**Article**

**Boolean-Based \((k, n, m)\) Multi-Secret Image Sharing**

Chien-Chang Chen, Cheng-Shian Lin * and Jia-Zhan Chen

Department of Computer Science and Information Engineering, Tamkang University, New Taipei City 25137, Taiwan; ccchen34@mail.tku.edu.tw (C.-C.C.); a22781911@gmail.com (J.-Z.C.)

* Correspondence: cslin@mail.tku.edu.tw

**Abstract:** Multi-secret image sharing technology shares multiple secret images among shared images for participants and the recovery requires enough shared images for fulfilling the security property. Simultaneously, the adopted Boolean method preserves high performance. The proposed \((k, n, m)\) multi-secret image sharing scheme shares \(m\) secret images among \(n\) participants and gathering \(k\) participants’ shared images perfectly recovers these \(m\) secret images. The proposed scheme exhibits high performance because of the usage of Boolean-based processing. Moreover, the usage of a sharing matrix based strategy acquires perfect reconstruction of the secret images under \((k, n)\) thresholds. Two proposed schemes, the source-random-mixed (SRM) and the source-random-separate (SRS) methods, are presented in this study from different strategies of sharing the random image mixed with the secret images or separated to the secret images, respectively. Moreover, the proposed SRS method exhibits the better sharing capacity. Experimental results show that the proposed schemes perform well on sharing efficiency and security.

**Keywords:** secret image sharing; Boolean-based; multiple sharing

**1. Introduction**

The popular usage of multimedia content leads to a security problem on transmission or storage being increasingly more important. Secret image sharing technique, aiming to protect the multimedia content among participants, is one of the protection methods and has become an important research field [1].

The secret image sharing problem was first presented by Thien and Lin [2] with solving by the Shamir–Lagrange method [3]. The secret image sharing technique is then extended to various research areas, such as sharing among host images [4,5], sharing using Boolean operations [6,7], progressive sharing [8–10], sharing with visual cryptography property [11], multiple sharing [12,13], and scalable sharing [14]. Among these research topics, sharing multiple secret images efficiently is an important topic in real cases. Moreover, the Boolean-based scheme outperforms the functional-based schemes on computational complexity but is hard to achieve various properties such as \((k, n)\) partial recovery, multiple sharing, or progressive sharing. Therefore, a Boolean-based secret image sharing scheme with \((k, n)\) thresholds on multiple secret images sharing is very important and merits our study.

The Boolean-based approaches can be classified into two categories, the stacking-based and the XOR-based methods. The stacking-based methods [15–18], which are based on visual cryptography [19], are the least computational Boolean-based methods. However, the stacking-based methods always lead to a rough visual effect. Consequently, the XOR-based methods are the key technique having both efficient computation complexity and perfect recovered results. Therefore, our proposed method is based on XOR-based technique. Moreover, sharing matrix strategy [20] is an efficient method to achieve perfect recovery. Therefore, the proposed scheme is based on XOR operations and the concept of sharing matrix.

Over the past years, several Boolean-based approaches have been presented. Wang et al. [6] adopted simple Boolean operations to present an \((n, n)\) threshold scheme. Chen and Wu [12]...
proposed a Boolean-based multiple secret image sharing method which used \( n - 1 \) secret images to generate \( n \) shared images. Collecting all shared images is the requirement of their proposed method. The Boolean-based multiple secret image sharing method [12] was then improved by Chen and Wu [13] by using a random number image generation function, which obtains the same random image from secret images or shared images. Therefore, the sharing capacity is increased due to the no random image requirement. Chen et al. [21] further modified the usage of random image generation to a Boolean-based symmetric recovery function. Notice that most previous Boolean-based schemes are mainly for \((n, n)\) thresholds. Bao et al. [20] proposed the sharing matrix to acquire Boolean-based \((k, n)\) secret image sharing. Prasetyo and Jing [22] present the Chinese remainder theorem and XOR-based \((n, n)\) secret image sharing. Chattopadhyay et al. [23] acquired \((n, n)\) secret image sharing with scalable shared images. Kabirirad and Eslami [24] present \((k, n)\) secret image sharing with recovery by consecutive \( k \) shared images. Kabirirad and Eslami [25] proposed a Boolean-based multiple secret image sharing scheme. However, the size of the generated shared image must be the same as the secret image. Azza and Lian [26] present a multi-secret image sharing scheme using cellular automaton with steganography. Chen and Wu [27] proposed a multi-secret sharing scheme based on Boolean operation. Based on the proposed predefined access structure, the recovery of secret images does not require collection of all the shared images. Moreover, the way to construct multi-secret image sharing with general \((k, n)\) thresholds is still not well solved.

This paper proposes a secure Boolean-based \((k, n, m)\) multiple secret image sharing approach that shares \( m \) secret images among \( n \) shared images, and collecting \( k \) shared images perfectly recovers these \( m \) secret images. Two schemes are presented in this study: the source-random-mixed scheme (SRM) sharing the XOR result among each shared image and the random image, and the source-random-separate scheme (SRS) that shares the random image individually. Experimental results show the novel properties of the two proposed approaches. Moreover, the proposed SRS scheme outperforms others for its usage of sharing matrix on secret images distribution.

The paper is organized into the following sections. Section 2 gives a brief review of related works [20,21]. Section 3 presents details of two proposed sharing and recovery approaches, the source-random-mixed scheme (SRM) and the source-random-separate scheme (SRS). 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 Boolean-based secret sharing schemes with the function of multiple images sharing. Section 2.1 introduces the secret sharing of multiple images based on the Boolean function proposed by Chen et al. [21]. Section 2.2 introduces the sharing matrix-based \((k, n)\) secret image sharing method by Bao et al. [20].

#### 2.1. Multiple Secret Images Sharing Scheme

Chen et al. [21] proposed a Boolean-based \((n, n)\) secret image sharing scheme, in which \( n \) secret images are shared among \( n \) shared images and by Boolean operation. The sharing steps are introduced as follows.

1. Acquire the random image \( R \) with the secret images \( I_i (0 \leq i \leq n - 1) \) by Equation (1):

\[
\begin{align*}
    h &= H(I_0 \oplus I_1 \oplus I_2 \cdots \oplus I_{n-1}), \\
    SI &= \text{image synthesis}(h), \\
    R &= \text{bit reverse}(I_0 \oplus I_1 \oplus I_2 \cdots \oplus I_{n-1}) \oplus SI,
\end{align*}
\]

where \( H() \) represents the SHA-256 hash function, \( \text{image synthesis}(h) \) synthesizes the random image from the seed \( h \), and \( \text{bit reverse}(l) \) reverses the image \( l \) on the bit plane.

2. Shift the random image with different pixels to acquire a series of random images \( R_i (0 \leq i \leq n - 1) \), as given in Equation (2):
\[ R_i(x, y) = R(x - i, y - i). \] (2)

3. Acquire shared images \( O_i \) for participant \( i \) from Equation (3):

\[
O_i = \begin{cases} 
I_0 \oplus R_0 \oplus R_1, & i = 0, \\
I_1 \oplus R_1 \oplus R_2, & i = 1, \\
I_2 \oplus R_2 \oplus R_3, & i = 2, \\
\vdots \\
I_{n-2} \oplus R_{n-2} \oplus R_{n-1}, & i = n - 2, \\
I_{n-1} \oplus R_{n-1} \oplus R_0, & i = n - 1.
\end{cases} 
\] (3)

The recovery steps are introduced as follows.

