mathematics

MDPI

Article

XOR-Based Progressively Secret Image Sharing

Cheng-Shian Lin, Chien-Chang Chen * and Yu-Cheng Chen

Department of Computer Science and Information Engineering, Tamkang University, New Taipei City 25137, Taiwan; 157446@mail.tku.edu.tw (C.-S.L.); a22781911@gmail.com (Y.-C.C.)
* Correspondence: ccchen34@mail.tku.edu.tw

Abstract: Secret image sharing technology is a strategy for jointly protecting secret images. The (n, n) secret image sharing problem can be solved by conventional Boolean calculation easily. However, how to recover secret images with progressive steps is not addressed. In this study, we proposed an XOR-based (m, t, T_i) multi-secret image sharing scheme that shares m secret images among m participants and recovers m shared images progressively with t thresholds. The proposed secret images partition strategy (SIPS) partitions m secret images to generate intermediate images for different thresholds in the sharing procedure. Based on progressive recovery property, the proposed recovery method recovers parts of the secret images by gathering consecutive shared images. Moreover, gathering all shared images can perfectly recover all secret images. The experimental results show that the proposed XOR-based multi-secret image sharing method has high security and efficiency.

Keywords: secret image sharing; XOR-based; progressive recovery

check for updates

Citation: Lin, C.-S.; Chen, C.-C.; Chen, Y.-C. XOR-Based Progressively Secret Image Sharing. *Mathematics* **2021**, *9*, 612. https://doi.org/10.3390/math9060612

Academic Editor: Ioana Boureanu

Received: 30 January 2021
Accepted: 10 March 2021
Published: 12 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the development of digital multimedia and the Internet, the sharing of multimedia content through the Internet has become more popular, and intellectual property rights and personal privacy have also become increasingly important. The secret image sharing technique is one of the protection methods that aim to protect multimedia content and has become an important field of recent research [1].

Thien and Lin [2] first proposed a secret image sharing scheme based on the Shamir-Lagrange method [3,4], and then a number of functional secret image sharing schemes have been proposed, such as sharing a secret image among host images [5,6], sharing using Boolean operations [7,8], progressive sharing [9–11], visual cryptography with secret image sharing [12], and scalable sharing [13]. Since most of the approaches perform only for sharing one secret image, they cannot share multiple secret images simultaneously in real circumstances.

Over the past few years, a number of approaches for multi-secret image sharing have been proposed. Chen and Chien [14] proposed sharing numerous images secretly with the reduced possessing load. The concept of the method is to generate a random number image with the same size as the secret image, and the random number image and the secret image are processed separately, and then the calculation is made public. Then use secret image sharing technology to share images containing random noise with all participants. Chen and Wu [15] proposed an efficient multi-secret image sharing method that used $n - 1$ secret images to generate n shared images based on Boolean operations. However, it is necessary to collect all the shared images to recover the secret images. Chen and Wu [16] improved the multi-secret image sharing method proposed in Chen and Wu [15] and proposed a random number image generation function, which used secret images or shared images to perform XOR operations and bit-shifting operations to obtain the random image. Since the random image does not need to be shared as shared images, the sharing capacity is increased. Chen et al. [17] proposed a method that also used the random number image generation function to generate images containing random noise but will perform

further bit shift processing on the generated random images. Kabirirad and Eslami [18] proposed a multi secret image sharing scheme based on Boolean operations. Although their approach is efficient, it requires that the size of the generated shared image must be the same as the secret image. Azza and Lian [19] present a cellular automaton-based multi-secret image sharing scheme with steganography. Chen and Wu [20] proposed a Boolean-based multi-secret sharing scheme with the general access structure. The proposed scheme does not necessarily collect all the shared images to recover the secret images by the predefined access structure. In addition, since image secret sharing schemes are widely used in watermarking, access control, identity authentication and so on, a number of image secret sharing schemes were proposed and applied in various practical applications, such as biometric privacy [21], share authentication [22,23], and e-voting [24]. Notice that most of the previous schemes are mainly for low computational complexity. How to process the secret images to achieve progressively secret image sharing and recovery is still not solved by them [25–31].

This paper proposed a secure XOR-based (m, t, T_i) multiple secret image sharing approach that shares m secret images among m participants and recovers m shared images progressively with t thresholds. The proposed method uses the random number image generation function to generate images containing random noise and further process the secret images to achieve progressively secret image sharing and recovery. The experimental results show the novel properties of our proposed approach.

The paper is organized into the following sections. Section 2 gives a brief review of related works [15–17]. Section 3 presents details of the proposed sharing and recovery approach. Section 4 presents the experimental results. Section 5 offers concluding remarks.

2. Review of Related Literature

This section briefly reviews the state-of-art in secret sharing of multiple images based on Boolean function. Section 2.1 introduces the secret sharing of multiple images based on the Boolean function proposed by Chen and Wu [15]. Section 2.2 introduces the method proposed by Chen and Wu [16]. Section 2.3 introduces the method proposed by Chen et al. [17].

2.1. Boolean Based Scheme of Chen and Wu

Chen and Wu [15] proposed a method of secretly sharing and restoring $n-1$ secret images through n shared images. The sharing steps are as follows:

1. Select $n-1$ secret images and a random number image R of the same size. The secret images are marked as G_i ($i = 1, \ldots, n-1$) in order;
2. Using Equation (1), perform an XOR operation on each of $n-1$ secret images and R to obtain noise images of $n-1$ secret images, labeled B_i ($i = 1, \ldots, n-1$);

$$B_i = G_i \oplus R \tag{1}$$

3. Use Equation (2) to generate shared images, labeled S_i ($i = 1, \ldots, n$);

$$\begin{cases} S_1 = B_1 \\ S_i = B_{i-1} \oplus B_i, \ for\ 2 \le i \le n-1 \\ S_n = B_{n-1} \oplus G_1 \end{cases} \tag{2}$$

The recovery steps are as follows:

1. Collect all n shared images, and perform XOR operation $S_1 \oplus S_2 \oplus \ldots \oplus S_n$ on all shared images to restore G_1. Moreover, restore the random number image R through $S_1 \oplus G_1$;
2. Use Equation (3) to recover $n-1$ noise images;

$$\begin{cases} B_1 = S_1 \\ B_k = S_k \oplus B_{k-1}, \ for\ 2 \le k \le n-1 \end{cases} \tag{3}$$

3. Using Equation (4), use $n-1$ noise images and images containing random noise R to restore all secret images.

$$G_i = B_i \oplus R \tag{4}$$

The above is the introduction of the Chen and Wu [15] method. If the number of shared secret images divided by the number of shared images is defined as the sharing rate, the sharing rate is $n-1/n$. However, a shared image S_i is calculated by Equation (5), and it is quite insufficient in security, as shown in Figure 1.

$$B_i \oplus B_{i-1} = (G_{i-1} \oplus R) \oplus (G_{i-1} \oplus R) = G_i \oplus G_{i-1} \tag{5}$$

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

Figure 1. The results of XOR operation: (**a**) Lena, (**b**) house, (**c**) XOR result of (**a**,**b**).

