

3.1. Crystal-Lattice Matrix

To reduce the distortion of secret image shadows, the fundamental model associated with each element on the main diagonal line of the 3D reference matrix should be arranged closely around it. To achieve this goal, a greedy algorithm is proposed to construct an optimal fundamental model. Inspired by the growth of a crystal material, we treated the elements on the main diagonal line as the seeds for crystallization. The fundamental models are the lattices grown simultaneously and crowdedly toward the radial directions from the stream of seeds. Meanwhile, the projections of the lattices on each axial plane should be unique to meet the requirement of (2, 3)-threshold secret sharing. Subject to these constraints, the greedy algorithm is applied to append the nearest element, one at a time, to each lattice at the same relative location until the predefined range is fully

searched. Suppose the seed elements are $(x, y, z) = (p, p, p)$, $p \in [w : 255 - w]$ and w is the window width of the search range. The candidate elements to be appended are $(p + d_x, p + d_y, p + d_z)$, $p \in [w : 255 - w]$ and $d_x, d_y, d_z \in [-w : +w]$. To ensure a greedy choice that minimizes the distortion each time, the candidates within the search range are fully listed and sorted in the ascending order of Euclidean distance. Table 1 lists the sorted processing queue of $w = 1$ together with their square Euclidean distance

$$D = (d_x)^2 + (d_y)^2 + (d_z)^2 \tag{3}$$

Table 1. The candidate elements in the processing queue.

Index	d_x	d_y	d_z	D	Index	d_x	d_y	d_z	D	Index	d_x	d_y	d_z	D
0	0	0	0	0	9	−1	0	1	2	18	1	1	0	2
1	0	−1	0	1	10	−1	1	0	2	19	−1	1	−1	3
2	0	0	−1	1	11	0	1	−1	2	20	1	−1	1	3
3	−1	0	0	1	12	0	−1	1	2	21	−1	−1	1	3
4	0	0	1	1	13	0	1	1	2	22	−1	1	1	3
5	0	1	0	1	14	1	−1	0	2	23	−1	−1	−1	3
6	1	0	0	1	15	1	0	−1	2	24	1	1	−1	3
7	0	−1	1	2	16	−1	0	−1	2	25	1	−1	−1	3
8	−1	−1	0	2	17	1	0	1	2	26	1	1	1	3

The crystal growth Algorithm 1 is summarized as follows.

The lattice model $\mathcal{M}$ and the crystal-lattice matrix $\mathcal{C}$ are equivalent. The former records the deviation vectors from the seed of all elements in a crystal lattice; the latter is a fully sized matrix which labels the lattice index of each matrix element. The two versions can be converted into each other through simple manipulations.

In Step 1, switching the order of scanning the elements in the predefined search range may change the queue list and the resulting output. As shown in Table 1, the elements indexed 1 to 6 in the queue are equidistant from the seed. Switching scanning order may change the order of these elements and thus change the greedy selection. Some possible results are mutually spatial symmetric. However, this factor does not lead to significant influence on the output performance.

A fully sized matrix $\mathcal{C}$ is created in Step 2 to record the lattice index of each occupied element. The initial value v_{max} is used to indicate an unoccupied state. Three two-dimensional matrices $\mathcal{P}_{xy}$, $\mathcal{P}_{yz}$, and $\mathcal{P}_{zx}$ are created in Step 3 to record the projected locations of included elements. In Step 4, we check the simultaneous growth of all models by including the new greedy choice that do not overlap each other in 3D space and the projected axial planes first. When the choice is available, it is recorded to the lattice model and the labeling matrices. Note that Equation (6) is not necessary, since Equations (7)–(9) are stricter constraints.

The final volume of the lattice model is determined by the required payload of each cover pixel. When the payload of each cover pixel is s bits, the final lattice volume is 2^n. A proper window width w should be set to ensure a sufficient range of searching. A slightly oversized window width is alright.

Table 2 lists the set of deviation vectors for the lattice model with $2^n = 16$. The 3D view of its corresponding crystal-lattice matrix together with projections on the three axial planes are provided in Figure 11. The embeddable elements are translationally symmetric duplications of the crystal-lattice model along the main diagonal line. The distribution is approximately a cylindrical shape, as expected. In addition, the projection views demonstrate the uniqueness at each location. As shown in the figures, the elements of a crystal lattice are not connected. An element that violates any of Equations (7)–(9) is not available for embedding. This strict rule results in a sparse distribution of the lattice elements.

Algorithm 1. The crystal growth algorithm

Input: The window width w, the lattice model size 2^n, the secret key $\mathcal{K}$.

Output: The lattice model $\mathcal{M} = \left\{ \left(d_x^m, d_y^m, d_z^m \right) \middle| m = 0, 1, 2, \ldots, 2^n \right\}$, the crystal-lattice matrix $\mathcal{C} = \{\mathcal{C}(x, y, z) | 0 \leq x, y, z < 256\}$.

1. Scan the search range, sort the candidate elements in the ascending order of Euclidean distance, and list the processing queue $\mathcal{Q} = \left\{ \left(\hat{d}_x^k, \hat{d}_y^k, \hat{d}_z^k \right), k = 0, 1, 2, \ldots, (2w+1)^3 - 1 \right\}$.
2. Initialize the counter $m = 0$ and the 3D matrix $\mathcal{C}$ sized $256 \times 256 \times 256$ by

$$\mathcal{C}(x, y, z) = \begin{cases} x, x = y = z \; and \; x \in [w : 255 - w] \\ v_{max}, \; \text{otherwise} \end{cases} \tag{4}$$

3. Initialize the three projection matrices $\mathcal{P}_{xy}$, $\mathcal{P}_{yz}$, and $\mathcal{P}_{zx}$ sized 256×256 by

$$\mathcal{P}_{xy}(i, j), \mathcal{P}_{yz}(i, j), \mathcal{P}_{zx}(i, j) = \begin{cases} i, \; i = j \; and \; i \in [w : 255 - w] \\ v_{max}, \; \text{otherwise} otherwise \end{cases} \tag{5}$$

4. Retrieve an element $\left(\hat{d}_x^k, \hat{d}_y^k, \hat{d}_z^k \right)$ from $\mathcal{Q}$. If Equations (6)–(9) hold, record $\left(\hat{d}_x^k, \hat{d}_y^k, \hat{d}_z^k \right)$ to $\mathcal{M}$, mark the matrix elements by Equations (10)–(13), and update the counter $m = m + 1$; else, skip this element.

