Combination of Sharing Matrix and Image Encryption for Lossless (k, n)-Secret Image Sharing

Long Bao, *Student Member, IEEE*, Shuang Yi, *Student Member, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*— This paper first introduces a (k, n)-sharing matrix $S^{(k,n)}$ and its generation algorithm. Mathematical analysis is provided to show its potential for secret image sharing. Combining sharing matrix with image encryption, we further propose a lossless (k, n)-secret image sharing scheme (SMIE-SIS). Only with no less than k shares, all the ciphertext information and security key can be reconstructed, which results in a lossless recovery of original information. This can be proved by the correctness and security analysis. Performance evaluation and security analysis demonstrate that the proposed SMIE-SIS with arbitrary settings of k and n has at least five advantages: 1) it is able to fully recover the original image without any distortion; 2) it has much lower pixel expansion than many existing methods; 3) its computation cost is much lower than the polynomial-based secret image sharing methods; 4) it is able to verify and detect a fake share; and 5) even using the same original image with the same initial settings of parameters, every execution of SMIE-SIS is able to generate completely different secret shares that are unpredictable and non-repetitive. This property offers SMIE-SIS a high level of security to withstand many different attacks.**

***Index Terms*— Secret image sharing, visual cryptography, sharing matrix, image encryption.**

I. Introduction

SECRET image sharing is an interesting research topic in multimedia security society. Its function is to encrypt an original image into n different shares. Using k ($k \leq n$) or more shares can successfully reconstruct the original image. With less than k shares, any information of the original image can not be accessed. This unique and interesting function allows secret image sharing to be used in many fields such as general access structures [1], discrete memoryless network [2], visual authentication and identification [3], [4], data sharing [5], and so on.

Manuscript received September 30, 2016; revised May 18, 2017 and July 18, 2017; accepted July 30, 2017. Date of publication August 11, 2017; date of current version September 1, 2017. This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/016/2015/A1 and in part by the Research Committee at the University of Macau under Grant MYRG2014-00003-FST and Grant MYRG2016-00123-FST. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wen Gao. *(Corresponding author: Yicong Zhou.)*

The authors are with the Department of Computer and Information Science, University of Macau, Macau 999078, China (e-mail: yicongzhou@umac.mo).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2738561

Secret image sharing has brought attentions of many scientists [6]–[10]. It can be roughly divided into two categories: visual cryptography (VC) and polynomial-based secret image sharing (PSIS). VC comes from the method proposed by Naor and Shamir [11]. In VC, each share is separate, noise-like and transparent. When people overlay no fewer than k shares, the original information will be revealed. It simply performs secret image sharing using the property of human visual system [12], but it has several drawbacks. For example, it is only applicable for binary images. Its image shares are noisy images that may not be user-friendly and draw attentions of attackers. Its reconstructed image is of low quality. Because it has a large pixel expansion ratio, its image shares are at least two times larger than the original image. Thus, VC requires a significantly large transmission and storage costs [13]. To solve these problems, many VC methods have been developed [14], [15]. For example, the VC application has been extended to color images using error diffusion [6]; the noise-like image share can be transformed into a meaningful image share [16]–[18]; the quality of reconstructed images is improved to obtain the optimal contrast [19]; and the high pixel expansion ratio is also reduced by probabilistic VC [20] and random grid-based VC [21]–[23]. However, these VC methods still have some intrinsic defects such as a lossy reconstruction and a large expansion ratio [24]. The lossy reconstruction means that noticeable distortions exist between the reconstructed and original images. The high expansion ratio causes a high transmission cost, because each share has a much larger size than the original image [6], [17].

PSIS, on the other hand, was firstly proposed by Shamir [25]. Its core idea is to utilize the Lagrange interpolation to generate image shares and reconstruct the original image with a minimal number of shares. However, it has three problems: (1) the Lagrange interpolation requires a huge computation cost, especially in the reconstruction phase; (2) a successful reconstruction depends on enough image shares and the right orders; and (3) the results of the Lagrange interpolation are in a data range different from one of the original image. This will result in a possible data loss or distortion in the reconstruction phase. To address the third problem, Li *et al.* proposed a method to calculate the Lagrange interpolation in $GF(2^8)$ to achieve a lossless reconstruction [26]. Chen *et al.* used quadratic residues to obtain a

lossless PSIS [27] but their method is suitable only for $(k = 1,\ n = 2)$ secret sharing. This framework was improved later by Ulutas *et al.* as a generalized model [28]. However, the first two problems remain unsolved.

Different from existing VC and PSIS methods, this paper proposes a lossless (k, n)-secret image sharing scheme based on combination of sharing matrix with image encryption (SMIE-SIS). In SMIE-SIS, we first introduce a new (k, n)-sharing matrix. This sharing matrix is given a strictly mathematical definition, including several special properties that are vital for secret sharing. We also introduce a simple but efficient algorithm to generate the (k, n)-sharing matrices. Based on these sharing matrices, the proposed SMIE-SIS contains a chaotic-based encryption process and a sharing encoding algorithm. Its computation cost is much lower than the reconstruction phase of PSIS. SMIE-SIS also has a low expansion ratio close to 0.5, indicating that each image share is only half size of the original image. This low expansion ratio is beneficial to reduce the cost of transmission and storage. Meanwhile, the proposed SMIE-SIS is a generalized method for any settings of k and n and for various formats of original images such as binary, grayscale or color images. More importantly, non-duplicate property of SMIE-SIS allows users to generate various shares in each execution with the same input. SMIE-SIS also has the verification function to identify a fake share involved in the reconstruction phase, which is important for real applications.