2.2. The Improved Scheme of Chen and Wu

Chen and Wu [16] improved their work [15] by using the proposed random number image generation function F to generate a random number image R. F contains two sub-functions F_1 and F_2. The function is defined as $F(x)= F_2\ (F_1(x))$, where F_1 is calculated by Equation (6), performs XOR operation on all images, F_2 then marks the image after F_1 operation as $\overline{G}$. For each pixel value, it is converted into a binary bit, and the bit is transposed, and the processed result is marked as $\overline{G'}$ and expressed by Equation (7).

$$\overline{G} = F_1(G_1, G_2, \ldots, G_k) = G_1 \oplus G_2 \oplus \ldots \oplus G_k \tag{6}$$

$$\overline{G'} = F_2\left(\overline{G}_{x,y}(b)\right) = \overline{G}_{x,y}(7-b) = \overline{G'}_{x,y}(b) \tag{7}$$

where x, y represent the pixel position in the image, and b ($0 \le b \le 7$) represents the pixel bit converted to binary. The sharing steps are introduced as follows:

1. Select n secret images and mark them as G_1, G_2, ... , G_n in sequence;
2. Using Equation (8), the random number image generation function obtains the random number image R;

$$\begin{aligned} R &= F(G_1, G_2, \ldots, G_k) \\ &= F_2(F_1(G_1, G_2, \ldots, G_k)) \\ &= F_2(G_1 \oplus G_2 \oplus \ldots \oplus G_k), where\ k = 2 \cdot \tfrac{n}{2} \end{aligned} \tag{8}$$

3. Using Equation (9), n secret images and images containing random noise R are used to generate n noise images, which are labeled N_i (i = 1, 2, ... , n);

$$N_i = G_i \oplus R \tag{9}$$

4. Using Equation (10), generate n shared images and share them with all participants, labeled S_i ($i = 1, \ldots, n$).

$$S_i = \begin{cases} N_1, \ i = 1 \\ N_2, \ i = 2 \\ N_i \oplus N_{i-1} \oplus N_{i-2}, \ for \ 3 \leq i \leq n \end{cases} \tag{10}$$

The recovery procedure is described as follows:

1. Using Equation (11), from the shared images $S_1, S_2, S_3, \ldots, S_n$, restore all the noise images $N_1, N_2, \ldots, N_n$ in sequence;

$$N_i = \begin{cases} S_1, \ i = 1 \\ S_2, \ i = 2 \\ S_i \oplus N_{i-1} \oplus N_{i-2}, \ for \ 3 \leq i \leq n \end{cases} \tag{11}$$

2. Using Equation (12), the $\overline{G}$ generated when sharing is obtained from the noise image. Then use Equation (7) to substitute $\overline{G}$ into the sub-function F_2 of the random number, the image generating function, and the random number image R generated during sharing can be recovered;

$$\overline{G} = (N_1 \oplus N_2 \oplus \ldots \oplus N_k) \tag{12}$$

3. According to Equation (13), the noise image and random number image R are recovered to the corresponding secret image.

$$G_i = N_i \oplus R \tag{13}$$

2.3. Scheme of Chen et al.

The sharing steps of the secret image sharing scheme proposed by Chen et al. are introduced as follows:

1. Prepare n secret images and mark them as I_i ($i = 0, \ldots, n-1$) in sequence;
2. After performing the XOR operation on all the secret images I, the Hash method is performed on the result to generate the matrix h;

$$h = H(I_0 \oplus I_1 \oplus \cdots \oplus I_{n-1}) \tag{14}$$

3. The matrix h is reorganized to generate the image SI;

$$SI = image\_synthesis(h) \tag{15}$$

4. Performing XOR operation on all secret images I_0–I_{n-1} and then bit shifting to generate the image and then performing XOR operation with the SI to generate random image R;

$$R = bit\_reverse(I_0 \oplus I_1 \oplus \cdots \oplus I_{n-1}) \oplus SI \tag{16}$$

5. Using Equation (17), n random images R_0–R_{n-1} are generated by n secret images performing bit shifting, where x, y represents the pixel positions in the image;

$$R_i(x, y) = R(x - i, y - i) \tag{17}$$

6. After all the secret images I_0–I_{n-1} and images containing random noise R_0–R_{n-1} are calculated according to Equation (18), n shared images O_i ($i = 0, \ldots, n-1$) can be obtained.

$$O_i = \begin{cases} I_i \oplus R_i \oplus R_{i+1}, \ i \neq n-1 \\ I_i \oplus R_i \oplus R_0, \ others \end{cases} \tag{18}$$

The recovery procedure is described as follows:

1. Collect all n shared images, and perform XOR operation on all shared images $O_0 \oplus O_1 \oplus \cdots \oplus O_{n-1}$. In this way, the result of $I_0 \oplus I_1 \oplus \cdots \oplus I_{n-1}$ is obtained;
2. Obtain h by Equation (14), and then obtain SI by Equation (15);
3. After obtaining the noise image R by Equation (16), then obtain the noise image $R_0, \cdots, R_{n-1}$ by Equation (17);
4. Then, by Equation (19), n noise images $R_0, \ldots, R_{n-1}$ are used to restore all the secret images.

$$I_i = \begin{cases} O_i \oplus R_i \oplus R_{i+1}, \ i \neq n-1 \\ O_i \oplus R_i \oplus R_0, others \end{cases} \quad (19)$$

Based on Chen et al.'s proposed algorithm [17], the sharing rate of this method is n/n.

3. The Proposed Approach

This section introduces our proposed XOR-based (m, t, T_i) multi-secret image sharing scheme, in which m secret images are shared among m participants with t different thresholds T_0 to T_{t-1} satisfying $T_0 < \ldots < T_{t-1}$. Moreover, the highest threshold T_{t-1} should be equal to m.

Section 3.1 introduces the proposed secret image segmentation strategy (*SIPS*) method to partition m secret images into t groups of intermediate images for t thresholds. Section 3.2 introduces the proposed sharing algorithm, and Section 3.3 introduces the recovery algorithm. Examples of sharing and recovery with thresholds (4, 3, {2, 3, 4}) are demonstrated in Section 3.4.

3.1. Secret Images Partition Strategy (SIPS)

Since the proposed scheme obtains the progressive recovery property, the following partition method is, therefore, presented to generate intermediate images for different thresholds. For sharing m secret images with the property of t thresholds, the first step in *SIPS* partitions the lowest 4 bit-planes (LSB0–3) of m secret images, which denoted by PS_{t-1} as group partition and the remaining highest 4 bit-planes (LSB4–7) are partitioning to PS_0–PS_{t-2} by bit sequence. Since any modification under LSB2 is hard to be noticed by human eyes, therefore, lowest and highest 4 bit-planes are determined. Figure 2 shows two examples of sharing a 5 pixels block to PS_i $(0 \leq i \leq t-1)$ two different thresholds of (a) $t = 3$ and (b) $t = 4$ by the proposed *SIPS*.

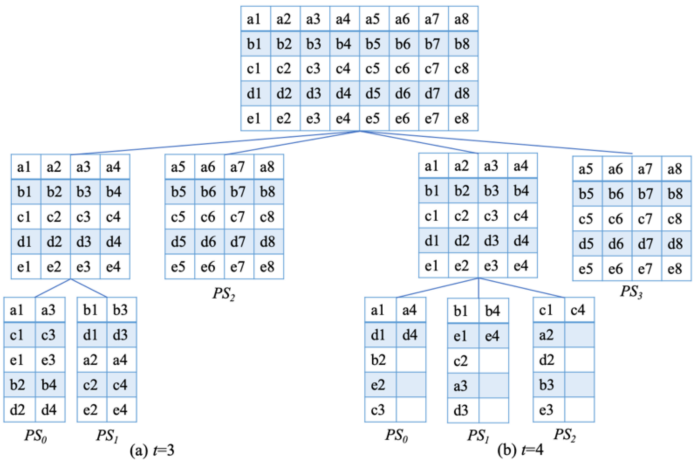

Figure 2. Example of sharing 5 pixels by the proposed secret image partition strategy (*SIPS*), (**a**) shares to 3 groups of PS_0, PS_1, PS_2, (**b**) shares to 4 groups of PS_0, PS_1, PS_2, PS_3.