$$\mathcal{C}\left(p + \hat{d}_x^k, p + \hat{d}_y^k, p + \hat{d}_z^k \right) = v_{max}, \; p \in [w : 255 - w] \tag{6}$$

$$\mathcal{P}_{xy}\left(p + \hat{d}_x^k, p + \hat{d}_y^k \right) = v_{max}, \; p \in [w : 255 - w] \tag{7}$$

$$\mathcal{P}_{yz}\left(p + \hat{d}_y^k, p + \hat{d}_z^k \right) = v_{max}, \; p \in [w : 255 - w] \tag{8}$$

$$\mathcal{P}_{zx}\left(p + \hat{d}_z^k, p + \hat{d}_x^k \right) = v_{max}, \; p \in [w : 255 - w] \tag{9}$$

$$\mathcal{C}\left(p + \hat{d}_x^k, p + \hat{d}_y^k, p + \hat{d}_z^k \right) = m, \; p \in [w : 255 - w] \tag{10}$$

$$\mathcal{P}_{xy}\left(p + \hat{d}_x^k, p + \hat{d}_y^k \right) = m, \; p \in [w : 255 - w] \tag{11}$$

$$\mathcal{P}_{yz}\left(p + \hat{d}_y^k, p + \hat{d}_z^k \right) = m, \; p \in [w : 255 - w] \tag{12}$$

$$\mathcal{P}_{zx}\left(p + \hat{d}_z^k, p + \hat{d}_x^k \right) = m, \; p \in [w : 255 - w] \tag{13}$$

5. Repeat Step 4 until the required queue volume 2^n is satisfied.
6. Fill each lattice model with a random permutation of 0 to $2^n - 1$ generated by key $\mathcal{K}$

Table 2. The lattice model $\mathcal{M}$ with $2^n = 16$.

Index	d_x	d_y	d_z	Index	d_x	d_y	d_z
0	0	0	0	8	−4	3	0
1	−1	0	1	9	4	−4	0
2	−1	1	0	10	4	0	−4
3	1	−2	0	11	0	−5	5
4	1	0	−2	12	0	5	−5
5	0	−2	3	13	−6	0	6
6	0	3	−2	14	−6	6	0

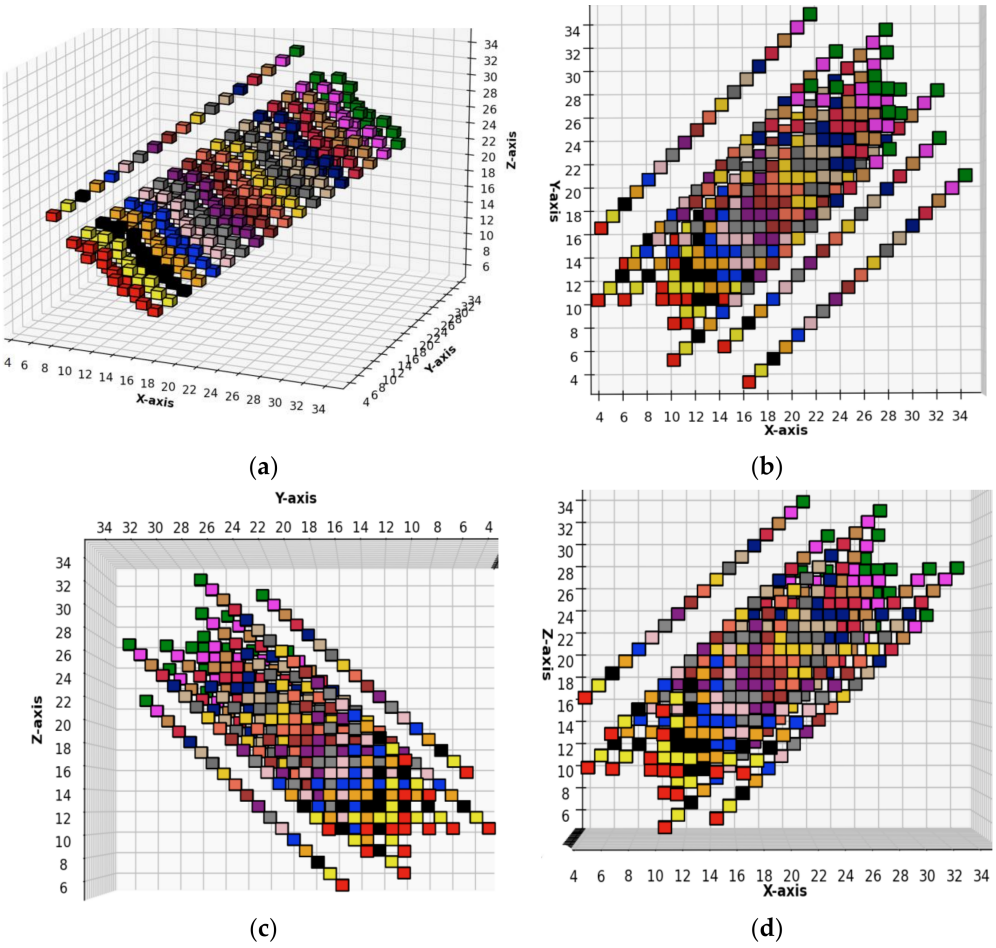


Figure 11. Partial view of the resulting crystal-lattice matrix with $2^n = 16$. (**a**) The crystal-lattice matrix $\mathcal{C}$, (**b**) the projection view $\mathcal{P}_{xy}$, (**c**) the projection view $\mathcal{P}_{yz}$, (**d**) the projection view $\mathcal{P}_{zx}$.

Recall that a lattice model is the embeddable space of its corresponding seed element. Before the crystal-lattice matrix can be applied as the 3D reference matrix for data embedding, a random permutation of distinct integer values from 0 to $2^n - 1$ should be assigned to the elements of each lattice model. The random permutation can be determined by a secret key $\mathcal{K}$ shared in advance.

3.2. Shadow Image Generation

As mentioned above, the proposed data hiding scheme shares the same scenario as the fractal matrix-based scheme proposed by Gao et al. in [30]. The system diagram of the new proposed scheme is shown in Figure 12. Through the cover of a regular image, three indistinguishable data-embedded shadows are generated and separately distributed to three participants. Any two participants can cooperate to decrypt the secret data and the cover image losslessly. When all three shadows are available, the third shadow can be exploited to check the integrity of these shadows. The shadow generation Algorithm 2 is given as follows.