The rest of this paper is presented as follows: Section II will introduce the definition of the (k, n)-sharing matrix and discuss its application to secret sharing. Section II-B will propose a fast generation algorithm of the (k, n)-sharing matrix. Section III will introduce a new lossless (k, n)-secret image sharing scheme based on combination of sharing matrix and image encryption. Section IV will provide the simulation results of image sharing. Section V will present its performance analysis. Section VI will show the security analysis. Finally, Section VII will reach a conclusion.

II. (k, n)-Sharing Matrix

This section first introduces the mathematical definition of the (k, n)-sharing matrix $(k \le n)$ and its generation algorithm. Then, an illustrative example will be given to describe the sharing matrix generation process in detail. Finally, some mathematical analysis and advantages of (k, n)-sharing matrix will be discussed.

A. Definition

Let $S^{(k,n)}$ be an $n \times w$ binary matrix, $S^{(k,n)}(i, j) \in \{0,\ 1\}$, where $1 \le i \le n$, $1 \le j \le w$. Randomly selecting any p rows of elements from matrix $S^{(k,n)}$ generates a $p \times w$ binary matrix $Z(s, j)$, where integers $1 \le p \le n$ and $1 \le s \le p$. If matrix $S^{(k,n)}$ satisfies three following conditions:

1) there is at least one "1" in each row in matrix $S^{(k,n)}$, namely

$$\sum_{j=1}^{w} S^{(k,n)}(i, j) \neq 0 \tag{1}$$

Fig. 1. The proposed (k, n)-sharing matrix generation algorithm.

2) there is at least one "1" in each column in matrix Z when $p \ge k$, namely

$$\sum_{s=1}^{p} Z(s, j) \neq 0 \tag{2}$$

3) there is at least one zero column in matrix Z when $p < k$, namely

$$\prod_{j=1}^{w}\left(\sum_{s=1}^{p} Z(s, j)\right) = 0 \tag{3}$$

where $\sum_{s=1}^{p} Z(s, j)$ calculates the sum of the j^{th} column in matrix Z and $\prod_{j=1}^{w}(.)$ is a successive multiplication function. Then, $S^{(k,n)}$ is called the (k, n)-sharing matrix.

For example: Eq. (4) is a $(3, 4)$-sharing matrix satisfying the conditions in Eqs. (1)-(3).

$$S^{(3,4)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

These conditions are also properties of the (k, n)-sharing matrix and make it suitable for secret sharing. They will be discussed later.

B. (k, n)-Sharing Matrix Generation

Here, we propose an algorithm to produce (k, n)-sharing matrices $S^{(k,n)}$.

To generate $S^{(k,n)}$, a straightforward way is to exclusively try all possible combinations of binary matrices, and determine each one whether it is a (k, n)-sharing matrix. However, finding all possible sharing matrices is tedious in practice and requires a significantly high computation cost.

Here, we introduce a simple but fast algorithm, as shown in Fig. 1, to generate the proposed (k, n)-sharing matrix $S^{(k,n)}$. The algorithm contains three main steps: initial matrix generation, matrix expansion and row extraction. The initial matrix generation is to create an initial matrix S_0. Matrix expansion is an iterative process to extend S_0 into a new matrix S_e with a larger size. Row extraction is to generate the final (k, n)-sharing matrix $S^{(k,n)}$ by randomly selecting n rows from S_e.

1) Initial Matrix Generation: In initial matrix generation, we first construct a matrix M_1 with size of $(2k - 2) \times 1$. M_1 contains $(k - 1)$ ones and $(k - 1)$ zeros. For example, $M_1 = [1\ 1\ 1\ 0\ 0\ 0]^T$ for $k = 4$. We obtain all possible permutations of M_1, denoting M_i, $i = 2, \ldots, \mathcal{N}$, where $\mathcal{N} = \frac{(2k-2)!}{(k-1)!(k-1)!}$ is the number of total possible permutations of M_1. These matrices are concatenated together to generate the initial matrix S_0 with size of $(2k - 2) \times \mathcal{N}$, as shown in Eq. (5).

$$S_0 = [M_1, M_2, \ldots, M_{\mathcal{N}}] \tag{5}$$

2) Matrix Expansion: Using the initial matrix S_0, matrix expansion is to generate a new matrix S_e with a large size according to the value of n. The iteration number equals to $\mathcal{I} = \max\{\lceil \log_2(n/(2k-2))\rceil, 0\}$, where $\lceil x \rceil$ is a ceiling function to obtain the least integer greater than or equal to x. Note that S_0 is also a (k, n)-sharing matrix where $n = 2k-2$. When $n \leq 2k-2$, iteration number $\mathcal{I} = 0$, no expansion process is needed, and $S_e = S_0$. When $n > 2k-2$, we expand S_0 to generate a new matrix S_e using self-repeating. In the $i^{th}(1 \leq i \leq \mathcal{I})$ iteration, matrix S_{i-1} is equally divided into $k-1$ sub-matrices in the vertical direction. Then, in S_{i-1}, the $j^{th}(1 \leq j \leq k-1)$ "1" of each column will be replaced by the j^{th} sub-matrix, while any "0" will be replaced by an all-one matrix with the same size of each sub-matrix. By this self-repeating process, the size of matrix S_e is $((2k-2)*2^{\mathcal{I}}) \times \mathcal{N}_s$, where $\mathcal{N}_s = \mathcal{N}^{\mathcal{I}+1}$.

3) Row Extraction: According to a random sequence or users' setting, row extraction is to randomly select n rows of elements from matrix S_e to generate the final (k, n)-sharing matrix $S^{(k,n)}$ with size of $n \times \mathcal{N}_s$.