Figure 3 shows the results of only part PS_i being recovered by filling the un-recovered PS_i by random or 0/1 bits under threshold $t = 3$. Figure 3a shows the original Barbara image, and Figure 3b–d shows the recovery by PS_0 with compensating by 0/1 and random bit, respectively. Moreover, Figure 3e–g shows the recovery by $\{PS_0, PS_1\}$ with compensating

by 0/1 and random bit, respectively. Figure 3 shows that compensating by 0 leads to a darker recovery. On the contrary, compensating by 1 acquires a brighter recovery. However, since the compensation by 0/1 mainly changes the brightness of the recovery result, the proposed strategy adopting the compensation by random bits, as shown in Figure 3d,g.

Figure 3. Results of an image by part of group partitions under threshold $t = 3$, (**a**) the original image: Barbara, (**b**–**d**) recovery by PS_0 with compensating by 0/1 and random bit, respectively, (**e**–**g**) recovery by $\{PS_0, PS_1\}$ with compensating by 0/1 and random bit, respectively.

Since each secret image has its group partitions $PS_i (0 \leq i \leq t-1)$, sharing m secret images acquires m multiples of their PS_i. The second step in *SPIS* is to generate intermediate images from PS_i. The lowest 4 bit-planes (LSB0–3) of PS_{t-1} construct the intermediate images $P_{t-1,j}$ $(0 \leq j \leq T_{t-1})$ for group $t-1$. The 4 bits higher planes (LSB4–7) of PS_0–PS_{t-2} are shared to $t-1$ groups, as shown in Figure 2, and the size of these intermediate images will reduce to $\frac{1}{2(t-1)}$ of the original image size. Each $PS_i (0 \leq i < t-1)$ is then reorganized to intermediate images $P_{i,k}$ $(0 \leq k \leq T_{i-1})$ for further sharing processing.

By applying the proposed *SIPS* and the predefined (m, t, T_i) thresholds, secret images with size $M \times N$ obtains the following intermediate images.

1. 4 lowest bit-planes (LSB0–3) of m secret images ($4mMN$ bits) are reconstructed to m intermediate images with the size of $\frac{M}{2} \times N$ pixels for threshold T_{t-1} (=m);
2. Total $4mMN/(t-1)$ bits from m secret images in highest bit-planes (LSB4–7) are reshaped to T_i ($0 \leq i < t-1$) intermediate images for different T_i thresholds, respectively. Therefore, sizes of intermediate images are $\frac{mM}{(2(t-1)T_i)} \times N$;
3. Finally, the size of the shared image is $\left(\sum_{i=0}^{t-1} \frac{m}{(2(t-1)T_i)} + \frac{1}{2}\right) \times MN$

Table 1 shows the sizes of shared images in the proposed *SIPS* with different thresholds under the size of secret images being 300 × 300.

Table 1. Sizes of shared images in the proposed secret image partition strategy (*SIPS*).

Thresholds	(3, 2, {2, 3})		(4, 2, {2, 4})		(4, 3, {2, 3, 4})	
	Threshold	Intermediate Image Size	Threshold	Intermediate Image Size	Threshold	Intermediate Image Size
Group 0	2	225 × 300	2	300 × 300	2	150 × 300
Group 1	3	150 × 300	4	150 × 300	3	100 × 300
Group 2					4	150 × 300

3.2. The Proposed Sharing Algorithm

The sharing algorithm of the proposed XOR-based (m, t, T_i) multiple secret image sharing scheme shares m secret images among m participants with the progressive property of t different thresholds T_i ($0 \leq i \leq t-1$) and $m = T_{t-1}$. These t thresholds must fit the requirement of $T_0 < T_1 < \cdots < T_{t-1}$. Figure 4 shows the proposed sharing structure. The intermediate images $P_{j,k}$ ($0 \leq j \leq t-1$, $0 \leq k < T_j$) are acquired from the proposed *SIPS* introduced in Section 3.1. The following steps adopt Chen et al.'s method [17] with consecutive modification on thresholds T_i, $0 \leq i < t-1$. This modification also means that in the $P_{t-1,k}$ intermediate images, the threshold is $T_{t-1,}$ and Chen et al.'s method [17] is directly applied. However, other thresholds T_i ($0 \leq i < t-1$) should give some modification to Chen et al.'s method [17]. Our proposed structure obtains the property of consecutive recovery on threshold T_i ($0 \leq i < t-1$). For example, if $T_i = 3$ and $m = 4$, four shared images (denoted by S_1, S_2, S_3, S_4) are generated and collecting 3 consecutive shared images like S_1, S_2, S_3 or S_2, S_3, S_4 can perfectly recover secret images of the threshold $T_i = 3$.

The sharing algorithm is introduced as follows:

1. Using the proposed secret images partition strategy (*SIPS*) to generate intermediate images $P_{j,k}$ ($0 \leq j \leq t-1$, $0 \leq k < T_j$) from m secret images;
2. Apply the XOR function, the SHA-256 hash function H, the *image_synthesis*(h) function, and the *bit_reverse*(x) function on these intermediate images $P_{j,k}$ by Equations (20) and (21) to generate random image R_j for each threshold;

$$X_j = P_{j,0} \oplus P_{j,1} \oplus \cdots \oplus P_{j,T_i-1} \tag{20}$$

$$R_j = image\_synthesis\left(H\left(X_j\right)\right) \oplus bit\_reverse\left(X_j\right) \tag{21}$$

   where the *image_synthesis*(h) function synthesizes the image from the seed h, and the *bit_reverse*(X) function reverses each bit of all pixels in an image X;
3. Generate a series of random images $R_{j,k}(0 \leq k \leq T_i - 1)$ from circular pixel shift processing on R_j as defined in Equation (22);

$$R_{j,k}(x,y) = R_j(x-k, y-k) \tag{22}$$

4. Generate a series of intermediate shares $S_{i,j}$ by Equation (23);

$$S_{i,j} = P_{j,(i\%\ T_j)} \oplus R_{j,(i\%\ T_j)} \oplus R_{j,((i+1)\%\ T_j)} \tag{23}$$