1. Collect all \( n \) shared images \( O_i \) and perform XOR operation \( O_0 \oplus \ldots \oplus O_{n-2} \oplus O_{n-1} \) on all shared images, as defined by Equation (4):

\[
O_0 \oplus \ldots \oplus O_{n-2} \oplus O_{n-1} = (I_0 \oplus R_0 \oplus R_1) \oplus \ldots \oplus (I_{n-1} \oplus R_{n-1} \oplus R_0),
\] (4)

2. Apply the result of Equation (4) to Equations (1) and (2) to obtain a series of random images \( R_i(0 \leq i \leq n - 1) \).

3. Acquire the recovered secret images \( I'_i \) by applying \( O_i \) and \( R_i \) to Equation (5):

\[
I'_i = \begin{cases} 
O_0 \oplus R_0 \oplus R_1, & i = 0, \\
O_1 \oplus R_1 \oplus R_2, & i = 1, \\
O_2 \oplus R_2 \oplus R_3, & i = 2, \\
\vdots \\
O_{n-2} \oplus R_{n-2} \oplus R_{n-1}, & i = n - 2, \\
O_{n-1} \oplus R_{n-1} \oplus R_0, & i = n - 1.
\end{cases} 
\] (5)

The recovered secret images \( I'_i \) and original secret images \( I_i \) are consistent when all the shared images \( O_i \) are correctly acquired. Moreover, Chen et al.’s scheme [21] only exhibits \((n, n)\) secret image sharing for sharing among \( n \) shared images and recovery requirement is all \( n \) shared images. For acquiring a Boolean-based and robust \((k, n)\) secret image sharing scheme, Bao et al. [22] presented sharing an important matrix-based method, which is introduced in the next section.

2.2. Boolean-Based \((k, n)\) Secret Image Sharing Scheme

The Boolean-based scheme is difficult to achieve with \((k, n)\) secret image sharing when \( k \leq n \). Bao et al. [20] present a novel \((k, n)\) secret image sharing scheme by using the proposed sharing matrix to achieve the recovery of only acquiring \( k \) shared images rather than \( n \) shared images. Assume that a secret image with size \( r \times c \) is shared among \( n \) shared images with recovery thresholds \((k, n)\); their sharing algorithm is introduced as follows.

1. Generate a series of \( n \) binary sharing matrices \( M_i(0 \leq i \leq n - 1) \) with size \( r \times c \), in which each matrix should satisfy the following two criteria:
   A. Gathering \( k \) sharing matrices leads to each entry containing at least a 1 among these sharing matrices.
   B. Gathering less than \( k \) sharing matrices leads to at least an entry without finding a 1 among these sharing matrices.

2. Encrypt the secret image \( I \) to acquire a randomlike image \( R \).

3. For a sharing matrix \( M_i \), each entry \( M_i(j,k)(0 \leq j \leq r - 1, 0 \leq k \leq c - 1) \) determines the corresponding shared image entry \( S_i(j,k) \), as defined in Equation (6):
Each sharing matrix

1. Reconstruct the same series of Rand where

2. Acquire the encrypted image among sharing matrices.

However, the selection of $P$ fits the requirement of 3.1 being at least one matrix having the value of 1 in all entries. For example, the selection of $P$ of $n \times m$ threshold (3, 4) and matrix size 4 is illustrated in Figure 1. The predetermined selection of $n = 4$ acquires 4 sharing matrices $P_1$ to $P_4$ and $k = 3$ shows that selecting 3 sharing matrices should fit the above criterion 3.1. For example, the selection of $P_1$, $P_2$, and $P_3$ fits the requirement of 3.1 being at least one matrix having the value of 1 in all entries. However, the selection of $P_1$, and $P_2$ fits the criterion of Step 3.2, with entries such as $P_1(1,1) = P_2(1,1) = 0$ or $P_1(2,6) = P_2(2,6) = 0$. Therefore, the $(k, n)$ thresholds determine the collection of $k$ sharing matrices acquires at least one matrix that has the value of 1 in all entries and the collection of $k - 1$ sharing matrices at least have the value of 0 in one entry among sharing matrices.

$$S_i(j, k) = \begin{cases} R(j,k), & \text{if } M_i(j,k) = 1, \\ \text{Rand()}, & \text{if } M_i(j,k) = 0, \end{cases}$$

(6)

where Rand() represents a random number generation function.

Since the encrypted image $R$ in Step 2 is a randomlike image, Step 3 acquires a shared image that is also randomlike because part of the entries of $R$ are $S_i$ and other entries are random numbers. According to the novel arrangement of the sharing matrices $M_i$, gathering $k$ shared images $S_i$ can acquire all entries $S_i(j,k)(0 \leq j \leq r - 1, 0 \leq k \leq c - 1)$ of $R$ and then decrypt to the secret image $I$ perfectly. However, gathering less than $k$ shared images cannot acquire all entries of $R$ and the secret image cannot then be decrypted.

The recovery algorithm is introduced as follows.

1. Reconstruct the same series of $n$ binary sharing matrices $M_i(0 \leq i \leq n - 1)$ as applied in the sharing algorithm.
2. Acquire the encrypted image $R$ from $M_i$ and $S_i$ by applying Equation (7)

$$R(j,k) = S_i(j,k) \text{ if } M_i(j,k) = 1.$$ (7)

3. Decrypt the secret image from $R$ to acquire the recovered secret image.

In Step 2, each entry of the randomlike image $R$ will be correctly recovered because of the property of binary sharing matrices $M_i$, as introduced in Step 1 of the sharing algorithm.

3. The Proposed Approach

This section introduces our proposed XOR-based $(k, n, m)$ multi-secret image sharing scheme, in which $m$ secret images are shared among $n$ participants with $k$ thresholds for perfectly recovering all secret images.

Section 3.1 introduces the generation of the sharing matrix, Sections 3.2 and 3.3 introduce two proposed XOR-based $(k, n, m)$ multi-secret image sharing algorithms: the source-random-mixed method (SRM) and source-random-separate method (SRS), respectively. Examples of the proposed SRS sharing and recovery with thresholds $(2, 3, 2)$ are demonstrated in Section 3.4.

3.1. Sharing Matrix Generation

Since the sharing matrix has to acquire the property of $(k, n)$ thresholds, all secret images are consistent in size and the sharing matrices $P_i(i = 1, 2, \ldots, n)$ have the following properties:

1. Each sharing matrix $P_i$ has the same size with secret images.
2. Each element in $P_i$ are either 0 or 1.
3. Selecting $p$ matrices of $P_i$, for example, $i = 1, 2, \ldots, p$, the following criteria should be satisfied.