C. An Illustrative Example

As an illustrative example, we use the proposed generation algorithm to produce a $(3, 5)$-sharing matrix $S^{(3,5)}$. Then, $k = 3$, $n = 5$, $\mathcal{N} = \frac{(2\times3-2)!}{(3-1)!(3-1)!} = 6$, $\mathcal{I} = \max\{\lceil log_2(5/4)\rceil, 0\} = 1$. Thus, the sizes of matrices S_0 and S_e are 4×6 and 8×36, respectively.

In the initial matrix generation, we have $M_1 = [1\ \ 1\ \ 0\ \ 0]^T$ with size of 4×1. Matrix M_1 contains 2 ones and 2 zeros. After a permutation process, we obtain other five permutated matrices M_2, M_3, M_4, M_5, and M_6. The following lists all six matrices.

$$M_1 = \begin{bmatrix}1\\1\\0\\0\end{bmatrix} M_2 = \begin{bmatrix}1\\0\\1\\0\end{bmatrix} M_3 = \begin{bmatrix}0\\1\\1\\0\end{bmatrix}$$
$$M_4 = \begin{bmatrix}1\\0\\0\\1\end{bmatrix} M_5 = \begin{bmatrix}0\\1\\0\\1\end{bmatrix} M_6 = \begin{bmatrix}0\\0\\1\\1\end{bmatrix}$$

According to Eq. (5), these six matrices are then concatenated together to form matrix S_0 with size of 4×6, as presented by

$$S_0 = \begin{bmatrix}1&1&0&1&0&0\\1&0&1&0&1&0\\0&1&1&0&0&1\\0&0&0&1&1&1\end{bmatrix} \tag{6}$$

Notice that S_0 is actually a $(3, 4)$-sharing matrix. Because $\mathcal{I} = 1$, we need to apply 1 round of matrix expansion procedure to S_0 to generate an expanded new matrix S_e. As shown in Eq. (7), as shown at the bottom of this page, S_e is with size of 8×36.

Finally, we randomly select five rows of elements from S_e to obtain the final $(3, 5)$-sharing matrix $S^{(3,5)}$ as shown in Eq. (8), as shown at the bottom of this page. Thus, $S^{(3,5)}$ is a size of 5×36.

D. Discussion

Here, we discuss the proposed (k, n)-sharing matrix from various aspects, namely: ***Topic 1***: why does any matrix produced by the sharing matrix generation algorithm meet the definition of (k, n)-sharing matrix? ***Topic 2***: the advantages of proposed (k, n)-sharing matrix generation algorithm; ***Topic 3***: why is the proposed (k, n)-sharing matrix suitable for secret sharing? ***Topic 4***: why do we combine the sharing matrix and encryption to propose the new lossless (k, n)-secret image sharing scheme in the next section?

Topic 1: A matrix example produced by the sharing matrix generation algorithm has been given in the end of Section II-C. It is only a justification example to show that the generated matrix satisfies the three conditions of (k, n)-sharing matrix in Eqs. (1)-(3). Here, we will provide the mathematical justification to prove that the initial matrix S_0 and matrix S_k that is generated by randomly selecting k rows from matrix S_e are both (k, n)-sharing matrices.

The initial matrix S_0 with size of $(2k-2) \times \mathcal{N}$ is the combination M_1 and all its permutations. The number of "1"s in each row is the same. This means that each row of S_0 has $\frac{\mathcal{N}}{2}$ "1"s. When $k \geq 2$, we can obtain $\frac{\mathcal{N}}{2} = \frac{(2k-2)!}{2(k-1)!(k-1)!} \geq 1$. Thus, each row of S_0 contains at least one "1", which perfectly meets the first condition in Eq. (1). Each column of S_0 contains

$$S_e = \left[\begin{array}{cccccc:cccccc:cccccc:cccccc:cccccc:cccccc}
1&1&0&1&0&0&1&1&0&1&0&0&1&1&1&1&1&1&1&1&0&1&0&0&1&1&1&1&1&1&1&1&1&1&1&1\\
1&0&1&0&1&0&1&0&1&0&1&0&1&1&1&1&1&1&1&0&1&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1\\
\hdashline
0&1&1&0&0&1&1&1&1&1&1&1&1&1&0&1&0&0&1&1&1&1&1&1&1&1&0&1&0&0&1&1&1&1&1&1\\
0&0&0&1&1&1&1&1&1&1&1&1&1&0&1&0&1&0&1&1&1&1&1&1&1&0&1&0&1&0&1&1&1&1&1&1\\
\hdashline
1&1&1&1&1&1&0&1&1&0&0&1&0&1&1&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&0&0\\
1&1&1&1&1&1&0&0&0&1&1&1&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&0&1&0\\
\hdashline
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&0&0&1&0&1&1&0&0&1&0&1&1&0&0&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&1&1&1&0&0&0&1&1&1&0&0&0&1&1&1
\end{array}\right] \tag{7}$$

$$S^{(3,5)} = \left[\begin{array}{cccccccccccccccccccccccccccccccccccc}
1&1&0&1&0&0&1&1&0&1&0&0&1&1&1&1&1&1&1&1&0&1&0&0&1&1&1&1&1&1&1&1&1&1&1&1\\
1&0&1&0&1&0&1&0&1&0&1&0&1&1&1&1&1&1&1&0&1&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1\\
0&1&1&0&0&1&1&1&1&1&1&1&1&1&0&1&0&0&1&1&1&1&1&1&1&1&0&1&0&0&1&1&1&1&1&1\\
0&0&0&1&1&1&1&1&1&1&1&1&1&0&1&0&1&0&1&1&1&1&1&1&1&0&1&0&1&0&1&1&1&1&1&1\\
1&1&1&1&1&1&0&1&1&0&0&1&0&1&1&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&0&0
\end{array}\right] \tag{8}$$

$(k-1)$ "0"s and $(k-1)$ "1"s. When any p rows are selected, if $p \geq k$, the number of "1"s in each column will be at least $p-(k-1) = p-k+1 \geq 1$. This satisfies the second condition in Eq. (2) that there must be at least one "1" in each column of these selected rows. When $p \leq (k-1)$, there must be at least one column with all "0". Because the maximum number of "0"s in each column of these selected rows is $(k-1)$ and $p \leq (k-1)$, the third condition in Eq. (3) is also satisfied.