5. Concatenate all the intermediate shares $S_{i,j}$ to acquire shared image S_i for participant i by Equation (24).

$$S_i = S_{i,0}||S_{i,1}||\cdots||S_{i,t-1} \tag{24}$$

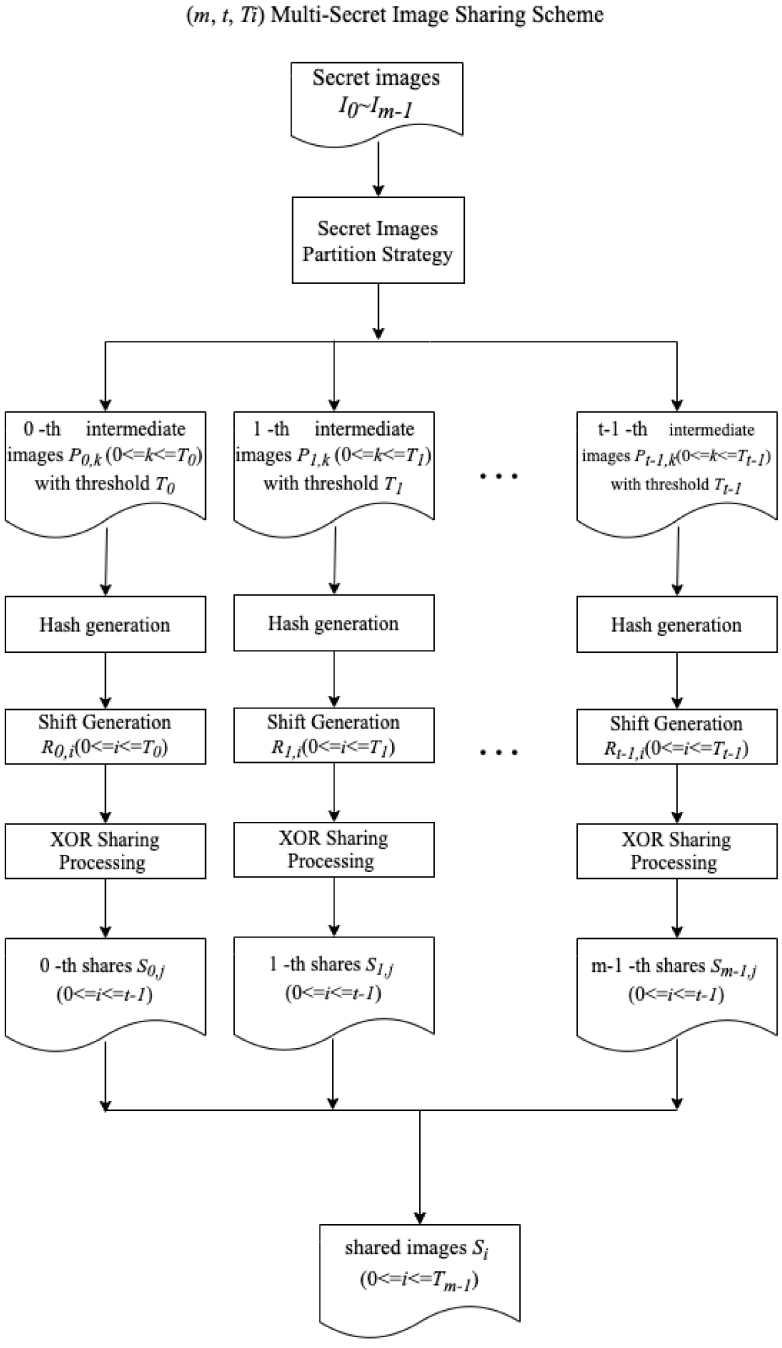

Figure 4. The proposed sharing algorithm.

3.3. The Proposed Recovery Algorithm

The recovery algorithm of the proposed XOR-based multiple secret image sharing scheme is introduced in this section. Enough shared images should be acquired for meeting the threshold criterium. In the proposed XOR-based (m, t, T_i) multiple secret image sharing scheme, the acquirement of consecutive k shared images recovers all thresholds T_i with $T_i \leq k$. The recovery algorithm is introduced as follows:

1. Acquire the consecutive k shared images. Without loss of generality, these shared images are denoted by $S_j (0 \leq j \leq k-1)$;
2. Split S_j for obtaining intermediate shares $S_{j,i} (0 \leq j \leq k-1,\ 0 \leq i \leq t-1)$;
3. Assume the number of collected shared images k is equal to one threshold T_m $(m \leq t-1)$. For each threshold $T_i (0 \leq i \leq m)$, apply the following steps to recover the intermediate image $P_{j,i}$;
   - 3.1. Applying $S_{j,i} (0 \leq j \leq T_i - 1)$ to Equations (20) and (21) acquire X_j and R_j, respectively;
   - 3.2. Generate a series of random images $R_{j,i} (0 \leq j \leq T_i - 1)$ from R_j;
   - 3.3. Acquire intermediate images $P_{j,i}$ from Equation (25)

$$P_{j,(i\%\ T_j)} = S_{i,j} \oplus R_{j,(i\%\ T_j)} \oplus R_{j,((i+1)\%\ T_j)}; \tag{25}$$

4. Apply the reverse processing of the proposed *SIPS* from the intermediate images $P_{j,i}$ to recover the bit-plane shares PS_i $(0 \leq i \leq t-1)$;
5. Combine all the recovered bit-plane shares PS_i $(0 \leq i \leq t-1)$ and replace the unrecovered bits with random bits for recovering all secret images.

3.4. Discussion of a Sharing Example

This section shows an example of sharing four secret images I_0–I_3 with size 300×300 by the proposed (4, 3, {2, 3, 4}) XOR-based multi-secret image sharing scheme. The detailed information for each step is introduced as follows:

1. Partition four secret images I_0–I_3 to group partition;
   - 1.1. Four highest bit planes of size 150×300 are partitioned to group partitions $PS_{0,i}$ $(0 \leq i \leq 3)$ and $PS_{1,i}$ $(0 \leq i \leq 3)$ with size 75×300;
   - 1.2. Four lowest bit planes construct the $PS_{2,i}$ $(0 \leq i \leq 3)$ with size 150×300;
2. Acquire the intermediate images $P_{i,j}$ $(0 \leq i \leq 3, 0 \leq j \leq 2)$ from PS_i $(0 \leq i \leq 2)$;
   - 2.1. Re-shape $PS_{0,i}$ $(0 \leq i \leq 3)$ to acquire $P_{0,0}$, $P_{0,1}$ with size 150×300 for sharing with threshold 2;
   - 2.2. Re-shape $PS_{1,i}$ $(0 \leq i \leq 3)$ to acquire $P_{1,0}$, $P_{1,1}$, $P_{1,2}$ with size 100×300 for sharing with threshold 3;
   - 2.3. Re-shape 4 lower bit planes $PS_{2,i}$ $(0 \leq i \leq 3)$ to acquire $P_{j,2}$ $(0 \leq j \leq 3)$ with size 150×300 for sharing with threshold 4;
3. In first threshold 2, R_0 is acquired by applying $P_{0,0}$, $P_{0,1}$ to Equation (20). Applying R_0 and Equation (21) generates $R_{0,0}$ and $R_{0,1}$. Four intermediate shares with the size of 150×300 are acquired from the following:

$$\begin{aligned} S_{0,0} &= P_{0,0} \oplus R_{0,0} \oplus R_{0,1} \\ S_{1,0} &= P_{0,1} \oplus R_{0,1} \oplus R_{0,0} \\ S_{2,0} &= P_{0,0} \oplus R_{0,0} \oplus R_{0,1} \\ S_{3,0} &= P_{0,1} \oplus R_{0,1} \oplus R_{0,0} \end{aligned}$$

4. In second threshold 3, R_1 is acquired by applying $P_{1,0}$, $P_{1,1}$, $P_{1,2}$ to Equation (20). Applying R_1 and Equation (21) generates $R_{1,0}$, $R_{1,1}$ and $R_{1,2}$. Four intermediate shares with the size of 100×300 are acquired from the following:

$$\begin{aligned} S_{0,1} &= P_{1,0} \oplus R_{1,0} \oplus R_{1,1} \\ S_{1,1} &= P_{1,1} \oplus R_{1,1} \oplus R_{1,2} \\ S_{2,1} &= P_{1,2} \oplus R_{1,2} \oplus R_{1,0} \\ S_{3,1} &= P_{1,0} \oplus R_{1,0} \oplus R_{1,1} \end{aligned}$$