   3.1 When $p \geq k$, at least one matrix has the value of 1 in all entries among these $p$ matrices.

   3.2 When $p < k$, at least one entry has the value of 0 in all $p$ matrices.

Above three properties construct the sharing matrices $P_i(i = 1, 2, \ldots, n)$ that will be used in our two proposed methods. Without loss of generality, an example of $(k, n)$ using threshold $(3, 4)$ and matrix size $4 \times 6$ is illustrated in Figure 1. The predetermined selection of $n = 4$ acquires 4 sharing matrices $P_1$ to $P_4$ and $k = 3$ shows that selecting 3 sharing matrices should fit the above criterion 3.1. For example, the selection of $P_1$, $P_2$, and $P_3$ fits the requirement of 3.1 being at least one matrix having the value of 1 in all entries. However, the selection of $P_1$, and $P_2$ fits the criterion of Step 3.2, with entries such as $P_1(1,1) = P_2(1,1) = 0$ or $P_1(2,6) = P_2(2,6) = 0$. Therefore, the $(k, n)$ thresholds determine the collection of $k$ sharing matrices acquires at least one matrix that has the value of 1 in all entries and the collection of $k - 1$ sharing matrices at least have the value of 0 in one entry among sharing matrices.
The sharing matrix generation algorithm, a modification of Bao et al.’s method [20], is illustrated as follows for obtaining sharing matrices $P_i (i = 1, 2, \ldots, n)$ for $(k, n, m)$ thresholds with $r \times c$ secret image size.

1. Use Equation (8) to acquire an $n \times 1$ binary vector $M$:

$$M = createVector(k, n)$$

where $M$ includes $(k-1)$ number of value 0 and $(n-k+1)$ number of value 1.

2. Acquire initial matrix $N$ by permuting through $M$ as Equation (9):

$$N = permuteMatrix(M)$$

where $permuteMatrix(M)$ duplicates vector $M t = \frac{m!}{(k-1)!(n-k+1)!}$ times to acquire matrix $N$ with size $n \times t$.

3. Extend one row vector of $N$, denoted by $N(i,:)$, to acquire an extended matrix $N_i$ with size $r \times t$ by using Equation (10):

$$N_i(x,:) = shiftVector(N(i,:), x), \quad 1 \leq x \leq r \text{ and } 1 \leq i \leq n$$

where the function $shiftVector(V, x)$ circularly shifts the vector $V$ to $x$ positions.

4. Exchange rows of $N_i$ to acquire exchanged matrix $N_i'$:

$$N_i' = exchangeRows(N_i, b)$$

where the function $exchangeRows(A, b)$ randomly permutes matrix $A$ on rows by the seed $b$.

5. Extend the matrix $N_i'$ to fit the sizes of secret image for obtaining sharing matrix $P_i$:

$$P_i = repeatMatrix(N_i, w), \quad w = \frac{c}{t}, i = 1, 2, \ldots, n$$

where the function $repeatMatrix(A, w)$ duplicates the $A$ matrix $w$ times for acquiring the required size, which is the same size as the secret images.

Figure 2 shows sharing matrices generation step by step. Since the $(k, n)$ thresholds are $(3, 4)$, the generated binary vector $M$, as shown in Figure 2a, is $4 \times 1$ and 2 of these 4 elements are 1, as denoted in Step 1 of Equation (8). Figure 2b shows the generated initial matrix $N$, as defined in Step 2 and Equation (9). Figure 2c–e shows a series of extended matrix $N_1$, exchanged matrix $N_1'$, and sharing matrix $P_1$, respectively. These matrices are acquired from Steps 3–5. In Figure 2c, $N_1$ is acquired from the first row of $N$ with one position circular shift. Moreover, Figure 2f of $N_2$ is acquired from the second row of $N$ with one positioned circular shift.
The sharing matrices introduced in Section 3.1 are needed in the proposed sharing and recovery algorithms. Figure 4 shows the proposed sharing strategy of the proposed algorithms. The input images \( I_i \) are processed to acquire temporary images \( E_i \). On the other hand, sharing matrices \( P_i \) are calculated from thresholds \((k, n)\), as introduced in Equations (8)–(12). At last, shared images \( S_j \) are acquired from temporary images \( E_i \) and sharing matrices \( P_i \).

![Figure 2](image_url) An example of \((3, 4)\) thresholds sharing matrix generation method size \(256 \times 256\): (a) binary vector \( M \), (b) initial matrix \( N \), (c) extended matrix \( N_1 \), (d) exchanged matrix \( N_1' \), (e) sharing matrix \( P_1 \), (f) extended matrix \( N_2 \), (g) exchanged matrix \( N_2' \), (h) sharing matrix \( P_2 \).

Figure 3 shows the sharing strategy of the proposed algorithms. The input images \( I_i \) are processed to acquire temporary images \( E_i \). On the other hand, sharing matrices \( P_i \) are calculated from thresholds \((k, n)\), as introduced in Equations (8)–(12). At last, shared images \( S_j \) are acquired from temporary images \( E_i \) and sharing matrices \( P_i \).

![Figure 3](image_url) Sharing strategy of the proposed two algorithms.

3.2. The Proposed \((k, n, m)\) Source-Random-Mixed (SRM) Algorithm

This section shows our first proposed \((k, n, m)\) SRM sharing algorithm, in which \( m \) secret images with identical size are shared among \( n \) shared images and collecting \( k \) shared images perfectly recover these \( n \) secret images. Assume all the secret images are denoted by \( I_i(1 \leq i \leq m) \) with the same size \( r \times c \), and the generated shared images are \( S_j(1 \leq j \leq n) \). The sharing matrices introduced in Section 3.1 are needed in the proposed sharing and recovery algorithms. Figure 4 shows the proposed SRM sharing steps and the proposed SRM sharing algorithm is introduced as follows.
1. Apply \( m \) secret images \( I_i \) \((1 \leq i \leq m)\) to XOR operation for acquiring \( I_{\text{XOR}} \) by Equation (13):

\[
I_{\text{XOR}} = I_1 \oplus I_2 \oplus \cdots \oplus I_m,
\]

where \( \oplus \) denotes bit-based XOR operation.

2. Generate a random image \( R \) by Equation (14):

\[
h = \text{hash}(I_{\text{XOR}}),
R = \text{image\_synthesis}(h, r, c),
\]

where the \( \text{hash()} \) is performed by SHA-256 hash function and the \( \text{image\_synthesis}(h,r,c) \) synthesizes a randomlike image with size \( r \times c \) from the seed \( h \).

3. Generate a series of randomlike images \( R_i \) \((1 \leq i \leq m)\) from Equation (15):

\[
R_i = \text{pixel\_shift}(R, i), \quad 1 \leq i \leq m,
\]

where \( \text{pixel\_shift}(R,i) \) right circularly shifts the image \( R \) with \( i \) pixels.

4. A series of temporary images \( E_i \) \((1 \leq i \leq m)\) are acquired from \( I_i \) and \( R_i \) by Equation (16):

\[
E_i = \begin{cases} 
I_1 \oplus R_1 \oplus R_2, & i = 1, \\
I_2 \oplus R_2 \oplus R_3, & i = 2, \\
I_3 \oplus R_3 \oplus R_4, & i = 3, \\
\vdots \\
I_{m-1} \oplus R_{m-1} \oplus R_m, & i = m - 1, \\
I_m \oplus R_m \oplus R_1, & i = m.
\end{cases}
\]

5. Based on \((k, n)\) thresholds, generate sharing matrix \( P_j \) \((1 \leq j \leq n)\) from Equations (8)–(12).

6. Calculate initial shared images \( S_i^j \) by images \( E_i \) \((1 \leq i \leq m)\) and \( P_j \) \((1 \leq j \leq n)\), which is defined as:
\[
S_j^i(x, y) = \begin{cases} 
E_i(x, y), & \text{if } P_j(x, y) = 1, \\
0, & \text{if } P_j(x, y) = 0,
\end{cases}
\]

where the number and size of \(S_j^i\) are \(m \times n\) and \(r \times c\), respectively.