Next, we will prove that the matrix S_k generated from S_e is also a (k,n)-sharing matrix. Firstly, the matrix expansion is to extend the matrix S_0 to S_e by adding more "1"s to S_0. Each column of S_e also has $(k-1)$ "0"s and the rest are all "1"s. The matrix expansion process can be also considered as permutations of the first column of S_e. Each row of S_e contains $(\frac{\mathcal{N}}{2})^2$ of "0"s. The number of "1"s can be calculated by

$$\begin{aligned}\mathcal{N}_s - (\frac{\mathcal{N}}{2})^2 &= \mathcal{N}^{\mathcal{I}+1} - \frac{1}{4}\mathcal{N}^2, \quad \mathcal{I} \geq 1 \\ &\geq \frac{3}{4}\mathcal{N}^2, \quad \mathcal{N} \geq 2 \\ &\geq 3\end{aligned}$$

Hence, when selecting any k rows from S_e to form a new matrix S_k, there will be at least 3 "1"s in each row, which satisfied the first condition in Eq. (1). Randomly selecting p rows from S_k obtains the matrix Z. If $p = k$, each column of Z must have at least one "1" and less than $(k-1)$ of "0"s; if $p \leq k-1$, there must be one column in Z with all "0". Because the maximum number of "0"s in each column of Z is $k-1$. Thus, the second and third conditions are definitely satisfied.

Hence, we can conclude that the sharing matrices produced by the proposed generation algorithm in Fig. 1 are the (k,n)-sharing matrices.

Topic 2: The proposed sharing matrix generation method is an efficient generation algorithm for producing different (k,n)-sharing matrices. The proposed algorithm has, but not limited to, following advantages:

(1) Matrix S_0 produced in the initial matrix generation process is actually a (k,n)-sharing matrix where $n = 2k-2$.
(2) Replacing the initial matrix generation process with any existing generation method of the (k,n)-sharing matrix, the proposed algorithm becomes an extended and generalized version of the existing method.
(3) The proposed algorithm is able to generate different (k,n)-sharing matrices under any settings of k and n.
(4) With the same settings of k and n, the proposed algorithm is able to produce different (k,n)-sharing matrices when using different methods as initial matrix generation and various random selection strategies for row extraction.

Topic 3: Here, we provide the mathematical analysis of sharing and reconstruction processes in secret sharing to prove that why the proposed (k,n)-sharing matrix is suitable for image sharing. The analysis contains two parts:1) Sharing process and 2) Reconstruction process. For better understanding, an illustrative example is also given in each part of the analysis.

1) Sharing Process: Suppose we want to share an integer data P with size of $1 \times w$. Firstly, we construct a matrix P_1 by

$$P_1(i,:) = P, \quad i = 1, 2, \ldots, n \tag{9}$$

Given a sharing matrix S with size of $n \times w$, the sharing process then generates a matrix R with the same size of S using Eq. (10).

$$R = P_1 * S \tag{10}$$

where, $*$ is point-to-point multiplication in this section. In matrix R, each row is a share and totally we have n shares.

Here, we give an illustrative example to show the detailed procedures of sharing process.

Suppose an original data matrix $P = [123\ 45\ 63\ 79\ 1\ 22]$ is to be shared by a sharing matrix $S^{(3,4)}$ as defined by

$$S^{(3,4)} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{11}$$

According to Eq. (9),

$$P_1 = \begin{bmatrix} 123 & 45 & 63 & 79 & 1 & 22 \\ 123 & 45 & 63 & 79 & 1 & 22 \\ 123 & 45 & 63 & 79 & 1 & 22 \\ 123 & 45 & 63 & 79 & 1 & 22 \end{bmatrix} \tag{12}$$

Using Eq. (10),

$$R = P_1 * S^{(3,4)} = \begin{bmatrix} 123 & 45 & 0 & 79 & 0 & 0 \\ 123 & 0 & 63 & 0 & 1 & 0 \\ 0 & 45 & 63 & 0 & 0 & 22 \\ 0 & 0 & 0 & 79 & 1 & 22 \end{bmatrix} \tag{13}$$

where each row of R is a share, and totally we have 4 shares.