5. In the last threshold 4, R_2 is acquired by applying $P_{2,0}$, $P_{2,1}$, $P_{2,2}$, $P_{2,3}$ to Equation (20). Applying R_2 and Equation (21) generates $R_{2,0}$, $R_{2,1}$, $R_{2,2}$ and $R_{2,3}$. Four intermediate shares with the size of 150 × 300 are acquired from the following:

$$\begin{aligned} S_{0,2} &= P_{2,0} \oplus R_{2,0} \oplus R_{2,1} \\ S_{1,2} &= P_{2,1} \oplus R_{2,1} \oplus R_{2,2} \\ S_{2,2} &= P_{2,2} \oplus R_{2,2} \oplus R_{2,3} \\ S_{3,2} &= P_{2,3} \oplus R_{2,3} \oplus R_{2,0} \end{aligned}$$

6. All shared images S_0, S_1, S_2, and S_3 with the size of 400 × 300 are obtained from the following concatenation processing:

$$\begin{aligned} S_0 &= S_{0,0} \parallel S_{0,1} \parallel S_{0,2} \\ S_1 &= S_{1,0} \parallel S_{1,1} \parallel S_{1,2} \\ S_2 &= S_{2,0} \parallel S_{2,1} \parallel S_{2,2} \\ S_3 &= S_{3,0} \parallel S_{3,1} \parallel S_{3,2} \end{aligned}$$

The recovery example is explained as follows: The proposed XOR-based multi-secret image sharing scheme should be recovered by collecting consecutive shared images. In the proposed (4, 3, {2, 3, 4}) XOR-based multi-secret image sharing scheme, the collections of $\{S_0, S_1\}$, $\{S_1, S_2\}$, or $\{S_2, S_3\}$ meet the requirement of threshold 2 and the consecutive requirement. Moreover, the collections of $\{S_0, S_1, S_2\}$ or $\{S_1, S_2, S_3\}$ meet the requirement of threshold 3 and the consecutive requirement. The following introduces the recovery of three different collections of shared images.

1. The collection of $\{S_1, S_2\}$;
   - 1.1. Splitting S_1 and S_2 acquire intermediate shares $S_{1,0}$, $S_{1,1}$, $S_{1,2}$ and $S_{2,0}$, $S_{2,1}$, $S_{2,2}$, respectively;
   - 1.2. Since $P_{0,1} = S_{1,0} \oplus R_{0,1} \oplus R_{0,0}$ and $P_{0,2} = S_{0,0} \oplus R_{0,0} \oplus R_{0,1}$, $S_{1,0}$ and $S_{0,0}$ can also be acquired by $P_{0,1} \oplus R_{0,1} \oplus R_{0,0}$ and $P_{0,2} \oplus R_{0,0} \oplus R_{0,1}$, respectively;
   - 1.3. Directly applying $S_{1,0} \oplus S_{0,0}$ acquires $P_{0,1} \oplus P_{0,2}$, and X_0 can be obtained by Equation (20);
   - 1.4. $R_{0,0}$ and $R_{0,1}$ are then acquired from Equations (21) and (22);
   - 1.5. By using $S_{1,0}$, $S_{0,0}(S_{2,0})$, $R_{0,0}$, and $R_{0,1}$, the intermediate images $P_{0,1}$ and $P_{0,2}$ can be then obtained;
   - 1.6. Finally, the reverse processing of the *SIPS* can obtain half bits of higher bit planes (LSB4–7) to have the rough image result;
2. The collection of $\{S_1, S_2, S_3\}$;
   - 2.1. Similar to the process in the previous collection, S_1, S_2, S_3 can also be split into intermediate shares $S_{1,0}$ $S_{1,1}$ $S_{1,2}$, $S_{2,0}$ $S_{2,1}$ $S_{2,2}$ and $S_{3,0}$ $S_{3,1}$ $S_{3,2}$, respectively;
   - 2.2. Directly applying $S_{1,1} \oplus S_{2,1} \oplus S_{3,1}(S_{0,1})$ acquires the same result of $P_{0,1} \oplus P_{0,2} \oplus P_{0,0}$;
   - 2.3. X_1 can be then obtained by Equation (20) so as to acquire $R_{1,0}$, $R_{1,1}$ and $R_{1,2}$;
   - 2.4. From $S_{1,0}$, $S_{2,0}$, $S_{3,0}(S_{0,0})$, $R_{1,0}$, $R_{1,1}$, $R_{1,2}$, and reverse processing of the *SIPS*, another half bits of a higher bit planes (LSB4–7) are then acquired;
   - 2.5. Therefore, $\{S_1, S_2, S_3\}$ recovers all bits under higher bit planes (LSB4–7);
3. The collection of $\{S_0, S_1, S_2, S_3\}$;
   - 3.1. Since $\{S_0, S_1, S_2, S_3\}$ includes $\{S_1, S_2\}$ and $\{S_1, S_2, S_3\}$, all bits in a higher bit planes (LSB4–7) are perfectly recovered;
   - 3.2. Splitting $\{S_0, S_1, S_2, S_3\}$ acquires $\{S_{0,0}, S_{0,1}, S_{0,2}\}$, $\{S_{1,0}, S_{1,1}, S_{1,2}\}$, $\{S_{2,0}, S_{2,1}, S_{2,2}\}$, and $\{S_{3,0}, S_{3,1}, S_{3,2}\}$, respectively;
   - 3.3. Using $S_{0,2}$, $S_{1,2}$, $S_{2,2}$, $S_{3,2}$, the X_2, $R_{2,0}$, $R_{2,1}$, $R_{2,2}$, and $R_{2,3}$ are orderly calculated for obtaining the lower bit planes (LSB0–3);
   - 3.4. The collection of $\{S_0, S_1, S_2, S_3\}$ recovers the secret images information of two half bits of a higher bit planes (LSB4–7) from $\{S_1, S_2\}$ and $\{S_1, S_2, S_3\}$, and the

lower bit planes (LSB0–3) from $\{S_0, S_1, S_2, S_3\}$. Therefore, all secret images can then be perfectly recovered.

4. Experimental Results and Discussions

This section demonstrates the experimental results of our proposed approach. Section 4.1 presents the experimental results of the proposed approach to share four and five secret images. All experiments are run on a PC with an Intel i5-4210 CPU and 12G RAM, using the MATLAB R2019b software. The size of all experimental secret images is 256 × 256. Section 4.2 presents the performance of the proposed approach by comparing other multiple secret image sharing methods.

4.1. Experimental Results

Figure 5 shows the experimental results of sharing by thresholds (4, 3, {2, 3, 4}) with sharing four secret images and 3 recovery thresholds by collecting 2, 3, or 4 consecutive shared images. Figure 5a–d shows the four secret images: Airplane, Baboon, Barbara, and Boat. Figure 5e shows the result of XOR of all pictures, and there are still afterimages of secret images. Figure 5f shows the generated random image R, which is obtained by hash and bit reverse calculation in Figure 5e. Figure 5g–j show four shared images with size 342 × 256, in which the size is acquired from the concatenation for three thresholds {2, 3, 4} by the equation $\frac{m}{(2(t-1)T_i)} \times MN$ as 128 × 256, 86 × 256, 128 × 256, respectively. Figure 5k–n are the recovery results by gathering consecutive shared images in Figure 5g–h of satisfying the first threshold 2. Moreover, gathering three consecutive shared images in Figure 5g–i recover the results shown in Figure 5o–r, which are a little different comparing with the original secret images in Figure 5a–d. Gathering all four secret images recovers the secret images perfectly, as shown in Figure 5a–d.