7. Obtain shared images \(S_j(1 \leq j \leq n)\) by concatenating initial shared images \(S_j^i(1 \leq i \leq m)\), as shown in Equation (18):

\[
S_j = [S_j^1 S_j^2 \ldots S_j^m].
\]

At last, the size of \(S_j\) is \(m \times (r \times c)\).

Figure 5 shows a shared image example of (3, 4, 5) thresholds. In the proposed (3, 4, 5) SRM sharing method, 5 secret images are shared to 4 shared images and collecting 3 shared images perfectly recovers these 5 secret images. Figure 5a–e shows 5 secret images and Figure 5f shows one of shared image \(S_1\), which is a concatenation of 5 initial shared images \(S_1^1 \sim S_5^5\) as virtually segmented by dotted lines.

![Figure 5](image-url)

**Figure 5.** Example of generated shared image by (3, 4, 5) thresholds: (a–e) secret images, (f) the generated shared image \(S_1\).

The recovery algorithm requires \(k\) shared images with the same size \(m \times r \times c\). The proposed recovery algorithm of SRM is introduced as follows.

1. Collect \(k\) shared images and denote them by \(S_1, S_2, \ldots, S_k\).
2. Separate all collected shared images \(S_i(1 \leq i \leq k)\) to recovered shared images \(S_j(1 \leq j \leq m)\) with size \(r \times c\).
3. Check all entries \((k, l)(1 \leq j \leq k, 1 \leq l \leq c)\) among the images \(S_j^i(1 \leq i \leq k)\) to acquire the temporary image \(S_j^{re}(1 \leq j \leq m)\). For the entry \((k, l)\) with one of the pixel \(S_j^i(k, l) \neq 0\), assign \(S_j^{re}(k, l)\) by the value of \(S_j^i(k, l)\). If all values in the entry \((k, l)\) are 0, then assign \(S_j^{re}(k, l)\) by 0.
4. Acquire the temporary image \(I_{XOR}^{re}\) by Equation (19):

\[
I_{XOR}^{re} = S_1^{re} \oplus S_2^{re} \oplus \cdots \oplus S_k^{re}.
\]

5. Apply \(I_{XOR}^{re}\) to Equation (14) for acquiring the random image \(R^{re}\).
6. Apply $R^{re}$ to Equation (15) for acquiring randomlike images $R^{re}_{i}(1 \leq i \leq m)$.

7. Apply $S^e_{i}$ and $R^{re}_{i}$ to Equation (20) for obtaining recovered secret images $I^{re}_{i}$:

$$I^{re}_{i} = \begin{cases} S^e_{1} \oplus R^{re}_{1} \oplus R^{re}_{2}, & i = 1, \\ S^e_{2} \oplus R^{re}_{2} \oplus R^{re}_{3}, & i = 2, \\ S^e_{3} \oplus R^{re}_{3} \oplus R^{re}_{4}, & i = 3, \\ \vdots \\ S^e_{m-1} \oplus R^{re}_{m-1} \oplus R^{re}_{m}, & i = m - 1, \\ S^e_{m} \oplus R^{re}_{m} \oplus R^{re}_{1}, & i = m, \end{cases}$$

(20)

After applying the above recovery algorithm by $k$ correct shared images, the recovered secret images $I^{re}_{i}(1 \leq i \leq m)$ are consistent with the original secret images. For the recovery of less than $k$ correct shared images, all randomlike images are acquired. Moreover, in Step 3, the nonzero values of all $S^e_{i}(k, j)(1 \leq i \leq k)$ should be consistent when no malicious attacks on shared images are applied.

Furthermore, the proposed SRM method assigns 0 to pixels of shared images when the corresponding coordinates of the sharing matrix are 0, as shown in Equation (17). Therefore, each shared image appears dark, as shown in Figure 5f.

3.3. The Proposed $(k, n, m)$ Source-Random-Separate (SRS) Algorithm

This section introduces our second proposed $(k, n, m)$ source-random-separate (SRS) algorithm. The sharing matrices introduced in Section 3.1 are also needed in the proposed sharing and recovery algorithms. Figure 6 shows the proposed SRS sharing steps, and the proposed SRS sharing algorithm is introduced as follows.

1. Apply $m$ secret images $I_{i}(1 \leq i \leq m)$ to XOR operation for acquiring $I_{XOR}$ by Equation (13).

2. Generate a randomlike image $Z$ with size $r \times c$ by the function $image\_synthesis(b, r, c)$ defined in Equation (14). The seed $b$ is the same seed as used to generate the exchanged matrix in Equation (11).

3. Apply the XOR operation between $I_{XOR}$ and $Z$ to acquire the randomlike image $SR_0$ by Equation (21):

$$SR_0 = I_{XOR} \oplus Z.$$  

(21)

4. Based on $(k, n)$ thresholds, generate sharing matrix $P_{j}(1 \leq j \leq n)$ by Equations (8)–(12).

5. Use Equation (22) to calculate images $SR_{j}(1 \leq j \leq n)$ by sharing matrix $P_{j}$:

$$SR_{j}(x, y) = \begin{cases} SR_0(x, y), & \text{if } P_{j}(x, y) = 1, \\ randi(255), & \text{if } P_{j}(x, y) = 0, \end{cases}$$

(22)

where $j$ ranges between 1 and $n$, and $randi(255)$ acquires a random integer between 0 and 255.

6. Apply $I_{XOR}$ and Equation (14) to generate the random image $R$.

7. Generate a series of randomlike images $R_{i}(1 \leq i \leq m)$ from the random image $R$ by Equation (15).

8. Apply $I_{i}$ and $R_{i}$ by Equation (16) to obtain encrypted images $E_{i}(1 \leq i \leq m)$.

9. Randomly select $m$ columns from the initial matrix $N$ in Equation (9) to acquire matrix $N_{part}$, i.e.,

$$N_{part} = randSelect(N, m, h),$$

(23)

where $h$ is acquired from Equation (14) and matrix $N$ is acquired from Equations (8) and (9) with size $n \times m$.

10. Use Equation (24) to acquire share $E_{j}^{e}$ by the matrix value of $N_{part}$:
\[ E_j^i = \begin{cases} E_{ij}, & \text{if } N_{part}(j, i) = 1, \\ \emptyset, & \text{if } N_{part}(j, i) = 0. \end{cases} \]  

(24)

where two coordinates \( i \) and \( j \) are ranged as \( 1 \leq i \leq m \) and \( 1 \leq j \leq n \).

11. Concatenate \( SR_j \) and \( E_j^i (1 \leq i \leq m) \) to acquire shared images \( S_j \), i.e.,

\[ S_j = SR_j E_j^1 \cdots E_j^m. \]  

(25)

Figure 6. Processes of the second proposed source-random-separate (SRS) algorithm.