2) Reconstruction Process: In this phase, we need to prove that the data P can be successfully reconstructed only when $k_r (k_r \geq k)$ shares are combined. To mathematically show the process of combining k_r shares (i.e., k_r rows from R) in the reconstruction phase, we first generate a matrix R_m with the same size of R. R_m contains k_r rows of 1s and 0s in all the rest rows. Using R_m, we can select k_r rows of data from R to obtain matrix R_1.

$$R_1 = R * R_m = P_1 * S * R_m \tag{14}$$

The reconstruction process then generates a reconstructed matrix R_r by

$$\begin{aligned} R_r(j) &= R_1(1,j) \| R_1(2,j) \|, \ldots, \| R_1(n,j) \\ &= (P_1(1,j) * S(1,j) * R_m(1,j)) \| (P_1(2,j) * S(2,j) \\ &\quad * R_m(2,j)) \|, \ldots, \| (P_1(n,j) * S(n,j) * R_m(n,j)) \\ &= (P(j) * S(1,j) * R_m(1,j)) \| (P(j) * S(2,j) \\ &\quad * R_m(2,j)) \|, \ldots, \| (P(j) * S(n,j) * R_m(n,j)) \\ &= P(j) * [(S(1,j) * R_m(1,j)) \| (S(2,j) * R_m(2,j)) \\ &\quad \| , \ldots, \| (S(n,j) * R_m(n,j))] \\ &= P(j) * R_S(j) \end{aligned} \tag{15}$$

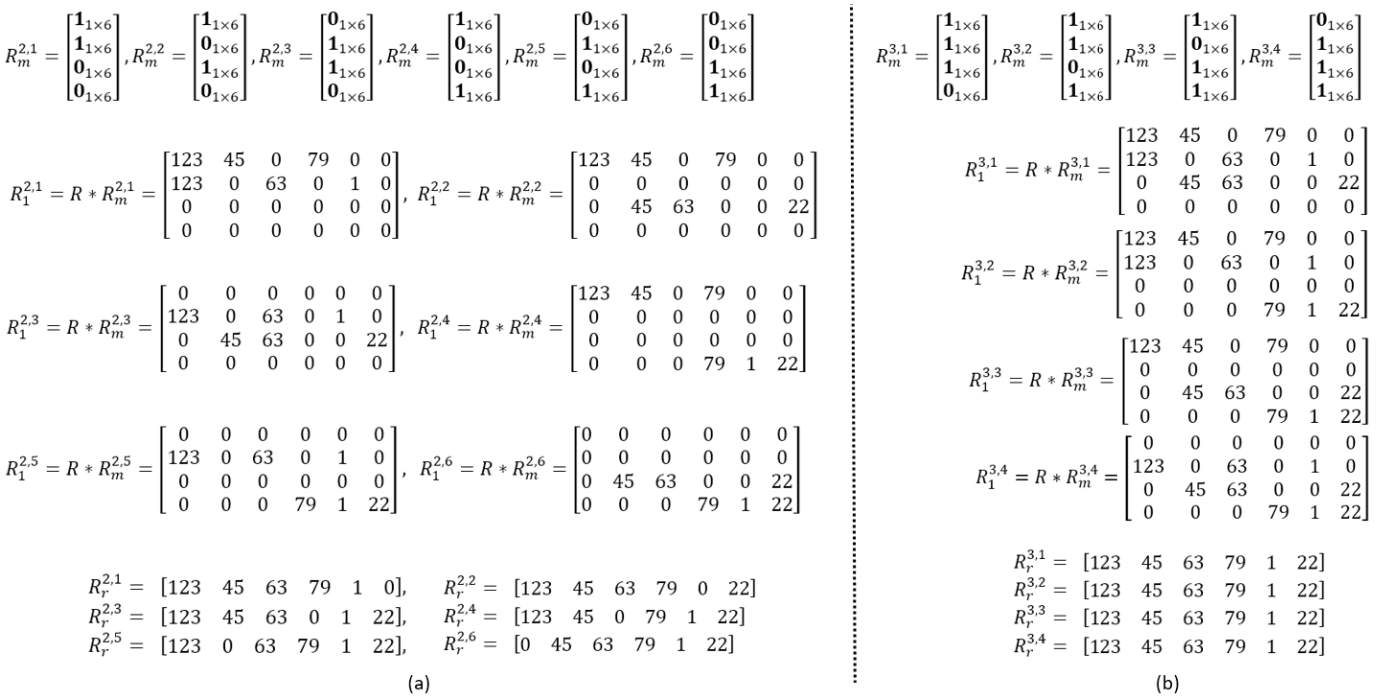


Fig. 2. Illustrative example of reconstruction processes when combining (a) two ($k_r = 2$) and (b) three ($k_r = 3$) shares, respectively.

where $j = 1, 2, \ldots, w$ and "$\|$" is the bit-level boolean function "or".

$$R_S(j) = (S(1, j) * R_m(1, j)) \| (S(2, j) * R_m(2, j)) \| \ldots \| (S(n, j) * R_m(n, j)) \quad (16)$$

where $R_S(j)$ is the result of selecting k_r rows in the j^{th} column of sharing matrix S. From the definition of the sharing matrix in Eq. (2), when $k_r \geq k$, there is at least one "1" in each column. This means that, for each j, there must be at least one element in Eq. (16) satisfying $S(i, j) * R_m(i, j) = 1$, $i \in 1, 2, \ldots, n$. Hence, $R_S(j) = 1$ and $R_r(j) = P(j)$. These demonstrate that the reconstructed R_r is the same as the original data P.

When $k_r < k$, from Eq. (3), for each j, there must be at least one element satisfying $S(i, j) * R_m(i, j) = 0$, $i \in 1, 2, \ldots, n$. Hence, there must be one j making $R_S(j) = 0$ and finally resulting in $R_r(j) = 0$. This demonstrates that the reconstructed $R_r \neq P$.

Thus, the (k, n)-sharing matrix is able to reconstruct the original data without any error when at least k shares are combined. This demonstrates that it is suitable for secret sharing.