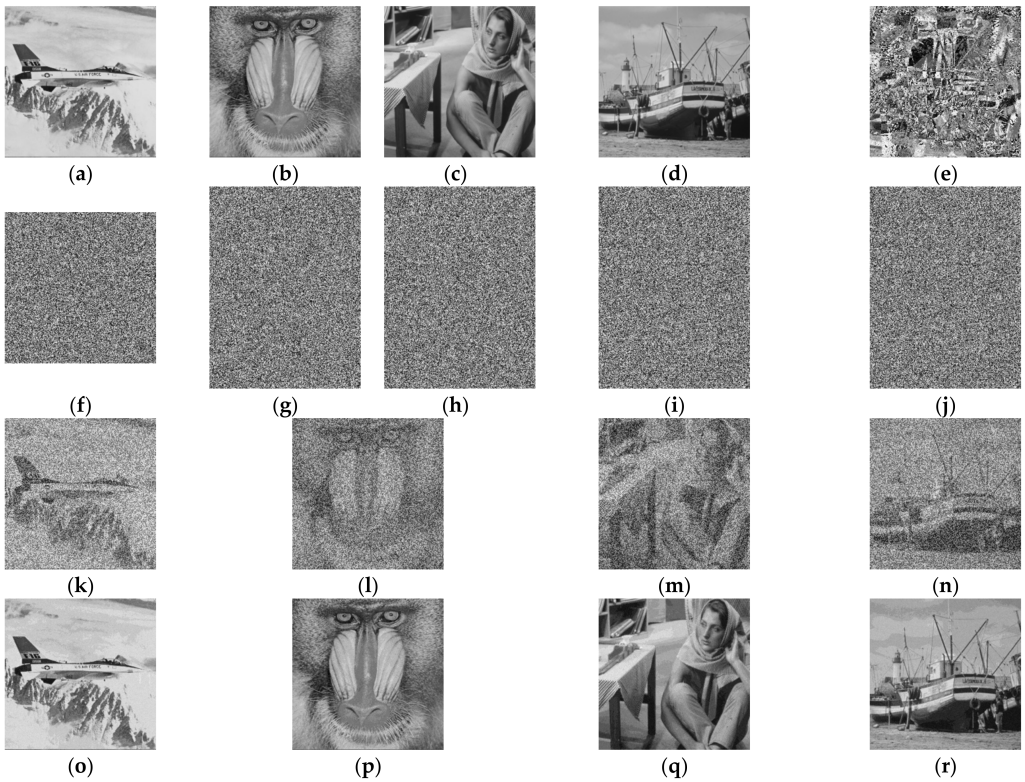

Figure 5. A (4, 3, {2, 3, 4}) XOR-based progressively secret image sharing example, (**a**–**d**) the secret images, (**e**) XOR result of (**a**–**d**), (**f**) the generated random image, (**g**–**j**) shared images, (**k**–**n**) recovery results by using (**g**–**h**), (**o**–**r**) recovery results by using (**g**–**i**).

Figure 6 shows another example of sharing five secret images by thresholds (5, 4, {2, 3, 4, 5}) consisting of 4 progressive recovery steps. Collecting 2, 3, 4, or 5 consecutive shared images recover high noise (Figure 6m–q), light noise (Figure 6r–v), clear (like Figure 5o–r), or perfect recovery (like Figure 6a–e), respectively. Figure 6h–l shows five shared images with size 361 × 256, in which the size is acquired from the concatenation for four thresholds {2, 3, 4, 5} by the equation $\frac{m}{(2(t-1)T_i)} \times MN$ as 107 × 256, 72 × 256, 54 × 256, and 128 × 256, respectively. However, the size of shared images may not be a square integer due to the proposed *SIPS* method and compensation with bit 0 is needed. Moreover, the number of compensation bits is limited and will not have a visual effect. The progressive recovery property of the proposed XOR-based (m, t, T_i) multi-secret image sharing scheme is clearly demonstrated in Figures 5 and 6.

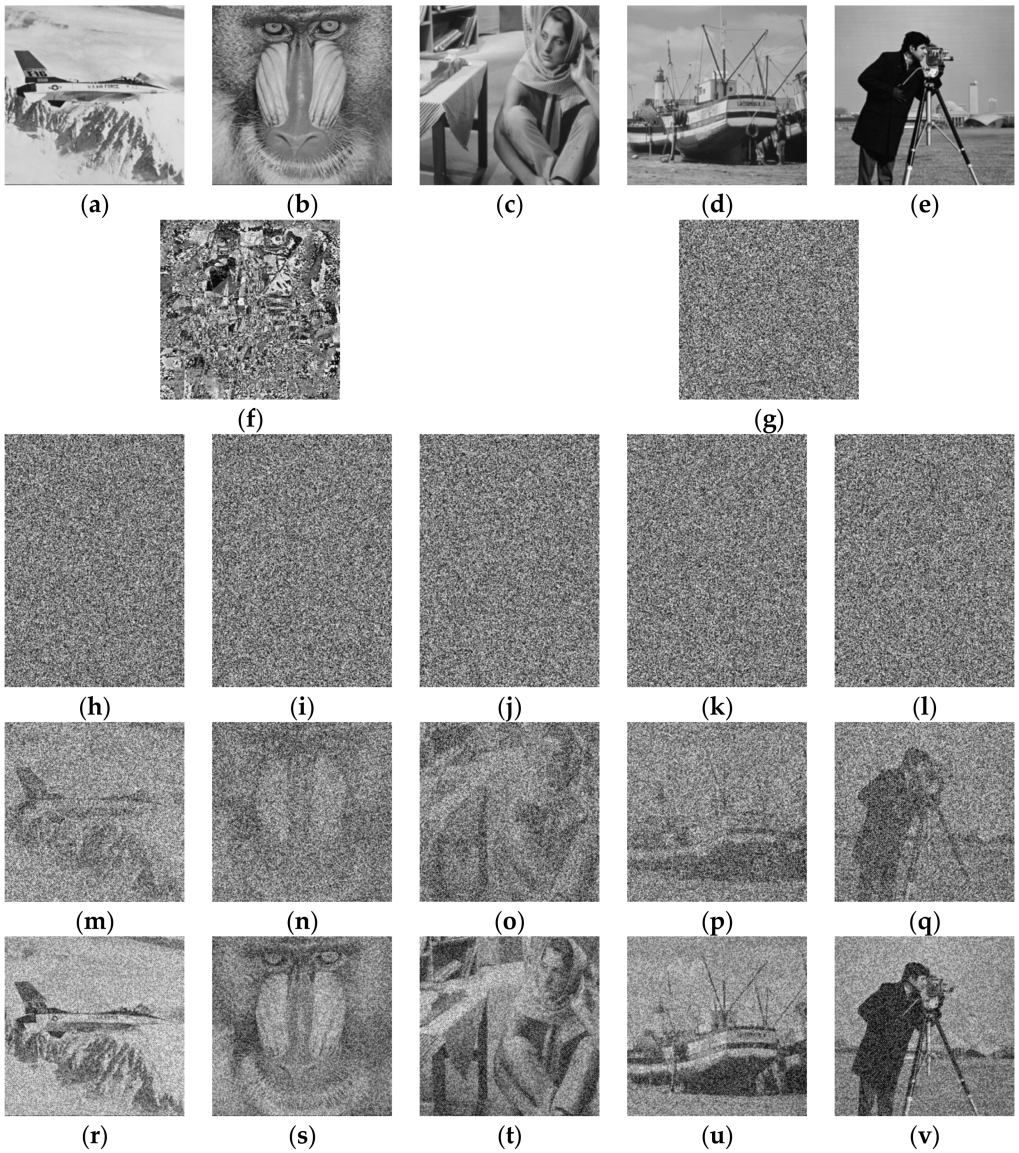

Figure 6. A (5, 4, {2, 3, 4, 5}) XOR-based progressively secret image sharing example, (**a**–**e**) the secret images, (**f**) XOR result of (**a**–**e**), (**g**) the generated random image, (**h**–**l**) shared images, (**m**–**q**) recovery results by using (**h**–**i**), (**r**–**v**) recovery results by using (**h**–**j**).