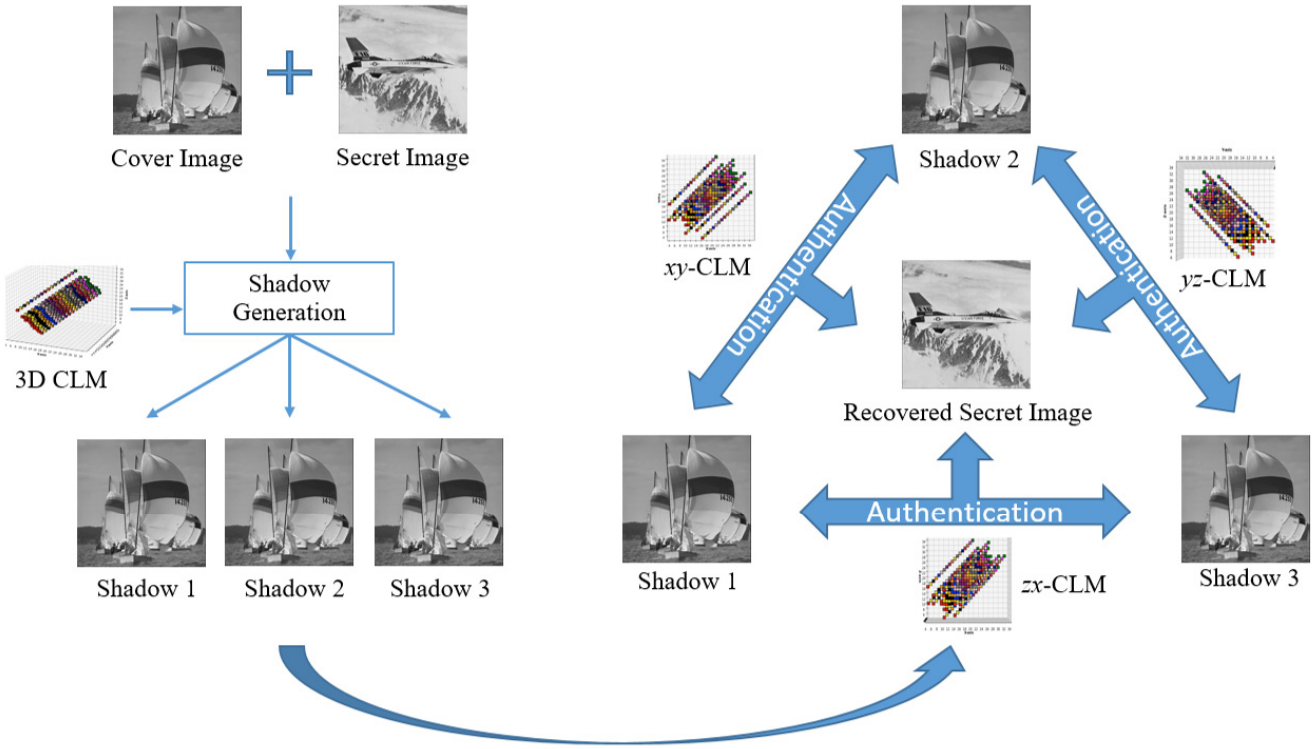


Figure 12. The system diagram of the proposed scheme.

Algorithm 2. The shadow generation algorithm

Input: The cover image $\mathbb{I}$, the binary secret stream $\mathcal{S}$, the parameters n, w, and the key $\mathcal{K}$.
Output: Three image shadows $\mathbb{S}_1$, $\mathbb{S}_2$, and $\mathbb{S}_3$.

1. Construct the crystal-lattice matrix $\mathcal{C}$ according to n, w, and the key $\mathcal{K}$.
2. Convert $\mathcal{S}$ into 2^n-ary number sequence $\mathcal{S}_n = \{s_k | k = 1, 2, \ldots, L\}$.
3. Rearrange $\mathbb{I}$ into a sequence $\mathbb{I}_V = \{p_i, i = 1, 2, \ldots, W \times H\}$ in the raster scan order.
4. For each pixel in $\mathbb{I}_V$, do

   If $p_i \in [w : 255 - w]$,
   Retrieve a secret digit s_k.
   Find $\mathcal{C}(p_{i1}, p_{i2}, p_{i3}) = s_k$ subject to $\mathcal{C}(p_{i1}, p_{i2}, p_{i3}) \in \mathcal{M}(p_i, p_i, p_i)$.
   Record p_{i1}, p_{i2}, p_{i3} to $\mathbb{S}_1, \mathbb{S}_2, \mathbb{S}_3$, respectively.
   Else
   Record p_i, p_i, p_i to $\mathbb{S}_1, \mathbb{S}_2, \mathbb{S}_3$, respectively.
   End

5. Terminate Step 4 when the secret sequence is exhausted.
6. Copy the remaining cover pixel values to the image shadows directly and close all files.

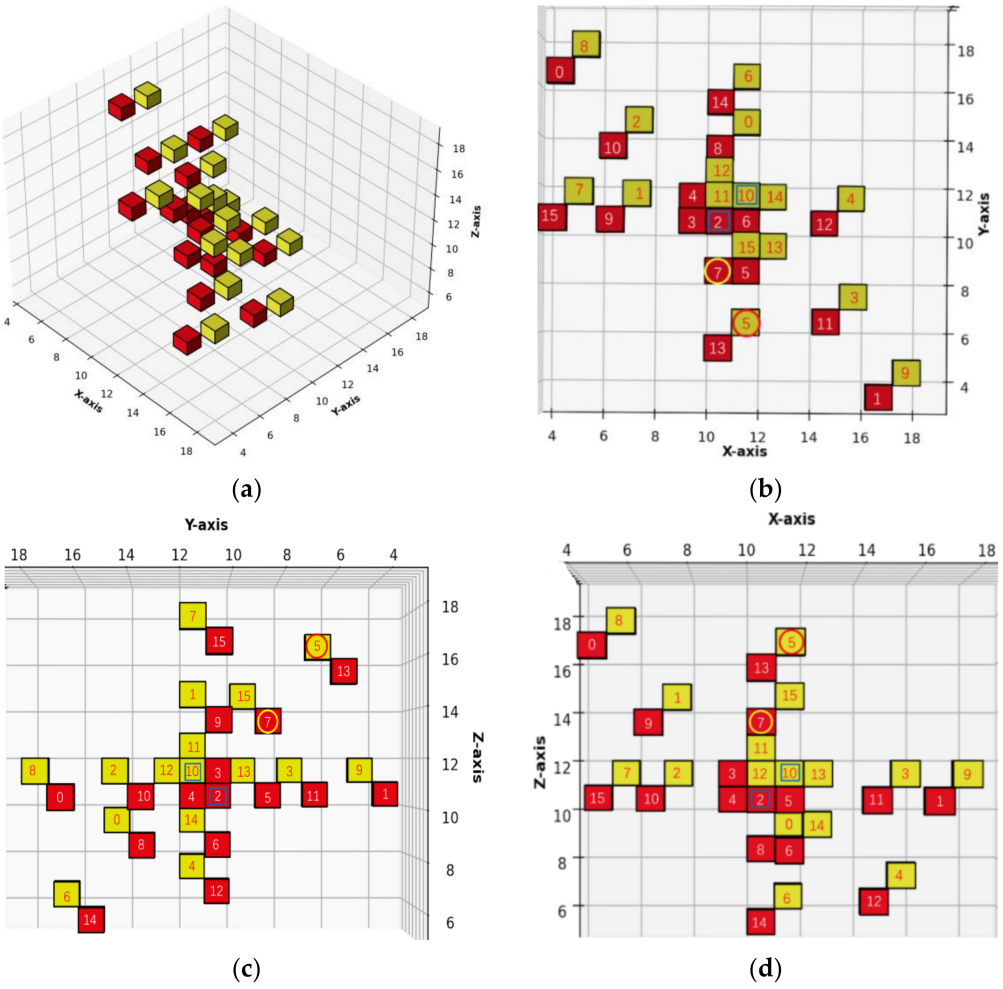
The notation $\mathcal{M}(p_i, p_i, p_i)$ represents a translated lattice model $\mathcal{M}(p_i, p_i, p_i) = \left\{ \left(p_i + d_x^m, p_i + d_y^m, p_i + d_z^m \right) \middle| m = 0, 1, 2, \ldots, 2^n \right\}$, whose seed element is $\mathcal{C}(p_i, p_i, p_i)$. To further elaborate the key process in Step 4, an example has been provided. Suppose the cover pixels are $\mathbb{I}_V = \{5, 10, 11\}$, $n = 4$, $w = 7$, and the secret digits are $\mathcal{S}_{16} = \{7, 5\}$. The detail of processing the three cover pixels are as follows.