Following the example in sharing process, we give an illustration in Fig. 2 to show the reconstruction process in detail. Here, $R_m^{k_r,j}$ means the j^{th} possible permutation of R_m when combining k_r shares. $\mathbf{1}_{1\times6}$ and $\mathbf{0}_{1\times6}$ are vectors with size of 1×6 and all "1"s and "0"s, respectively. Thus, when combining any 2 shares, 6 possible reconstruction results $\{R_r^{2,j}\}_{j=1}^{6}$ will be generated as shown in Fig. 2(a). Because the $(3, 4)$-sharing matrix is being used, combining any two shares is not sufficient to successfully reconstruct the original data. On the other hand, any three shares are able to reconstruct the original data matrix P without distortion (see the reconstruction results $\{R_r^{3,j}\}_{j=1}^{4}$ in Fig. 2(b)).

Topic 4: As analyzed in ***Topic 3***, the proposed (k, n)-sharing matrix $S^{(k,n)}$ is able to successfully recover the original data when at least k shares are combined. However, directly using $S^{(k,n)}$ for secret sharing may cause information leakage. This is because each generated share contains partial information of the original data (see the example in ***Topic 3***). This motivates us to propose the lossless (k, n)-secret image sharing scheme by combining of sharing matrix and image encryption (SMIE-SIS) (see Section III). And more discussions of about these can be found in security analysis in Section VI.

III. (k, n)-Secret Image Sharing Scheme by Combining the Sharing Matrix and Image Encryption

In this section, we propose a new (k, n)-secret image sharing scheme by combining the sharing matrix and image encryption (SMIE-SIS). SMIE-SIS encrypts the original image into a noise-like random sequence before secret sharing with hiding the "key" into secret share. Thus, each generated share is noise-like, and has no information leakage. In the reconstruction phase, only when the recovered encrypted image is identical to encrypted image before secret sharing, the correct decryption key can be successfully extracted to obtain the original image with right ciphertext. As an important property to ensure high security, even with the same original image and same settings of k and n, each execution of the proposed SMIE-SIS yields completely different, unpredictable, and non-repetitive secret shares.

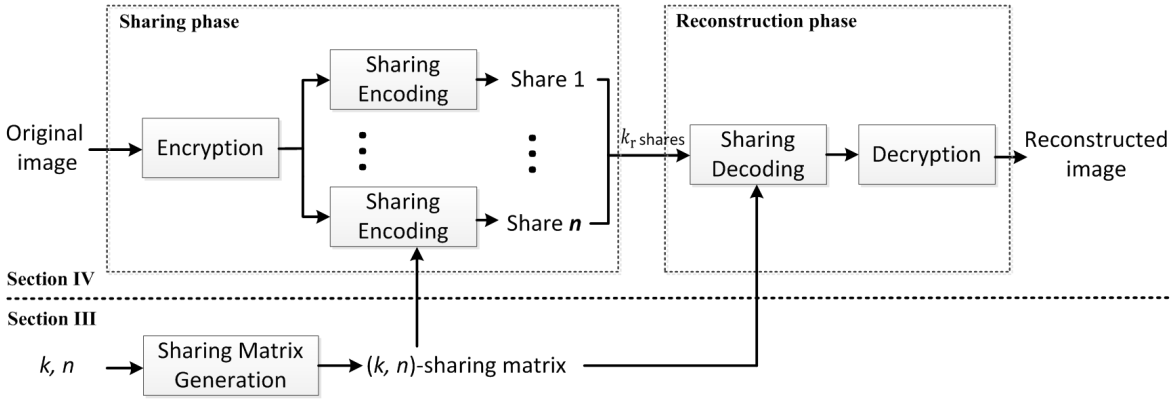


Fig. 3. The proposed SMIE-SIS.

A. SMIE-SIS

The flow chart of the proposed SMIE-SIS is shown in Fig. 3. SMIE-SIS first encrypts the original image using a substitution process, and utilizes a sharing encoding process to generate different shares. Combining k or more shares, the authorized users are able to completely reconstruct the original image.

1) Encryption: The encryption process is to transfer a $W \times L$ original image into a 1D noise-like data sequence for the subsequent sharing coding process. The Tent map used in Eq. (19) is to generate a random chaotic sequence. Due to the chaotic properties of the Tent map, a tiny change of its input or parameter yields a significant change in its output. Notice that the users have flexibility to choose another chaotic map to generate the random sequence.

Firstly, we use a random number generator to produce a security key $K = \{k_1, k_2, \ldots, k_{192}\}$, where $k_\psi \in [0, 1]$, $1 \le \psi \le 192$. Using K, two sets of initial parameters, $(r_1, C_1(1))$ and $(r_2, C_2(1))$, for the Tent map are generated using Eqs. (17) and (18).

$$r_x = \left(\prod_{i=1}^{4} a_i * 2^{48} + a_x\right) \mod 0.4 + 3.6 \tag{17}$$

$$C_x(1) = \left(\prod_{i=1}^{4} a_i * 2^{48} + a_{x+2}\right) \mod 1 \tag{18}$$

where $x = 1, 2$ and $a_i = \frac{\sum_{t=48i-47}^{48i} k_t * 2^{t-48i+47}}{2^{48}}$ for $i = 1, 2, 3, 4$.

Two random sequences C_1 and C_2 are then produced by Eq. (19)

$$C_x(y) = \begin{cases} C_x(1) & \text{if } y = 1 \\ \frac{1}{2} r_x C_x(y-1) & \text{if } y \neq 1 \ \& \ C_x(y-1) < 0.5 \\ \frac{1}{2} r_x (1 - C_x(y-1)) & \text{if } y \neq 1 \ \& \ C_x(y-1) \ge 0.5 \end{cases} \tag{19}$$

where $y = 1, 2, \ldots, W \times L$.