The proposed SRS method shares XOR result \( I_{\text{XOR}} \) and the randomlike images \( R_i (1 \leq i \leq m) \) with secret images \( I_i (1 \leq i \leq m) \) separately. Especially, in Step 10, the images \( E_i \) are selected according to the \( N_{part} \) matrix. Assume in the proposed \((3, 4, 5)\) SRS method with the encrypted images \( E_i (1 \leq i \leq 5) \) and the matrix \( N_{part} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \), the concatenation of images \( E_i \) is acquired from the matrix element of \( N_{part} \). In the first row, the 1st and 3rd elements being 1 leads to share \( E_1^i (1 \leq i \leq 5) = \{ E_1, \emptyset, E_3, \emptyset, \emptyset \} \). From the \( N_{part} \) matrix, we can also acquire shares \( E_2^i = \{ E_1, \emptyset, \emptyset, E_4, E_5 \} \), \( E_3^i = \{ \emptyset, E_2, \emptyset, E_4, \emptyset \} \), and \( E_4^i = \{ \emptyset, E_2, E_3, \emptyset, E_5 \} \). Therefore, gathering 3 shares among \( E_j^i \) \((1 \leq j \leq 4)\) acquires all \( E_j^i (1 \leq i \leq 5) \) for the further recovery processing. Furthermore, size of the shared images is determined by the number of elements containing 1 in each row of the \( N_{part} \) matrix. Therefore, size of shared image \( S_j \) is \( t \times r \times c \), in which \( t \) is equal to \( 1 + |N_{part}^i| \) and \( |N_{part}^j| \) denoted number of elements with 1 in the \( j \) rows of matrix \( N_{part} \).

The recovery algorithm of the proposed \((k, n, m)\) SRS is introduced as follows.

1. Collect \( k \) shared images and denote them by \( S_j (1 \leq j \leq k) \).
2. Extract each shared image \( S_j \) with size \( r \times c \) to acquire \( SR_j (1 \leq j \leq k) \).
3. Generate an initial matrix \( N \) and sharing matrix \( P_j (1 \leq j \leq k) \) through Equations (8)–(12).
4. Generate a randomlike image \( Z \) with size \( r \times c \) by the function \( \text{image synthesis}(b, r, c) \) defined in Equation (14), in which the seed \( b \) is the same seed as used to generate the exchanged matrix in sharing matrix generation step of Equation (11).
5. Use Equation (26) to acquire \( SR_0 \):

\[ SR_0(x, y) = SR_j(x, y), \text{ if } P_j(x, y) = 1, \]  

(26)
where $1 \leq x \leq r$, $1 \leq y \leq c$ and $1 \leq j \leq k$.

6. Apply XOR operation between the $SR_0$ and the randomlike image $Z$ to acquire random image $I_{\text{XOR}}$, i.e.,

$$I_{\text{XOR}} = SR_0 \bigoplus Z.$$  \hspace{1cm} (27)

7. Apply Equations (14) and (15) with $I_{\text{XOR}}$ to acquire $h$, $R$, and $R_j(1 \leq i \leq m)$.

8. Apply the initial matrix $N$ and coefficient $h$ with Equation (23) to acquire the matrix $N_{part}$.

9. Split the remaining shared image $S_j$ in Step 2 to groups of $r \times c$ matrices and assign each $r \times c$ share to $E_{ij}$ ($1 \leq i \leq m$) by Equation (28):

$$E_{ij} = \begin{cases} \text{new } r \times c \text{ matrix}, & \text{if } N_{part}(j, i) = 1, \\ \emptyset, & \text{if } N_{part}(j, i) = 0. \end{cases} \hspace{1cm} (28)$$

10. Replace $E_i$ by extracting the nonempty $r \times c$ matrix to share $E_{ij}$ ($1 \leq j \leq k$). Moreover, two nonempty elements among $r \times c$ share $E_{ij_1}$ and $E_{ij_2}$ ($j_1 \neq j_2$) should be identical.

11. Apply $E_i$ and $R_i$ to Equation (29) for covering the secret images $I_{re_i} (1 \leq i \leq m)$, i.e.,

$$I_{re_i} = \begin{cases} E_1 \oplus R_1 \oplus R_2, & i = 1, \\ E_2 \oplus R_2 \oplus R_3, & i = 2, \\ E_3 \oplus R_3 \oplus R_4, & i = 3, \\ \vdots & \\ E_{m-1} \oplus R_{m-1} \oplus R_m, & i = m - 1, \\ E_m \oplus R_m \oplus R_1, & i = m. \end{cases} \hspace{1cm} (29)$$

In Step 10, if two nonempty $r \times c$ sharing $E_{ij_1}$ and $E_{ij_2}$ ($1 \neq j \neq 2$) are not identical, at least one of the $r \times c$ shares $E_{ij_1}$ or $E_{ij_2}$ is maliciously modified.

### 3.4. Examples of Sharing and Recovery of the Proposed SRS with Thresholds (2, 3, 2)

In this section, examples of the (2, 3, 2) SRS sharing and recovery processes are introduced. The first way is to generate sharing matrix by the predefined (2, 3, 2) thresholds as follows.

1. Use Equation (8) to generate matrix $M = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ by threshold $k = 2$, $n = 3$.

2. Apply matrix $M$ to Equation (9) for acquiring matrix $N$, i.e.,

$$N = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$  \hspace{1cm} (29)

3. Acquire matrices $N_1$, $N_2$, and $N_3$ by applying matrix $N$ to Equation (10), i.e.,

$$N_1 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}_{256 \times 3}, N_2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ \vdots \end{bmatrix}_{256 \times 3}, N_3 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}_{256 \times 3}.$$  \hspace{1cm} (29)

4. Acquire exchanged matrix $N'_1$, $N'_2$, and $N'_3$ by applying matrices $N_i (1 \leq i \leq 3)$ to Equation (11), i.e.,

$$N'_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ \vdots \end{bmatrix}_{256 \times 3}, N'_2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}_{256 \times 3}, N'_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ \vdots \end{bmatrix}_{256 \times 3}.$$  \hspace{1cm} (29)
Acquire sharing matrices $P_1$, $P_2$, and $P_3$ by applying matrices $N'_i (1 \leq i \leq 3)$ to Equation (12):

\[
N'_1 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad N'_2 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad N'_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}
\]

Figure 7 shows an example of the proposed SRS method with thresholds (2, 3, 2). In the proposed (2, 3, 2) method, 2 secret images are shared among 3 shared images with XOR operations and gathering 2 of these 3 shared images perfectly recover these 2 secret images. Figure 7a,b shows two secret images with size 256 \times 256 and Figure 7c,d shows two encrypted images in Step 8 of sharing algorithm. Figure 7e–g shows three generated shared images with size 256 \times 512, 256 \times 512, 256 \times 768, respectively. The recovery of acquiring two secret images should collect 2 shared images. Applying Figure 7e,f to the proposed recovery algorithm acquires 2 secret images, as shown in Figure 7a,b.

The sharing processes of the proposed (2, 3, 2) SRS method is introduced as follows.

1. Apply secret images $I_1$ and $I_2$ to Equation (13) for obtaining $I_{XOR}$.
2. Generate randomlike image $Z$ by Equation (14).
3. Generate randomlike image $SR_0$ by Equation (21).
4. Generate the sharing matrices $P_i (1 \leq i \leq 3)$ above by thresholds (2, 3, 2) with size 256 \times 256 by Equations (8)--(12).
5. Applying $SR_0$ and $P_i (1 \leq i \leq 3)$ to Equation (22) acquires $SR_i (1 \leq i \leq 3)$.
6. Apply $I_{XOR}$ and Equation (14) to generate the random image $R$.
7. Generate randomlike images $R_1$ and $R_2$ from the random image $R$ by Equation (15).
8. Apply $I_1$, $I_2$ and $R_1$, $R_2$ by Equation (16) to obtain encrypted images $E_1$, $E_2$, as shown in Figure 7c,d, respectively.
9. Randomly select two columns of $N = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ to acquire matrix $N_{part} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$.
10. Apply $N_{part}$ and $E_1$, $E_2$ to Equation (24) to generate $E'_1 = \{\emptyset, E_2\}$, $E'_2 = \{E_1, \emptyset\}$, $E'_3 = \{E_1, E_2\}$.
11. Acquire shared images by $S_1 = [SR_1 \parallel E_2]$, $S_2 = [SR_2 \parallel E_1]$, $S_3 = [SR_3 \parallel E_1 \parallel E_2]$, as shown in Figure 7e–g, respectively.
The recovery processes of the proposed (2, 3, 2) SRS method is introduced as follows.