(1) Pixel $p_i = 5$: This pixel value is not within the range of $[w : 255 - w] = [7 : 148]$, it is not embeddable and the duplications 5, 5, 5 are recorded to $\mathbb{S}_1, \mathbb{S}_2, \mathbb{S}_3$, respectively.

(2) Pixel $p_i = 10$: This pixel value belongs to the embeddable range, a secret digit $s_k = 7$ is retrieved from $\mathcal{S}_{16}$. The translated lattice model $\mathcal{M}(10, 10, 10)$ is the group of red elements displayed in Figure 13a, whose projections on the three axial planes are displayed in Figure 13b–d. Its seed element valued 2 is squared in blue in the projection views. Since the secret digit to be embedded is $s_k = 7$, the element

$\mathcal{C}(10,8,13) = 7$, circled in yellow, is the targeted. The pixel values $10,8,13$ are recorded to $\mathbb{S}_1, \mathbb{S}_2, \mathbb{S}_3$, respectively.

(3) Pixel $p_i = 11$: This pixel value also belongs to the embeddable range, the next digit $s_k = 5$ is retrieved from $\mathcal{S}_{16}$. The translated lattice model $\mathcal{M}(11,11,11)$ is the group of yellow elements displayed in Figure 13. Its seed element valued 10 is squared in blue in the projection views. Since the secret digit to be embedded is s_k=5, the element $\mathcal{C}(11,6,16) = 5$, circled in red, is the targeted. The pixel values $11,6,16$ are recorded to $\mathbb{S}_1, \mathbb{S}_2, \mathbb{S}_3$, respectively. The resulting shadows are $\mathbb{S}_1 = \{5,10,11\}$, $\mathbb{S}_2 = \{5,8,6\}$, $\mathbb{S}_3 = \{5,13,16\}$.



(**a**) (**b**) (**c**) (**d**)

Figure 13. Separate display of two lattice models in the example. (**a**) The 3D view of lattice models $\mathcal{M}(10,10,10)$ and $\mathcal{M}(11,11,11)$, (**b**) the projection $\mathcal{P}_{xy}$, (**c**) the projection $\mathcal{P}_{yz}$. (**d**) the projection $\mathcal{P}_{zx}$.

3.3. Secret Decryption, Image Recovery, and Authentication

Recall that any two shadows among the three can decrypt the secret and restore the cover image for a (2, 3) threshold RSIS scheme. Without a loss of generality, suppose the shadows $\mathbb{S}_1$ and $\mathbb{S}_2$ are available; the following Algorithm 3 can be applied to decrypt the secret digits and restore the cover image.

Algorithm 3. The secret decryption and image recovery algorithm

Input: Two image shadows $\mathbb{S}_1$ and $\mathbb{S}_2$, the matrix parameters n, w, the key $\mathcal{K}$.
Output: The cover image $\mathbb{I}$, the binary secret stream $\mathcal{S}$.

1. Construct the crystal-lattice matrix $\mathcal{C}$ according to n, w, and the key $\mathcal{K}$.
2. Create the projection matrix $\mathcal{P}_{xy}$ and fill in the element values by referring to $\mathcal{C}$.
3. Rearrange $\mathbb{S}_1$ and $\mathbb{S}_2$ into pixel sequences $\mathbb{S}_{1V} = \{p_{1i}, i = 1, 2, \ldots, W \times H\}$ and $\mathbb{S}_{2V} = \{p_{2i}, i = 1, 2, \ldots, W \times H\}$ in the raster scan order.
4. For each pixel pair (p_{i1}, p_{i2}) in $\mathbb{S}_{1V}$ and $\mathbb{S}_{2V}$, do

   If $(p_{i1} = p_{i2})$ and $(p_{i1}, p_{i2} \in [0 : w-1]$ or $[255-(w-1) : 255])$,
   Record p_{i1} to $\mathbb{I}$.
   Else
   Find the secret digit and the cover pixel value by

$$s_k = \mathcal{P}_{xy}(p_{i1}, p_{i2}) \tag{14}$$

$$p_i = p_j, \text{subject to } \mathcal{P}_{xy}(p_{i1}, p_{i2}) \in \mathcal{M}_{xy}\left(p_j, p_j\right) \tag{15}$$

   Record the secret digit s_k to $\mathcal{S}_n$; record p_i to $\mathbb{I}$
   End
5. Convert $\mathcal{S}_n$ into the binary secret stream $\mathcal{S}$.

The notation $\mathcal{M}_{\text{xy}}(p_j, p_j)$ represents the projection of $\mathcal{M}(p_j, p_j, p_j)$ onto the xy-plane. The example secret image shadows $\mathbb{S}_1 = \{5, 10, 11\}$ and $\mathbb{S}_2 = \{5, 8, 6\}$ are applied to demonstrate the key process of Step 4. Three pixel pairs $(5, 5)$, $(10, 8)$, and $(11, 6)$ are consecutively processed as follows.

(1) Pixel pair $(p_{i1}, p_{i2}) = (5, 5)$: This pixel pair is constituted by equal value pixels and the value does not belong to the embeddable range. Therefore, record the value 5 to the output image directly.
(2) Pixel pair $(p_{i1}, p_{i2}) = (10, 8)$: By using $(10, 8)$ as the coordinates of $\mathcal{P}_{xy}(p_{i1}, p_{i2})$, refer to Figure 13b, the secret digit can be obtained by $\mathcal{P}_{xy}(10, 8) = 7$. In addition, the seed element of the $\mathcal{P}_{xy}(10, 8)$ is $\mathcal{P}_{xy}(10, 10)$. Therefore, the cover pixel value 10 is recorded to the output image.
(3) Pixel pair $(p_{i1}, p_{i2}) = (11, 6)$: Similarly, by using $(11, 6)$ as the coordinates of $\mathcal{P}_{xy}(p_{i1}, p_{i2})$, the secret digit can be obtained by $\mathcal{P}_{xy}(11, 6) = 5$. The seed element of the $\mathcal{P}_{xy}(11, 6)$ is $\mathcal{P}_{xy}(11, 11)$. Therefore, the cover pixel value 11 is recorded to the output image.

In the process of secret image generation, pixel values are in fact the spatial coordinates of the model elements. Recall that the crystal-lattice models are all seeded at the main diagonal line of the crystal-lattice matrix. Therefore, the embeddable elements are confined around the line. By leveraging data integrity of the image shadows, we can authenticate a suspected shadow based on a faithful share. Suppose we hold the faithful shadow $\mathbb{S}_1$. The authentication of the suspected shadow $\mathbb{S}_2$ is given in Algorithm 4.

Based on the same concept, we can devise an authentication algorithm for three image shadows. Since the secret binary stream and the cover image can be restored with two secret shares, the pixel values of the additional third share are uniquely determined. The data integrity of three shares provides a strong restriction to detect tampered shadows. The authentication for three image shadows is given in Algorithm 5.