Figure 7 shows an experiment of collecting all 5 shared images in a (5, 4, {2, 3, 4, 5}) XOR-based progressively secret image sharing, in which one shared image Figure 7a is under attacked. In our proposed scheme, collecting 5 shared images with no attacked shared image leads to perfect recovery. In Figure 7a, a region with black pixels is replaced at the center of the shared image, and the attacked region modifies the data required to

recover in threshold 2. Therefore, the recovery of threshold 2 cannot be fulfilled. However, the data required to recover in threshold 3, 4, or 5 are still correct. Thus, the recovery results in Figure 7f–j exhibit the property of gathering satisfying the thresholds 3, 4 and 5.

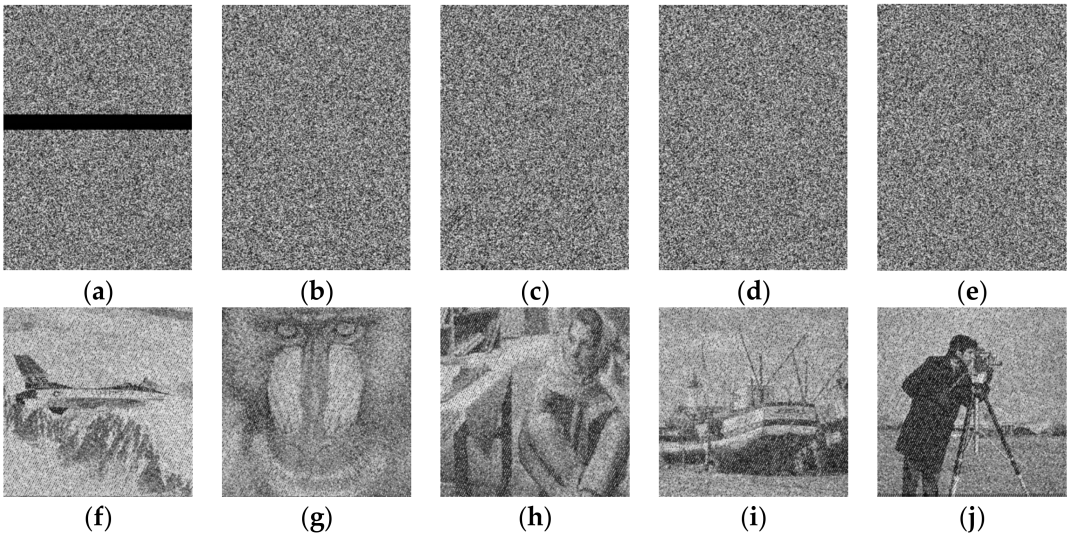

Figure 7. An example of collecting 5 shared images with one attacked shared image in a (5, 4, {2, 3, 4, 5}) XOR-based progressively secret image sharing, (**a**) the attacked shared image, (**b**–**e**) the correct shared images, (**f**–**j**) the recover results by using (**a**–**e**).

Figure 8 shows another attack experiment. In Figure 8a, the attacked region covers the data required to satisfy thresholds 2, 3, 4, and 5. Therefore, no correct secret images can be recovered, and the recovered images shown in Figure 8f–j all look like noised images.

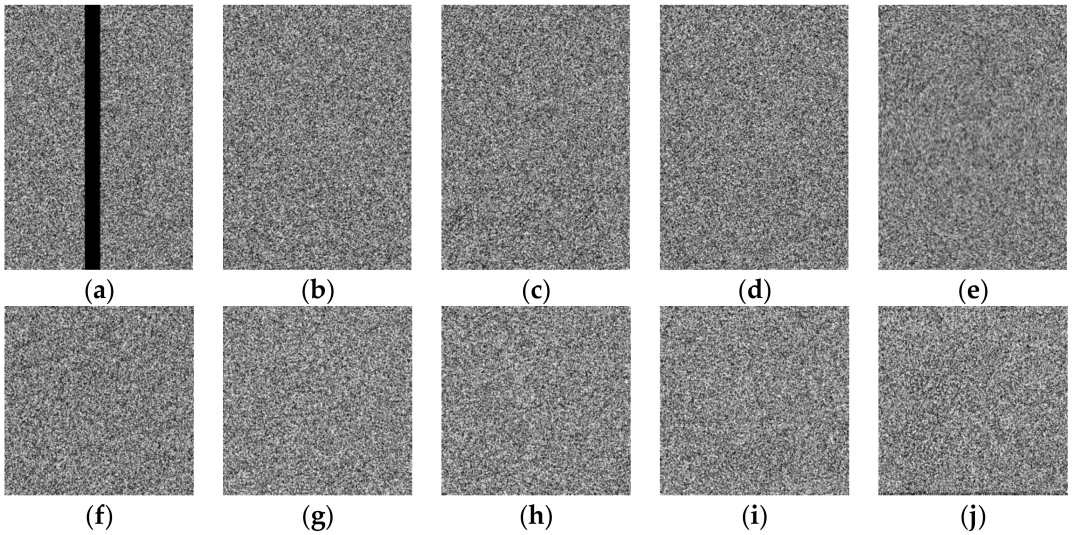

Figure 8. Another example of collecting 5 shared images with one attacked shared image in a (5, 4, {2, 3, 4, 5}) XOR-based progressively secret image sharing, (**a**) the attacked shared image, (**b**–**e**) the correct shared images, (**f**–**j**) the recover results by using (**a**–**e**).

Table 2 shows the computation time of the proposed scheme in two thresholds. The proposed Boolean-based scheme requires limited CPU computation time, and the performance of the proposed scheme is, therefore, exhibited.

Table 2. Performance of sharing and recovery procedures at different thresholds.

Thresholds	Sharing Computation Time (s)	Recovery Computation Time (s)
(4, 3, {2, 3, 4})	1.734	1.422
(5, 4, {2, 3, 4, 5})	2.334	2.038

4.2. Comparison and Discussion

This section compares the proposed method with other multiple secret image sharing methods. The comparison metrics include recovery results, color levels, recovery methods, sharing types, sharing rates, and progressive or not. The recovery results are divided into recognizable recovery or lossless recovery. The color scales are divided into binary images and grayscale images by secret images. The recovery methods are to superimpose shared images or use a calculation to restore. The sharing rates are the value of the size of secret images divided by the size of shared images. Although the shared image size produced by the proposed method is not fixed, the total size of shared images is larger than the original image due to the presented consecutive recovery strategy. Table 3 shows that the proposed method is the only XOR-based method that can perform progressive sharing with no distortion reduction and grayscale.

Table 3. Comparisons between the proposed approach and related multiple secret image sharing methods.