1. Collect two shared images $S_1$ and $S_2$.
2. Extract $SR_1, SR_2$ from $S_1, S_2$ with size of $256 \times 256$.
3. Generate the initial matrix $N = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ and sharing matrix $P_1, P_2, P_3$ from Equations (8)–(12).

4. Generate a randomlike image $Z$ with size $r \times c$ by Equation (14).
5. Use Equation (26) with $P_1, P_2$ and $SR_1, SR_2$ to recover $SR_0$.
6. Use Equation (27) with $SR_0$ and $Z$ to acquire $I_{XOR}$.
7. Apply Equations (14) and (15) with $I_{XOR}$ to acquire $R_1$ and $R_2$.
8. Acquire the matrix $N_{part} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.
9. Assign the remaining part of shared images $S_1$ and $S_2$ to $E_1$ and $E_2$, respectively.
10. Apply Equation (29) with $R_1, R_2$ and $E_1, E_2$ to recover secret images $I_1, I_2$.

4. Experimental Results and Discussions

This section demonstrates the experimental results of our proposed $(k, n, m)$ approach. Section 4.1 presents experimental results of the proposed approach with different thresholds and recovery of attacked shared images. All experiments are performed on a PC with an Intel i7-4720HQ CPU, and 16 G RAM, using MATLAB 9.10 software. The size of all secret images is $256 \times 256$. Section 4.2 presents the performance of the proposed approach with comparing other multiple secret image sharing methods.
4.1. Experimental Results

Figures 8 and 9 show the experimental results of two proposed schemes, the SRM and SRS methods, by sharing with thresholds \((k, n, m) = (3, 4, 5)\) for sharing 5 secret images among 4 shared images and collecting 3 shared images that perfectly recover these 5 secret images. Figure 8 shows the results of our SRM method presented first. Figure 8a–e shows the 5 secret images: Lenna, Baboon, House, Cameraman, and Barbara with size \(256 \times 256\). Figure 8f–i shows 4 shared images with size \(256 \times 1280\), in which each shared image is the combination of five parts for every secret image with size \(256 \times 256\). Therefore, the size of the shared images is proportional to the number of secret images as 5 times \(256 \times 256\) in this example. Two kinds of recovered secret images are provided. Figure 8j–n shows 5 recovered secret images from Figure 8f–h, and Figure 8o–s shows 5 recovered secret images from the Figure 8f,g collection. Figure 8f–h fits the recovery threshold \(k = 3\) for perfectly acquiring the images in Figure 8j–n that are identical to the original secret images. However, the collection shown in Figure 8f,g cannot satisfy the recovery threshold \(k = 3\) and the recovered secret images all appear as noise images because of the failed acquisition of \(I_{\text{XOR}}\) in Equation (19). Moreover, the proposed SRM method uses a sharing matrix to assign part of pixels in shared images to 0, as defined in Equation (17). Therefore, the shared images all appear darker than the random images. Moreover, because of the large usage of 0 pixels, some kinds of regularity can also be found.

Figure 9 shows the results of our second proposed SRS method. Figure 9a–e shows the same 5 secret images with size \(256 \times 256\). Figure 9f–i shows 4 shared images of the parameter \(b = 1\) in step 2 of sharing algorithm and generating matrix of size \(4 \times 5\) as

\[
N_{\text{part}} = \begin{bmatrix}
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 
\end{bmatrix}
\]

in Equation (23). The first and third elements in the first row of the matrix \(N_{\text{part}}\) exhibit the element 1 and it leads the shares of \(E_1\) and \(E_3\) as defined in Equation (24). These two shares with the sharing of \(SR_1\) lead to the shared image with a size that is three times that of the secret image. Therefore, the first shared image is shown in Figure 9f with the size \(256 \times 768\), which is three times the secret image size \(256 \times 256\). Moreover, the third row of \(N_{\text{part}}\) leads to the share of \(E_3, E_4,\) and \(E_5\) that are needed. Consequently, four times the secret image size is obtained, as shown in Figure 9h, with the size \(256 \times 1024\). Therefore, sizes of shared images are according to the number of elements with 1, as shown in Figure 9f–i, being \(256 \times 768, 256 \times 768, 256 \times 1024,\) and \(256 \times 1024\), respectively. In the recovery process, collecting \(k = 3\) number of shared images is needed for perfect recovery. Collecting shared images shown in Figure 9f–h recovers the reconstructed images, as shown in Figure 9j–n. The collected shared images less than \(k\), as shown in Figure 9g,i, recover the reconstructed images, as shown in Figure 9o–s, in which all recovered images appear as noise images.

Table 1 shows the computation time of the two proposed schemes, SRM and SRS, among three different thresholds. The two proposed XOR-based schemes require limited CPU computation time, and the performances of the proposed schemes are exhibited. Moreover, the computation complexities between the two proposed schemes, SRM and SRS, are different. In our first proposed SRM method, the complexity of the sharing and recovery algorithms are \(O(nm)\) and \(O(km)\), respectively. Furthermore, the sharing and recovery complexities of our second SRS method are \(O(n)\) and \(O(k)\), respectively.
Figure 8. Results of the first proposed (3, 4, 5) SRM method. (a–e): 5 secret images with size 256 × 256; (f–i): 4 shared images with size 256 × 1280; (j–n): 5 recovered secret images from (f–h); (o–s): 5 recovered secret images from (f, g).
Axioms 2022, 11, x FOR PEER REVIEW 15 of 21

Axioms 2022, 11, x FOR PEER REVIEW 15 of 21

age with a size that is three times that of the secret image. Therefore, the first shared image is shown in Figure 9f with the size $256 \times 768$, which is three times the secret image size $256 \times 256$. Moreover, the third row of $N_{\text{par}}$ leads to the share of $E_3$, $E_4$, and $E_5$ that are needed. Consequently, four times the secret image size is obtained, as shown in Figure 9h, with the size $256 \times 1024$. Therefore, sizes of shared images are according to the number of elements, as shown in Figure 9f–i, being $256 \times 768$, $256 \times 768$, $256 \times 1024$, and $256 \times 1024$, respectively. In the recovery process, collecting $k = 3$ number of shared images is needed for perfect recovery. Collecting shared images shown in Figure 9f–h recovers the reconstructed images, as shown in Figure 9j–n. The collected shared images less than $k$, as shown in Figure 9g, i, recover the reconstructed images, as shown in Figure 9o–s, in which all recovered images appear as noise images.

Figure 9. Results of the second proposed (3, 4, 5) SRS method. (a–e): 5 secret images with size $256 \times 256$; (f–g): 2 shared images with size $256 \times 768$; (h,i): 2 shared images with size $256 \times 1024$; (j–n): 5 recovered secret images from (f–h); (o–s): 5 recovered secret images from (g,i).