Note that the two authentication algorithms are both based on the data integrity of image shadows. The tampered shares can only be detected based on faithful shares. When we only get a faithful share in hand, Algorithm 4 can be applied first to check data integrity. In case the integrity check is failed, we can detect the tampered share by using Algorithm 5. However, the detection rate of the two-shadow version is slightly weaker, which will be further discussed in the next section.

Algorithm 4. The authentication algorithm for two image shadows

Input: Two image shadows $\mathbb{S}_1$ and $\mathbb{S}_2$, the matrix parameters n, w, and the key $\mathcal{K}$.
Output: Authentication report.

1. Construct the crystal-lattice matrix $\mathcal{C}$ according to n, w, and the key $\mathcal{K}$.
2. Create the projection matrix $\mathcal{P}_{xy}$ and fill in the element values by referring to $\mathcal{C}$.
3. Rearrange $\mathbb{S}_1$ and $\mathbb{S}_2$ into pixel sequences $\mathbb{S}_{1V} = \{p_{1i}, i = 1, 2, \ldots, W \times H\}$ and $\mathbb{S}_{2V} = \{p_{2i}, i = 1, 2, \ldots, W \times H\}$ in the raster scan order.
4. For each pixel pair (p_{i1}, p_{i2}) in $\mathbb{S}_{1V}$ and $\mathbb{S}_{2V}$, do
   If $(p_{i1} = p_{i2})$ and $(p_{i1} \in [0 : w-1]$ or $[255-(w-1) : 255])$,
   Current pixel passed.
   Else
   If $\mathcal{P}_{xy}(p_{i1}, p_{i2}) \in \mathbb{U}$, current pixel passed,
   $$\mathbb{U} = \left\{ \mathcal{M}_{xy}\left(p_j, p_j\right) \middle| p_j \in [w : 255-w] \right\} \quad (16)$$
   Else Authentication failed and program stop.
   End
5. Image shadow authentication passed.

Algorithm 5. The authentication algorithm for three image shadows

Input: Three image shadows $\mathbb{S}_1$, $\mathbb{S}_2$, and $\mathbb{S}_3$, the matrix parameters n, w, and the key $\mathcal{K}$.
Output: Authentication report.

1. Construct the crystal-lattice matrix $\mathcal{C}$ according to n, w, and the key $\mathcal{K}$.
2. Rearrange $\mathbb{S}_1$, $\mathbb{S}_2$, and $\mathbb{S}_3$ into sequences $\mathbb{S}_{1V} = \{p_{1i}, i = 1, 2, \ldots, W \times H\}$, and $\mathbb{S}_{2V} = \{p_{2i}, i = 1, 2, \ldots, W \times H\}$, and $\mathbb{S}_{3V} = \{p_{3i}, i = 1, 2, \ldots, W \times H\}$ in the raster scan order.
3. For each pixel triplet (p_{i1}, p_{i2}, p_{i3}) in $\mathbb{S}_{1V}$, $\mathbb{S}_{2V}$, and $\mathbb{S}_{3V}$, do
   If $(p_{i1} = p_{i2} = p_{i3})$ and $(p_{i1} \in [0 : w-1]$ or $[255-(w-1) : 255])$,
   Current pixel passed.
   Else
   If $\mathcal{C}(p_{i1}, p_{i2}, p_{i3}) \in \mathbb{U}$, current pixel passed,
   $$\mathbb{U} = \left\{ \mathcal{M}\left(p_j, p_j, p_j\right) \middle| p_j \in [w : 255-w] \right\} \quad (17)$$
   Else Authentication failed and program stop.
   End
4. Image shadow authentication passed.

4. Experimental Results

This section demonstrates the performance of the proposed scheme by some simulations. The programs are all implemented by MATLAB R2017b software running on a MacBook Pro (Retina, 15-inch, Late 2013) computer. The macOS High Sierra operating system is loaded in the computer, and its CPU and RAM are 2.3 GHZ Intel Core i7 and 16 GB, respectively. Eight standard grayscale test images of size 512×512 are applied in our experiment, as shown in Figure 14.

Commonly, the PSNR, defined in Equation (18), is exploited to evaluate the quality of the generated shadows.

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{e_{MSE}}, (\text{dB}) \quad (18)$$

where e_{MSE}, defined in Equation (19), is the mean square error between the cover image and the compared shadow.

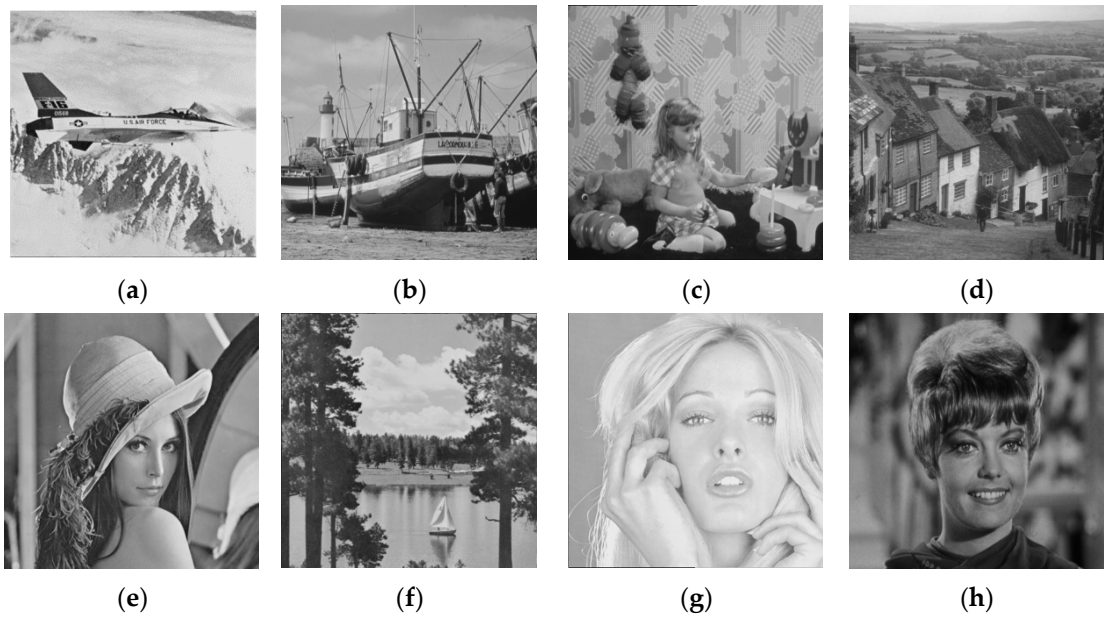


Figure 14. The eight test images sized 512 × 512. (**a**) Airplane, (**b**) Boat, (**c**) Girl, (**d**) Goldhill, (**e**) Lena, (**f**) Lake, (**g**) Tiffany, and (**h**) Zelda.

$$e_{MSE} = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \left(\mathbb{I}_{ij} - \mathbb{S}_{ij}\right)^2, \tag{19}$$

where W and H are the width and the height of the images. $\mathbb{I}_{ij}$ and $\mathbb{S}_{ij}$ are the pixel values at the location (i, j) of the cover image and the shadow, respectively.