	Recovery Results	Color Levels	Recovery Methods	Sharing Types	Sharing Rates	Progressive
Wu and Chang [29]	Recognizable	Binary	Superimpose	Circle	$\frac{2}{2\times4}$	No
Chen et al. [25]	Recognizable	Binary	Superimpose	Circle	$\frac{n}{2\times4}$	No
Shyu et al. [26]	Recognizable	Binary	Superimpose	Circle	$\frac{n}{2\times(2\times n)}$	No
Lin et al. [27]	Recognizable	Binary	Superimpose	Rectangle	$\frac{2}{2}$	No
Wang et al. [7]	Lossless	Grayscale	XOR	Rectangle	$\frac{1}{n}$	No
Chen and Wu [15]	Lossless	Grayscale	XOR	Rectangle	$\frac{n-1}{n}$	No
Chen and Wu [16]	Lossless	Grayscale	XOR	Rectangle	$\frac{n}{n}$	No
Chen et al. [17]	Lossless	Grayscale	XOR	Rectangle	$\frac{n}{n}$	No
The proposed approach	Lossless	Grayscale	XOR	rectangle	$\sum_{i=0}^{t-1} \frac{m}{(2(t-1)T_i)} + \frac{1}{2}$	Yes

5. Conclusions

This paper proposed a high security and efficiency XOR-based (*m*, *t*, *Ti*) multi-secret image sharing scheme. The proposed method only needs to perform XOR operations without recording random images. The proposed secret image partition strategy (*SIPS*) generates intermediate images for different thresholds in the sharing procedure. Based on the property of progressive recovery, the proposed scheme should be recovered by collecting consecutive shared images. The experimental results and analyses show that the proposed method outperforms previous schemes. In the future, we will study how to limit the sharing rate to further improve our approach.

Author Contributions: Conceptualization, C.-S.L., C.-C.C.; methodology, C.-C.C.; validation, C.-S.L., C.-C.C., Y.-C.C.; formal analysis, C.-S.L., C.-C.C.; investigation, Y.-C.C.; writing—original draft preparation, writing—review and editing, C.-S.L., C.-C.C., Y.-C.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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. Chanu, O.B.; Neelima, A. A survey paper on secret image sharing schemes. *Int. J. Multimed. Inf. Retr.* **2019**, *8*, 195–215. [CrossRef]
2. Thien, C.-C.; Lin, J.-C. Secret image sharing. *Comput. Graph.* **2002**, *26*, 765–770. [CrossRef]
3. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
4. Blakley, G.R. Safeguarding cryptographic keys. In Proceedings of the 1979 International Workshop on Managing Requirements Knowledge (MARK), New York, NY, USA, 4–6 June 1979; Volume 48, pp. 313–317.
5. Ulutas, M.; Ulutas, G.; Nabiyev, V.V. Invertible secret image sharing for gray level and dithered cover images. *J. Syst. Softw.* **2013**, *86*, 485–500. [CrossRef]
6. Wu, X.; Ou, D.; Liang, Q.; Sun, W. A user-friendly secret image sharing scheme with reversible steganography based on cellular automata. *J. Syst. Softw.* **2012**, *85*, 1852–1863. [CrossRef]
7. Wang, D.; Zhang, L.; Ma, N.; Li, X. Two secret sharing schemes based on Boolean operations. *Pattern Recognit.* **2007**, *40*, 2776–2785. [CrossRef]
8. Kabirirad, S.; Eslami, Z. Improvement of (n, n)-multi-secret image sharing schemes based on Boolean operations. *J. Inf. Secur. Appl.* **2019**, *47*, 16–27. [CrossRef]
9. Dhara, B.C.; Chanda, B. A fast progressive image transmission scheme using block truncation coding by pattern fitting. *J. Vis. Commun. Image Represent.* **2012**, *23*, 313–322. [CrossRef]
10. Fang, W.-P. Friendly progressive visual secret sharing. *Pattern Recognit.* **2008**, *41*, 1410–1414. [CrossRef]
11. Huang, C.-P.; Hsieh, C.-H.; Huang, P.S. Progressive sharing for a secret image. *J. Syst. Softw.* **2010**, *83*, 517–527. [CrossRef]
12. Lin, S.-J.; Lin, J.-C. VCPSS: A two-in-one two-decoding-options image sharing method combining visual cryptography (VC) and polynomial-style sharing (PSS) approaches. *Pattern Recognit.* **2007**, *40*, 3652–3666. [CrossRef]
13. Wang, R.-Z.; Su, C.-H. Secret image sharing with smaller shadow images. *Pattern Recognit. Lett.* **2006**, *27*, 551–555. [CrossRef]
14. Chen, C.C.; Chien, Y.W. Sharing numerous images secretly with reduced possessing load. *Fundam. Inf.* **2008**, *86*, 447–458.
15. Chen, T.-H.; Wu, C.-S. Efficient multi-secret image sharing based on Boolean operations. *Signal Process.* **2011**, *91*, 90–97. [CrossRef]
16. Chen, C.-C.; Wu, W.-J. A secure Boolean-based multi-secret image sharing scheme. *J. Syst. Softw.* **2014**, *92*, 107–114. [CrossRef]
17. Chen, C.-C.; Wu, W.-J.; Chen, J.-L. Highly efficient and secure multi-secret image sharing scheme. *Multimedia Tools Appl.* **2015**, *75*, 7113–7128. [CrossRef]
18. Kabirirad, S.; Eslami, Z. A(t,n)-multi secret image sharing scheme based on Boolean operations. *J. Vis. Commun. Image Represent.* **2018**, *57*, 39–47. [CrossRef]
19. Azza, A.A.; Lian, S. Multi-secret image sharing based on elementary cellular automata with steganography. *Multimed. Tools Appl.* **2020**, *79*, 21241–21264. [CrossRef]
20. Chen, T.-H.; Wu, X.-W. Multiple secret image sharing with general access structure. *Multimed. Tools Appl.* **2020**, *79*, 13247–13265. [CrossRef]
21. Ross, A.; Othman, A. Visual Cryptography for Biometric Privacy. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 70–81. [CrossRef]
22. Yan, X.; Lu, Y.; Yang, C.-N.; Zhang, X.; Wang, S. A Common Method of Share Authentication in Image Secret Sharing. *IEEE Trans. Circuits Syst. Video Technol.* **2020**. [CrossRef]
23. Yue, J.; Yan, X.; Qi, J.; Lu, Y.; Zhou, X. Secret image sharing with dealer-participatory and non-dealer-participatory mutual shadow authentication capabilities. *Mathematics* **2020**, *8*, 234.
24. Li, J.; Wang, X.; Huang, Z.; Wang, L.; Xiang, Y. Multi-level multi-secret sharing scheme for decentralized e-voting in cloud computing. *J. Parallel Distrib. Comput.* **2019**, *130*, 91–97. [CrossRef]
25. Chen, J.; Chen, Y.S.; Hsu, H.C.; Chen, H.W. New visual cryptography system based on circular shadow image and fixed angle segmentation. *J. Electron. Imaging* **2005**, *14*, 033018. [CrossRef]
26. Shyu, S.J.; Huang, S.-Y.; Lee, Y.-K.; Wang, R.-Z.; Chen, K. Sharing multiple secrets in visual cryptography. *Pattern Recognit.* **2007**, *40*, 3633–3651. [CrossRef]
27. Lin, S.-J.; Chen, S.-K.; Lin, J.-C. Flip visual cryptography (FVC) with perfect security, conditionally-optimal contrast, and no expansion. *J. Vis. Commun. Image Represent.* **2010**, *21*, 900–916. [CrossRef]
28. Chen, C.C.; Fu, W.F. A geometry-based secret image sharing approach. *J. Informat. Sci. Eng.* **2008**, *24*, 1567–1577.
29. Wu, H.C.; Chang, C.C. Sharing visual multi-secrets using circle shares. *Comput. Stand. Interfaces* **2005**, *28*, 123–135.
30. Bhattacharjee, T.; Maity, S.P.; Islam, S.R. Hierarchical secret image sharing scheme in compressed sensing. *Signal Process. Image Commun.* **2018**, *61*, 21–32. [CrossRef]
31. Chanu, O.B.; Neelima, A. A new multi-secret image sharing scheme based on DCT. *Vis. Comput.* **2019**, *36*, 939–950. [CrossRef]