Table 1. Performance of sharing and recovery procedures at different thresholds.

| Thresholds | Sharing Computation Time (s) | Recovery Computation Time (s) |
|------------|------------------------------|------------------------------|
| $k, n, m$  | SRM | SRS | SRM | SRS |
| (2, 3, 4)  | 1.123222 | 0.377484 | 0.419478 | 0.333834 |
| (3, 4, 5)  | 1.873811 | 0.521364 | 0.583452 | 0.415034 |
| (3, 5, 3)  | 1.222763 | 0.539960 | 0.374655 | 0.332443 |

Since our second proposed SRS method outperforms our first proposed SRM method in both less computation time and smaller size of shared images, more experimental results for our second proposed SRS method are illustrated. Figure 10 shows experimental results of the proposed (3, 5, 3) SRS method by the shared images with homogeneous regions.
Figure 10a–c are three generated secret images, which are constructed by simple color regions with size $256 \times 256$. The generated $N_{\text{part}}$ is defined by:

\[
\begin{bmatrix}
0 & 0 & 1 \\
1 & 1 & 0 \\
1 & 0 & 1 \\
1 & 1 & 1 \\
0 & 1 & 0
\end{bmatrix}
\]

with size $5 \times 3$, the size that is determined by $n \times m$ under thresholds $(k, n, m)$. From definition of the first row in $N_{\text{part}}$, the first shared image, shown in Figure 10d, includes the random image $SR_1$ and one share of the third secret image from Figure 10c. Therefore, the size of the first shared image is twice the secret images at $256 \times 512$. Moreover, Figure 10e shows the second shared image of the combination of the random image $SR_2$ and two shares of the first and second secret images from Figure 10a,b, respectively. Therefore, the size of the second shared image is three times that of the secret images at $256 \times 768$. At last, the fourth shared image, shown in Figure 10g with size $256 \times 1024$, includes the random image $SR_4$ and three shares of the first, second, and third secret images from Figure 10a–c, respectively. Therefore, three different kind of sizes are found in the proposed $(3, 5, 3)$ SRS method, as shown in Figure 10d–h, and these shared images all exhibit randomlike images. Two kinds of recovery results are also provided. Figure 10i–k are the recovered secret images from Figure 10d–f. Figure 10l–n are the recovered secret images from Figure 10g,h.

Since the collected number of shared images is less than threshold $k$, the recovered secret images are all noiselike images that exhibit the threshold recovery property of our proposed scheme.

![Secret Images](image_url)

**Figure 10.** Results of the second proposed $(3, 5, 3)$ SRS method. (a–c): secret images with size $256 \times 256$; (d): the first shared image with size $256 \times 512$; (e,f): the second and third shared images with size $256 \times 768$; (g): the fourth shared image with size $256 \times 1024$; (h): the fifth shared image with size $256 \times 512$; (i–k): recovered secret images from (d–f); (l–n): recovered secret images from (g,h).

Figure 11 shows the recovery results of applying attacks on shared images of homogeneous secret images. Figure 11a–e exhibit five shared images, which are the same as...
Figure 10d–h as the shared images of Figure 11a–c. Figure 11f shows a malicious attack result on Figure 11a by replacing one small black block in the left area. Figure 11g–i shows the recovered shared image from Figure 11a–c. The recovery of Figure 11b,c,f is shown in Figure 11j–l, which shows that any attack leads to false recovery because of the incorrect random image generated. Therefore, security of the proposed scheme is identified.

![Figure 11](image-url)  
Figure 11. Results of the recovery of attacked shared images in the proposed (3, 5, 3) SRS method. (a–e): 5 shared images as in Figure 10d–h; (f) maliciously attacked shared image of (a); (g–i): 3 recovered secret images from (a–c); (j–l): 3 recovered secret images from (f,b,c).

4.2. Comparison and Discussion

This section compares the proposed method with other stacking or XOR-based secret image sharing methods. The comparison metrics include lossless recovery or not, recovery strategy, sharing threshold, multiple sharing or not, and sharing capacity. The stacking-based schemes generate visual similarity on recovered secret images. Therefore, only stacking-based schemes are not lossless recovery and other XOR-based methods can recover without loss all secret images. In sharing thresholds, the method of Kabirirad and Eslami [24] satisfies restrictively consecutive \( k \) shared images and only adopt sharing matrix strategies that satisfy the general \((k, n)\) threshold property with \( k \leq n \). Other methods, such as stacking-based or XOR-based, only have \((2, 2)\) or \((n, n)\) thresholds. It should be noted that many related methods cannot solve the way to construct multi-secret image sharing with general \((k, n)\) thresholds; however, the proposed multiple sharing property exhibits the structure of sharing several secret images simultaneously. In a Boolean-based \((n, n)\) secret image sharing method, all the secret and shared images are the same size. The sharing capacity is the measurement of dividing the number of secret images by the number of shared images required to recover secret images. Therefore, in a Boolean-based \((k, n)\) secret image sharing method, the sharing capacity is extended to dividing the size of secret images by the size of shared images required to recover secret images. Large sharing capacity means smaller shared image acquired. The sharing capacities of the proposed
SRM method and SRS method are $\frac{1}{k}$ and $\frac{n}{\sum_{i=1}^{k} u_i + 1}$ with $u_i < n$, respectively. Furthermore, the value $u_i$ is determined by the matrix $N_{part}$ as the number of elements with 1 in the corresponding row.

Table 2 shows that the proposed SRM and SRS methods are the only XOR and sharing matrix-based methods that can perform $(k, n)$ thresholds on multiple sharing with good sharing capacity.

Table 2. Comparisons between the two proposed approaches and related multiple secret image sharing methods.

|                  | Lossless Recovery | Recovery Strategy | Sharing Threshold | Multiple Sharing | Sharing Capacity |
|------------------|-------------------|-------------------|-------------------|------------------|------------------|
| Wu et al. [18]   | No                | Stacking          | (2, 2)            | No               | $\frac{2}{n+4}$  |
| Chen et al. [15] | No                | Stacking          | (2, 2)            | Yes              | $\frac{n}{n+4}$  |
| Shyu et al. [16] | No                | Stacking          | (2, 2)            | Yes              | $\frac{n}{2 \times (2 \times n)}$ |
| Lin et al. [17]  | No                | Stacking          | (2, 2)            | Yes              | $\frac{2}{n}$    |
| Wang et al. [6]  | Yes               | XOR               | $(n, n)$          | No               | $\frac{1}{n}$    |
| Chen and Wu [12] | Yes               | XOR               | $(n, n)$          | Yes              | $\frac{n-1}{n}$  |
| Chen and Wu [13] | Yes               | XOR               | $(n, n)$          | Yes              | $\frac{n}{n}$    |
| Chen et al. [21] | Yes               | XOR               | $(n, n)$          | Yes              | $\frac{n}{n}$    |
| Bao et al. [20]  | Yes               | sharing matrix    | $(k, n)$          | No               | $\approx \frac{2}{k}$ |
| Prasetyo and Jing [22] | Yes       | Chinese remainder theorem | $(n, n)$          | Yes              | $\frac{n}{n}$    |
| Chattopadhyay et. al. [23] | Yes | XOR               | $(n, n)$          | Yes              | $\frac{n}{n}$    |
| Kabirirad and Eslami [24] | Yes | XOR              | Consecutive $(k, n)$ | Yes              | $\frac{n}{n}$    |
| Our proposed SRM method | Yes | XOR with sharing matrix | $(k, n)$          | Yes              | $\frac{1}{k}$    |
| Our proposed SRS method | Yes | XOR with sharing matrix | $(k, n)$          | Yes              | $\frac{n}{\sum_{i=1}^{k} u_i + 1}$ |