The metric EC, defined in Equation (20), is the embedding capacity measured in bits per pixel (bpp),

$$\text{EC} = \frac{N_{\mathcal{S}}}{k \times W \times H}, \tag{20}$$

where $N_{\mathcal{S}}$ represents the total length of embedded secret stream and k is the number of shadows. Although three image shadows are generated, the restoration of secret data and cover image requires only two shares. We apply $k = 2$ to calculate EC in the following experimental data.

4.1. Visual Quality of Image Shadows

The matrix parameter n controls the volume of the lattice model 2^n and thus determines the embedding capacity. To embed integer number of secret bits for each cover pixel, we apply the values $n = 2$, 3, and 4 in our experiments. According to Equation (20), EC values are 1, 1.5, and 2, respectively. The average PSNR values of the three image shadows, over the eight test images, are listed in Table 3. As the value n increases, the visual quality of image shadows degrades. Besides, the average PSNR values of three shadows are not the same. The worst case for $n = 2$, 3, and 4 are $\mathbb{S}_2$, $\mathbb{S}_3$, and $\mathbb{S}_2$, respectively. Note that the PSNR value is calculated from the deviation of modified pixel-value. By referring to Table 2, we can obtain the four leading entries that applied in the case of $n = 2$, where the maximum deviation -2 occurs at d_y. That is why the worst PSNR occurs at the shadow $\mathbb{S}_2$. The other cases can be explained in the same way. Recall that the scanning order of candidate elements may alter the queue sequence; thus the resulting list in the lattice model can affect the PSNR relationship of three shadows.

Table 3. The average PSNR (dB) for different ECs (bpp).

EC (bpp)	1 (2^n = 4)	1.5 (2^n = 8)	2 (2^n = 16)
$\mathbb{S}_1$	49.38	44.15	37.71
$\mathbb{S}_2$	47.16	44.62	37.66
$\mathbb{S}_3$	54.15	42.85	39.09

4.2. Comparison with Gao et al.'s Scheme

In this subsection, we compare our scheme with the (2, 3) threshold secret image sharing scheme proposed by Gao et al. [30]. In their scheme, two fractal models sized $2 \times 2 \times 2$ and $3 \times 3 \times 3$ are provided. The two models comprise four and nine embeddable elements, respectively. Fortunately, the volume of our lattice model is adjustable. To make a fair comparison, we set the same volumes and calculate experimental data as listed in Tables 4 and 5.

Table 4. Comparison with Gao et al.'s scheme (model volume 4).

Images	Gao et al.'s Scheme				Proposed Scheme			
	$\mathbb{S}_1$	$\mathbb{S}_2$	$\mathbb{S}_3$	EC (bits)	$\mathbb{S}_1$	$\mathbb{S}_2$	$\mathbb{S}_3$	EC (bits)
Airplane	46.38	44.18	51.13	524289	49.37	47.15	54.18	524289
Boat	46.42	44.18	51.14	524289	49.37	47.15	54.15	524289
Girl	46.36	44.15	51.13	524289	49.37	47.17	54.13	524289
Goldhill	46.37	44.16	51.14	524289	49.38	47.15	54.15	524289
Lena	46.35	44.13	51.13	524289	49.38	47.15	54.16	524289
Lake	46.36	44.16	51.14	524289	49.38	47.16	54.13	524289
Tiffany	46.36	44.18	51.12	524289	49.38	47.17	54.17	524289
Zelda	46.38	44.18	51.13	524289	49.38	47.16	54.13	524289
Average	46.37	44.16	51.13	524289	49.38	47.16	54.15	524289
		47.22				50.23		

Table 5. Comparison with Gao et al.'s scheme (model volume 9).

Images	Gao et al.'s Scheme				Proposed Scheme			
	$\mathbb{S}_1$	$\mathbb{S}_2$	$\mathbb{S}_3$	EC (bits)	$\mathbb{S}_1$	$\mathbb{S}_2$	$\mathbb{S}_3$	EC (bits)
Airplane	38.45	36.76	47	819507	44.15	43.64	43.65	819246
Boat	38.41	36.84	47.02	819025	44.15	43.64	43.64	819157
Girl	38.43	36.86	47.02	819046	44.15	43.64	43.61	819176
Goldhill	38.42	36.87	47.01	819219	44.12	43.64	43.63	819441
Lena	38.42	36.89	47.01	819003	44.15	43.65	43.64	819151
Lake	38.45	36.76	47	819067	44.13	43.63	43.65	819255
Tiffany	38.37	36.9	47.01	819043	44.17	43.63	43.64	819016
Zelda	38.4	36.91	47.01	819386	44.16	43.63	43.63	819120
Average	38.42	36.85	47.01	819162	44.15	43.64	43.64	819195
		40.76				43.81		

As shown in the tables, the visual quality of image shadows produced by our scheme outperforms Gao et al.'s scheme with a gap about 3 dB. The improvement, as expected,

can be explained by referring to Figures 10 and 11, where the embeddable elements are distributed in lumped shapes and in a uniform cylindrical shape, respectively. In addition, the total embedded bits are also listed in the tables. Due to different solutions for the boundary problem of 3D reference matrices, the total payload of the proposed scheme is slightly greater than Gao et al.'s scheme.

4.3. Authentication

In this section, we conduct a series of experiments to verify the applicability of the proposed Algorithms 4 and 5 for authentication, which are based on the integrity check of two shadows and three shadows, respectively. The secret image shadows are generated with a lattice model $\mathcal{M}$ of $2^n = 16$.

Verification of Algorithms 4: A demonstration of two-shadow authentication is given in Figure 15. By using image Boat as the cover image, the shadow generation algorithm produces three image shadows. Suppose we hold a faithful shadow $\mathbb{S}_1$, as shown in Figure 15a, while shadow $\mathbb{S}_2$ has been tampered with a window region replaced by image Cameraman, as shown in Figure 15b. The tamper detection result by applying Algorithm 4 to the shadows $\mathbb{S}_1$ and $\hat{\mathbb{S}}_2$ is displayed in Figure 15c, where black pixels in the window region fail to pass the integrity check. Only a small portion of pixels displayed in white has passed.

(**a**) (**b**) (**c**)

Figure 15. The detection result of Algorithms 4. (**a**) Real: $\mathbb{S}_1$, (**b**) Tampered: $\hat{\mathbb{S}}_2$, (**c**) Detection result.

Verification of Algorithm 5: A demonstration of two-shadow authentication is presented in Figure 16, where real shadows $\mathbb{S}_1$, $\mathbb{S}_2$ and tampered shadow $\hat{\mathbb{S}}_3$ are displayed in Figure 16a–c, respectively. The tamper detection result by applying Algorithms 5 is displayed in Figure 16d.

(**a**) (**b**) (**c**) (**d**)

Figure 16. The detection result of Algorithms 5. (**a**) Real: $\mathbb{S}_1$, (**b**) Real: $\mathbb{S}_2$, (**c**) Tampered: $\hat{\mathbb{S}}_3$, (**d**) Detection result.