Scanning the original image O from left to right and then up to down, the original image is transformed into a 1D data matrix V. Applying the random sequences C_1 and C_2 to Eq. (20), the substitution process encrypts matrix V into a 1D matrix $E_2(j)$.

$$E_2(y) = \begin{cases} E_1(y) & \text{if } y = W \times L \\ \left(E_1(y) + \lfloor C_2(y) \times 10^{13} \rfloor + E_1(y+1)\right) \mod 256 & \text{otherwise} \end{cases} \tag{20}$$

where $\lfloor \cdot \rfloor$ and mod are the floor and modulo functions, and

$$E_1(y) = \begin{cases} V(y) & \text{if } y = 1 \\ \left(V(y) + \lfloor C_1(y) \times 10^{13} \rfloor + V(y-1)\right) \mod 256 & \text{otherwise} \end{cases} \tag{21}$$

Combining the encrypted data matrix E_2 with the security key K_s, we obtain the final encrypted data sequence E,

$$E = (K_s, E_2) \tag{22}$$

where $K_s = \langle K \oplus [\![\sum E_2]\!] \rangle$. Here $\oplus$ is the bitwise XOR and function $[\![*]\!]$ converts an integer into a binary sequence. Since K contains 192 bits, the summation value $\sum E_2$ of all pixels in E_2 is converted into 192 bits as well. Finally, function $\langle * \rangle$ converts the binary sequence into 24 integers in which each one is produced by 8 binary bits. Thus, the size of E is $1 \times (W \times L + 24)$.

2) Sharing Encoding: SMIE-SIS uses four main steps to perform sharing encoding.

The first step is to use the proposed generation algorithm in Section II-B to produce the (k, n)-sharing matrix $S^{(k,n)}$ with size of $n \times \mathcal{N}_s$. With the (k, n)-sharing matrix $S^{(k,n)}$, we repeat this sharing matrix to be the same size of encrypted data as the reference for sharing encoding.

For each encrypted data matrix E, if its corresponding value in the same position of sharing matrix is equal to one, it will be kept in the data sequence. But if this corresponding value in sharing matrix is equal to 0, this encrypted data in E will be deleted. By this referenced process, encoded matrix H^i will be generated.

After generating encoded matrix, all the important information will be fused into each 1D encoded share F^i. Each 1D

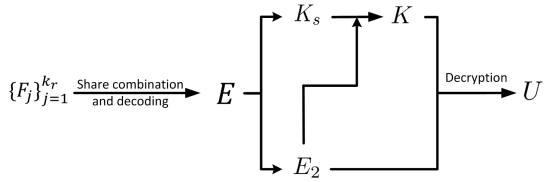


Fig. 4. Image reconstruction of SMIE-SIS.

encoded share F^i consists of three parts as defined by

$$F^i = \left(D, B^i, H^i\right) \tag{23}$$

where D is a 1D matrix with size of 1×2. It is utilized to store the value of $\mathcal{N}_s$ and calculated by

$$D(1) = \lfloor \frac{\mathcal{N}_s}{256} \rfloor \tag{24}$$

$$D(2) = \mathcal{N}_s \bmod 256 \tag{25}$$

and B^i is also a 1D matrix with size of $1 \times \lceil \frac{\mathcal{N}_s}{8} \rceil$, where $\lceil \cdot \rceil$ is the ceiling function. It converts $S(i,:)$ into a series of 8-bit binary sequences and stores them in a decimal format using Eq. (26).

$$B^i(j) = \sum_{j_1=8j-7}^{8j} \hat{S}(i, j_1) \times 2^{j_1-8j+7} \quad j = 1, 2, \ldots, \lceil \frac{\mathcal{N}_s}{8} \rceil \tag{26}$$

Here $\hat{S}(i,:) = (S(i,:), \mathbf{0}_{1\times z})$, and $\mathbf{0}_{1\times z}$ is a 1D zero matrix with size of $1 \times z$, where $z = f(\mathcal{N}_s, 8)$ and function $f(a,b)$ is defined by Eq. (27).

$$f(a,b) = \begin{cases} 0, & \text{if } a \bmod b = 0 \\ b - (a \bmod b), & \text{otherwise} \end{cases} \tag{27}$$

The previous data in process are all in 1D. Since the final output should be in 2D, a transformation from 1D to 2D will be applied to each 1D encoded share. To do the transformation, we first generate a 1D expanded share G^i using Eq. (28), and reshape G^i to obtain the final 2D image share with size of $W \times \lceil \frac{l_i}{W} \rceil$ where l_i is the length of the encoded share F^i.

$$G^i = \begin{cases} F^i & \text{if } (l_i \bmod W) = 0 \\ \left(F^i, \mathbf{0}_{1\times v}\right) & \text{otherwise} \end{cases} \tag{28}$$

where $\mathbf{0}_{1\times v}$ is a 1D zero matrix and $v = f(l_i, W)$.

Next, we will discuss how the authorized users reconstruct the original image using k or more image shares.

B. Image Reconstruction

To completely reconstruct the original image, the authorized users should receive k_r ($k_r \geq k$) image shares. The reconstruction procedure is unrelated to the orders of image shares. Each 2D share is transformed to a 1D sequence, and then divided into three parts: (1) the first two values to recover $\mathcal{N}_s$ using the inverse processes of Eqs. (24)-(25); (2) subsequent $\lceil \frac{\mathcal{N}_s}{8} \rceil$ integers to be transformed to a binary sequence and set to the i^{th} row of $S^{(k,n)}(i,:)$ using Eq. (26); and (3) the rest data. The procedure of image reconstruction can be shown in Fig. 4.