Table 3 shows an example of sharing capacity comparisons between the two proposed methods and other related methods. Since some conventional methods [6,12,13,15–18,21] only have the $(n, n)$ thresholds; therefore, these methods are restricted to $(2, 2)$ and $(3, 3)$ sharing capacities. Moreover, only Bao et al. [20] and our two proposed methods satisfy the requirement of $(k, n)$ thresholds. Therefore, these three methods have $(2, 3)$ and $(3, 4)$ sharing capacities. Although the method of Bao et al. [20] can only share one image secretly, its sharing capacity is higher because of adopting an encryption method during the sharing process. Finally, the two proposed schemes are the only methods for sharing multiple images secretly. Moreover, the proposed SRS method outperforms the proposed SRM method on more randomly shared images and better sharing capacity.
Table 3. Sharing capacity comparisons among the proposed approaches and related multiple secret image sharing methods.

| Recovery Quality | (2, 2) Sharing Capacity | (2, 3) Sharing Capacity | (3, 3) Sharing Capacity | (3, 4) Sharing Capacity |
|------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Wu et al. [18]   | recognizable             | 0.25                     | ×                        |                          |
| Chen et al. [15] | recognizable             | 0.375                    | 0.5                      |                          |
| Shyu et al. [16] | recognizable             | 0.25                     | 0.25                     |                          |
| Lin et al. [17]  | recognizable             | 1                        | ×                        |                          |
| Wang et al. [6]  | 100%                     | 0.33                     | 0.25                     |                          |
| Chen and Wu [12] | 100%                     | 0.66                     | 0.75                     |                          |
| Chen and Wu [13] | 100%                     | 1                        | 1                        |                          |
| Bao et al. [21]  | 100%                     | ≈1                       | ≈0.66                    |                          |
| Our proposed SRM method | 100% | 0.5 | 0.33 | |
| Our proposed SRS method | 100% | 0.38 | |

5. Conclusions

In this study, our contribution is a novel framework that can share \( m \) secret images among \( n \) participants, and gathering \( k \) participants’ shared images perfectly recover these \( m \) secret images. We propose two XOR-based \( (k, n, m) \) multi-secret image sharing schemes. The first is the source-random-mixed (SRM) method, which mixes the random image and secret images to generate shares for each secret image. The second is the source-random-separate (SRS) method, which separates the random image and secret images for constructing the shares for each secret image. Both of the proposed schemes reach the general \( (k, n) \) threshold by the usage of a sharing matrix, which is a popular property in mathematical-calculation-based methods such as the Shamir–Lagrange-based method. The proposed schemes have the properties of lossless recovery on collecting \( k \) shared images and no useful information acquired on collecting less than \( k \) shared images. The proposed schemes outperform other multiple sharing methods on the sharing rate of larger than or equal to 1. Therefore, these two proposed schemes exhibit good improvements on sharing multiple secret images through \( (k, n) \) thresholds by using XOR and a sharing matrix based Boolean operations.

In this study, we propose two efficient XOR-based \( (k, n, m) \) multi-secret image sharing schemes; however, there are several challenges and limitations to the proposed approach. First, the size of all the secret images used in the proposed approach must be equal, the multiple shared images generated are affected by the randomly generated matrix, and the sizes of each shared image are likely to be different. Second, the performance is relatively low compared to the secret image sharing method that simply uses Boolean operations. Based on these challenges and limitations, how to improve our proposed SRS method to identical size and how to apply the proposed approach in various actual applications, such as share authentication [28,29] and e-voting [30], merit our future study.

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

Funding: This work was supported in part by Ministry of Science and Technology, Taiwan, Republic of China under Grants MOST-110-2221-E-032-027 and MOST-110-2221-E-032-032.
Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.
Data Availability Statement: Data is contained within the article.
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. 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]
5. 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]
6. Wang, D.; Zhang, L.; Ma, N.; Li, X. Two secret sharing schemes based on Boolean operations. *Pattern Recognit.* 2007, 40, 2776–2785. [CrossRef]
7. 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]
8. 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]
9. Fang, W.-P. Friendly progressive visual secret sharing. *Pattern Recognit.* 2008, 41, 1410–1414. [CrossRef]
10. Huang, C.-P.; Hsieh, C.-H.; Huang, P.S. Progressive sharing for a secret image. *J. Syst. Softw.* 2009, 83, 517–527. [CrossRef]
11. 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]
12. Chen, T.-H.; Wu, C.-S. Efficient multi-secret image sharing based on Boolean operations. *Signal Process.* 2011, 91, 90–97. [CrossRef]
13. Chen, C.-C.; Wu, W.-J. A secure Boolean-based multi-secret image sharing scheme. *J. Syst. Softw.* 2014, 92, 107–114. [CrossRef]
14. Wang, R.-Z.; Su, C.-H. Secret image sharing with smaller shadow images. *Pattern Recognit. Lett.* 2006, 27, 551–555. [CrossRef]
15. 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, 033018-1–033018-5. [CrossRef]
16. 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]
17. 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]
18. Wu, H.-C.; Chang, C.-C. Sharing visual multi-secrets using circle shares. *Comput. Stand. Interfaces* 2005, 28, 123–135. [CrossRef]
19. Naor, M.; Shamir, A. Visual Cryptography. *Proc. Int. Conf. Adv. Cryptol.* 1995, 1–12.
20. Bao, L.; Yi, S.; Zhou, Y. Combination of Sharing Matrix and Image Encryption for Lossless ((n, n))-Secret Image Sharing. *IEEE Trans. Image Process.* 2017, 26, 5618–5631. [CrossRef]
21. Chen, C.-C.; Wu, W.-J.; Chen, J.-L. Highly efficient and secure multi-secret image sharing scheme. *Multimed. Tools Appl.* 2016, 75, 7113–7128. [CrossRef]
22. Prasetyo, H.; Jing, G.J. A note on multiple secret sharing using Chinese remainder theorem and exclusive-or. *IEEE Access* 2019, 7, 37473–37497. [CrossRef]
23. Chattopadhyay, A.K.; Ghosh, D.; Maitra, P.; Nag, A.; Saha, H.N. A Verifiable (n, n) Secret Image Sharing Scheme Using XOR Operations. In Proceedings of the 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 8–10 November 2018; pp. 1025–1031.
24. Kabirirad, S.; Eslami, Z. (n, n)-multi secret image sharing scheme based on Boolean operations. *J. Vis. Commun. Image Represent.* 2018, 57, 39–47. [CrossRef]
25. 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]
26. Azza, A.A.; Lian, S. Multi-secret image sharing based on elementary cellular automata with steganography. *Multimed. Tools Appl.* 2020, 79, 21241–21264. [CrossRef]
27. Chen, T.-H.; Wu, X.-W. Multiple secret image sharing with general access structure. *Multimed. Tools Appl.* 2020, 79, 13247–13265. [CrossRef]
28. 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, 31, 2896–2908. [CrossRef]
29. Jiang, Y.; 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. [CrossRef]
30. 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]