The detection rate (DR) to evaluate the performance of integrity check is defined by

$$\mathrm{DR} = \frac{N_D}{N_T}, \tag{21}$$

where N_T denotes the total number of tampered pixels and N_D denotes the number of detected ones. To investigate the performance of our authentication algorithms, the detection rates for the eight cover images are listed in Table 6. The DR value is above 90 percent for the two-shadow version and above 99 percent for the three-shadow version. The high DR value is not surprising, since the embeddable elements just occupy a small portion of the 3D crystal-lattice matrix. As the volume of lattice model increases, the DR value slightly decreases. Nonetheless, in any case, it is almost impossible for a tampered shadow to pass the authentication algorithms.

Table 6. DR values for the two authentication algorithms.

Images	Algorithm 4			Algorithm 5		
	$2^n = 4$	$2^n = 8$	$2^n = 16$	$2^n = 4$	$2^n = 8$	$2^n = 16$
Airplane	0.984	0.968	0.936	0.996	0.996	0.996
Boat	0.963	0.921	0.848	0.99	0.99	0.99
Girl	0.971	0.944	0.881	0.993	0.993	0.993
Goldhill	0.982	0.964	0.926	0.995	0.996	0.996
Lena	0.983	0.967	0.932	0.996	0.996	0.996
Lake	0.993	0.988	0.975	0.999	0.999	0.999
Tiffany	0.968	0.934	0.882	0.993	0.992	0.991
Zelda	0.977	0.953	0.899	0.995	0.996	0.993
Average	0.977	0.953	0.906	0.994	0.994	0.994

4.4. Comparison with Other Related Schemes

In this section, we compare the features of the proposed scheme with other different secret image sharing schemes, including Chang et al.'s scheme in 2014 [10], Chang et al.'s scheme in 2020 [28], and Li et al.'s scheme [29]. As shown in Table 7, the schemes proposed in [10] and [28] use multiple cover images, and these cover images cannot be recovered after extracting secret data. While the proposed scheme and Li et al.'s scheme [29] use a single cover image to generate multiple image shares, the cover image can be recovered by the recipient. Besides, our new scheme provides two versions of authentication. When the third faithful shadow is available, the detection rate of our scheme is the highest among all. Even if one of the three shadows is not available, the proposed scheme can still reach a cheating detection ratio of 95 percent.

Table 7. Comparison of our method with some methods.

Features	[10]	[28]	[29]	Proposed
Reversibility	No	No	Yes	Yes
Multiple cover images	Yes	Yes	No	No
(k, n)- SIS	(2, 2)	(2, 2)	(3, 3)	(2, 3)
Average authentication ability	0.5	0.43	0.98	0.95/0.99

4.5. Time Efficiency

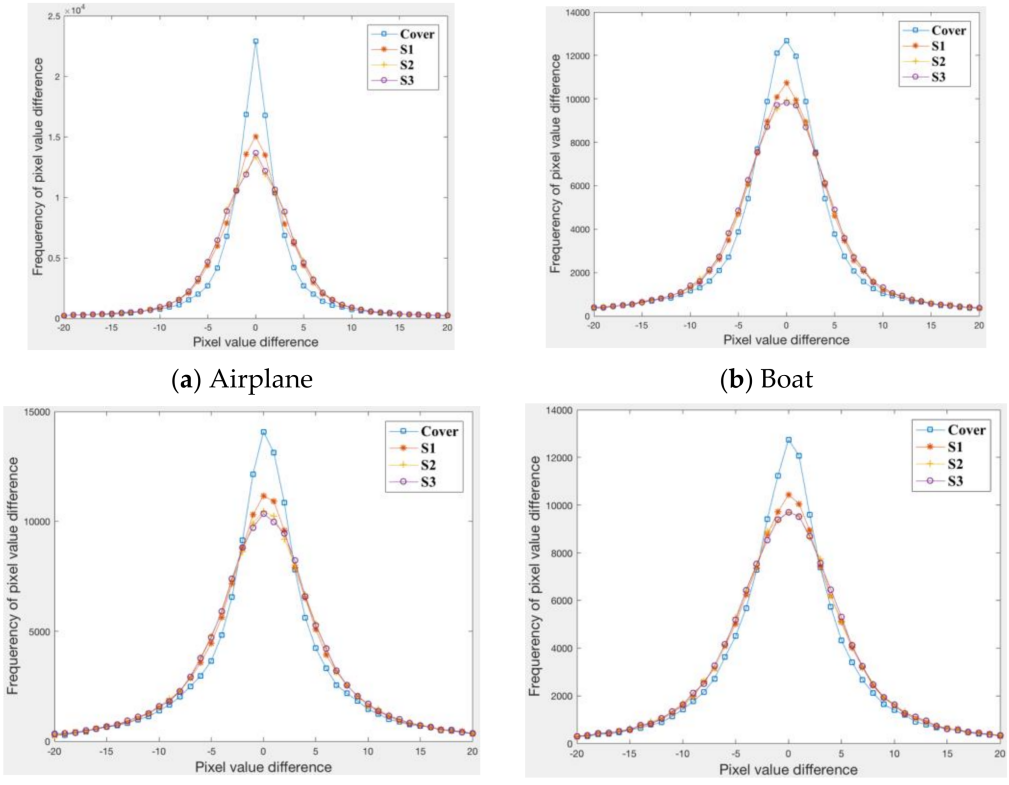
Table 8 shows the execution time for the embedding and extracting phases of our scheme when P is set as 4 and 9. The execution time is less than 0.2 s for the embedding phase and less than 0.4 s for the extraction phase. We can conclude that the proposed scheme is computationally efficient and suitable for real time applications.

Table 8. Execution time (s) of the proposed scheme.

Images	$P = 4$		$P = 9$	
	Embedding	**Extracting**	**Embedding**	**Extracting**
Airplane	0.17	0.38	0.18	0.37
Boat	0.16	0.38	0.16	0.35
Girl	0.16	0.37	0.17	0.37
Goldhill	0.17	0.38	0.18	0.38
Lena	0.16	0.36	0.16	0.36
Lake	0.17	0.38	0.17	0.38
Tiffany	0.16	0.38	0.17	0.35
Zelda	0.16	0.37	0.16	0.35
Average	0.16	0.38	0.17	0.36

4.6. PDH Analysis

The pixel-value differencing histogram (PDH) is a histogram which is constructed based on the frequency of the difference between every two adjacent elements in an image. For a natural image, the PDH should exhibit a peak at the zero-difference value and gradually descend outward as the blue curves, as shown in Figure 17. The PDH of four cover images together with their corresponding shadows are plotted in Figure 17, where the high embedding mode of $P = 9$ is applied. Obviously, the normal PDH shape of a natural image is well preserved for all the image shadows.



(**a**) Airplane (**b**) Boat

(**c**) Girl (**d**) Goldhill

Figure 17. PDH diagrams of four applied cover images with their shadows.

5. Conclusions

This paper uses a crystal-lattice matrix to improve the visual quality of image shadows of the (2, 3) threshold RSIS scheme. A greedy algorithm is proposed to automatically generate the lattice modFel, which is the fundamental unit of the crystal-lattice matrix. The

volume of the lattice model is adjustable to meet the desired embedding capacity. In addition, two authentication algorithms are devised based on the data integrity. Experimental results demonstrate the applicability of the proposed scheme. Besides, the visual quality of image shadows is significantly improved, as expected.

The (k, n)-threshold RSIS schemes with $k < n$ is a novel frame structure that is more flexible in application than the conventional approach of (n, n)-threshold RSIS. We will try to find better solutions to improve the overall performance of the secret image sharing scheme.

Author Contributions: Data curation, J.-Y.L.; software, J.-Y.L.; formal analysis, J.-H.H.; funding acquisition, J.-H.H.; investigation, J.-H.H.; methodology, J.-H.H.; writing—original draft, J.-Y.L.; writing—review and editing, J.-H.H.; project administration, C.-C.C.; supervision, C.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: The authors thank the Ministry of Science and Technology of Taiwan for its sponsorship to this research (Grant#: MOST 110-2221-E-507-003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chan, C.K.; Cheng, L.M. Hiding data in images by simple LSB substitution. *Pattern Recognit.* **2004**, *37*, 469–474. [CrossRef]
2. Mielikainen, J. LSB matching revisited. *IEEE Signal Process. Lett.* **2006**, *13*, 285–287. [CrossRef]
3. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circ. Syst. Video* **2003**, *13*, 890–896. [CrossRef]
4. Alattar, A.M. Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156. [CrossRef] [PubMed]
5. Ni, Z. Reversible data hiding. *IEEE Trans. Circ. Syst. Video* **2006**, *16*, 354–362.
6. Qin, C.; Chang, C.C.; Huang, Y.-H.; Liao, L.-T. An Inpainting-Assisted Reversible Steganographic Scheme Using a Histogram Shifting Mechanism. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 1109–1118. [CrossRef]
7. Zhang, X.; Wang, S. Efficient Steganographic Embedding by Exploiting Modification Direction. *IEEE Commun. Lett.* **2006**, *10*, 781–783. [CrossRef]
8. Kim, H.J. Improved modification direction methods. *Comput. Math. Appl.* **2010**, *60*, 319–325. [CrossRef]
9. Hong, W.; Chen, T.-S.; Shiu, C.-W. A Minimal Euclidean Distance Searching Technique for Sudoku Steganography. In Proceedings of the 2008 International Symposium on Information Science and Engineering, Shanghai, China, 20–22 December 2008; Volume 1, pp. 515–518. [CrossRef]
10. Chang, C.C.; Liu, Y.; Nguyen, T.S. A novel turtle shell based scheme for data hiding. In Proceedings of the Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kitakyushu, Japan, 27–29 August 2014.
11. Wu, D.-C.; Tsai, W.-H. A steganographic method for images by pixel-value differencing. *Pattern Recognit. Lett.* **2013**, *24*, 1613–1626. [CrossRef]
12. Chen, J. A PVD-based data hiding scheme with histogram preserving using pixel pair matching. *Signal Process. Image. Commun.* **2014**, *29*, 375–384. [CrossRef]
13. Naor, M.; Shamir, A. Visual cryptography. In *1994: Workshop on the Theory and Application of Cryptographic Techniques*; Lofthus Norway; Springer: Berlin/Heidelberg, Germany, 1994.
14. Fang, W.P. Non-expansion visual secret sharing in reversible style. *Int. J. Univers. Comput. Sci. Netw. Secur.* **2009**, *9*, 204–208.
15. Tsai, D.S.; Chen, T.-H.; Horng, G. On generating meaningful shares in visual secret sharing scheme. *Imaging Sci. J.* **2008**, *56*, 49–55. [CrossRef]
16. Shyu, S.J.; Chen, M.C. Optimum Pixel Expansions for Threshold Visual Secret Sharing Schemes. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 960–969. [CrossRef]
17. Harn, L.; Xia, Z.; Hsu, C.; Liu, Y. Secret sharing with secure secret reconstruction. *Inf. Sci.* **2020**, *519*, 1–8. [CrossRef]
18. Chang, C.C.; Kieu, T.D.; Chou, Y.-C. Reversible data hiding scheme using two steganographic images. In Proceedings of the TENCON 2007-2007 IEEE Region 10 Conference, Taipei, Taiwan, 30 October–2 November 2007; pp. 1–4. [CrossRef]
19. Chang, C.C.; Chou, Y.C.; Kieu, D.T. Information hiding in dual images with reversibility. In Proceedings of the Third International Conference on Multimedia and Ubiquitous Engineering, Qingdao, China, 4–6 June 2009.
20. Chen, X.; Hong, C. An Efficient Dual-image Reversible Data Hiding Scheme Based on Exploiting Modification Direction. *J. Inf. Secur. Appl.* **2021**, *58*, 102702. [CrossRef]

21. Lee, C.-F.; Huang, Y.-L. Reversible data hiding scheme based on dual stegano-images using orientation combinations. *Telecommun. Syst.* **2013**, *52*, 2237–2247. [CrossRef]
22. Karati, S.; Das, A. Faster Batch Verification of Standard ECDSA Signatures Using Summation Polynomials. In Proceedings of the International Conference on Applied Cryptography and Network Security, Lausanne, Switzerland, 10–13 June 2014; Springer: Cham, Switzerland, 2014; Volume 8479, pp. 438–456. [CrossRef]
23. Semaev, I.A. Summation polynomials and the discrete logarithm problem on elliptic curves. *IACR Cryptol. ePrint Arch.* **2004**, *2004*, 31.
24. Xiong, H. On the Design of Blockchain-based ECDSA with Fault-tolerant Batch Verication Protocol for Blockchain-enabled IoMT. *IEEE J. Biomed. Health Informaties* **2021**. [CrossRef] [PubMed]
25. 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]
26. Liu, Y.; Chang, C.C. A turtle shell-based visual secret sharing scheme with reversibility and authentication. *Multimedia Tools Appl.* **2018**, *77*, 25295–25310. [CrossRef]
27. Lin, J.-Y.; Chen, Y.; Chang, C.C.; Hu, Y.C. Dual-image-based reversible data hiding scheme with integrity verification using exploiting modification direction. *Multimedia Tools Appl.* **2019**, *78*, 25855–25872. [CrossRef]
28. Chang, C.C.; Horng, J.H.; Shih, C.S.; Chang, C.C. A maze matrix-based secret image sharing scheme with cheater detection. *Sensors.* **2020**, *20*, 3802. [CrossRef]
29. Li, X.-S.; Chang, C.C.; He, M.-X.; Lin, C.-C. A lightweight authenticable visual secret sharing scheme based on turtle shell structure matrix. *Multimedia Tools Appl.* **2020**, *79*, 453–476. [CrossRef]
30. Gao, K.; Horng, J.-H.; Chang, C.C. A Novel (2, 3) Reversible Secret Image Sharing Based on Fractal Matrix. *IEEE Access* **2020**, *8*, 174325–174341. [CrossRef]