Using this recovered sharing matrix $S^{(k,n)}$ as a reference, the rest data of each received share $\{H_j\}_{j=1}^{k_r}$ will be combined together as matrix H_d to obtain their corresponding reconstructed encrypted matrix E_d.

Next, we decrypt E_d to obtain the reconstructed original image U. Matrix E_d is first divided into two parts: the first 24 integers K_s and the rest data E_2. Using Eq. (29), the encryption key K is reconstructed and used to generate two chaotic sequences C_1 and C_2. Using Eqs. (30)-(31) and transforming 1D sequence V to 2D, the original image U is finally reconstructed.

$$K = [\![K_s]\!] \oplus [\![\sum E_2]\!] \tag{29}$$

$$V(j) = \begin{cases} E_1(y), & \text{if } y = W \times L \\ \left(E_1(y) + \lfloor C_1(y) \times 10^{13} \rfloor + E_1(y+1)\right) \bmod 256, & \\ \qquad \text{otherwise} & \end{cases} \tag{30}$$

$$E_1(j) = \begin{cases} E_2(y), & \text{if } y = 1 \\ \left(E_2(y) + \lfloor C_2(y) \times 10^{13} \rfloor + E_2(y-1)\right) \bmod 256, & \\ \qquad \text{otherwise} & \end{cases} \tag{31}$$

If $k_r \geq k$, the reconstructed image U is the same as the original image; Otherwise, it will be a noise-like image, preventing information leakage.

C. Theoretical Security Analysis

The security of the proposed SMIE-SIS is theoretically analyzed in three aspects: (1) image encryption; (2) sharing process; (3) the combination of encryption and sharing to achieve "$1 + 1 > 2$".

The security of image encryption can be ensured by four factors: 1) large key space; 2) high diffusion property, 3) high confusion property, 4) non-duplicate property. The key space of the proposed SMIE-SIS is 2^{192}, which is larger than the basic key space requirement of 2^{100} [29]. This large key space provides the ability to defend the brute-force attack. The high confusion property can be ensured by the noise-like appearance of each share. From these noise-like shares, no visual information about original images can be disclosed. The high diffusion property can be verified by the high sensitivity of the proposed encryption part in SMIE-SIS to the plaintext and ciphertext. A tiny change in the plaintext or ciphertext will be spread over the whole encryption or decryption process to generate totally different results. The non-duplicate property allows that any two executions of image encryption generate different encrypted results, even with same inputs. This property solves the weakness of many existing image encryption methods and makes this encryption part to well defend some cryptanalysis attacks [30]–[33], such as differential attack and known-plaintext attack.

Based on these four properties, a mathematical proof of security can be presented in Eqs.(32)-(38), where $Pr[\ ]$ shows the probability and $Enc(o,k)$ represents the process of the introduced encryption part (Enc) with a key ($k \in K$) and an original image ($o \in O$).

For an arbitrary encrypted result $e \in E$ and original image $o \in O$

$$Pr[E = e|O = o] = \frac{Pr[E = e, O = o]}{Pr[O = o]} \tag{32}$$

Since $Enc(o, k) = e$ and $Enc(.)$ is a one-to-one function,

$$Pr[E = e, O = o] = Pr[K = k, O = o] \tag{33}$$

Since K and O are independent,

$$\begin{aligned} \therefore Pr[E = e, O = o] &= Pr[K = k, O = o] \\ &= Pr[K = k] \cdot Pr[O = o] \end{aligned} \tag{34}$$

$$\begin{aligned} \therefore Pr[E = e|O = o] &= \frac{Pr[K = k] \cdot Pr[O = o]}{Pr[O = o]} \\ &= Pr[K = k] \\ &= 2^{-192} \end{aligned} \tag{35}$$

where, the key K is a uniform 192-bit string.

$$\begin{aligned} Pr[E = e] &= \Sigma_{o \in O} Pr[E = e|O = o] \cdot Pr[O = o] \\ &= 2^{-192} \cdot \Sigma_{o \in O} Pr[O = o] \\ &= 2^{-192} \end{aligned} \tag{36}$$

According to Bayes' Theorem,

$$\begin{aligned} Pr[O = o|E = e] &= \frac{Pr[E = e|O = o] \cdot Pr[O = o]}{Pr[E = e]} \\ &= \frac{2^{-192} \cdot Pr[O = o]}{2^{-192}} \\ &= Pr[O = o] \end{aligned} \tag{37}$$

According to the definition of secure in [34], an encryption scheme (Gen, Enc, Dec) with message space O is perfectly secret if for every probability distribution over E, every message $o \in O$, and every ciphertext $e \in E$ for which $Pr[e = E] > 0$:

$$Pr[O = o|E = e] = Pr[O = o] \tag{38}$$

Hence, this encryption part has a certain security level to prevent original information leakage. Further, any existing image encryption algorithm with a high security level can be used in our SMIE-SIS. Hence, the security of SMIE-SIS can be improved by including the choice of image encryption algorithm [35]–[37].

The security provided by the sharing process is based on two facts: 1) each share contains only a part of encrypted information, and 2) the sharing process is able to detect a fake share. As analyzed in Section II-D, a true share contains only a part of encrypted information and cannot disclose all encrypted information. And, once a fake share is involved in the reconstruction phase, it will be quickly detected and located. Hence, only a combination of not less than k true shares can successfully reconstruct the original image. The ability of detecting a fake share will be further demonstrated by experimental results in Section VI-D.

From above analysis, we have carefully considered security issues in the image encryption and sharing processes. For the encryption part, a successful attack is to guess the key with a probability in Eq. (39).

$$Pr[Enc^{break}] = 2^{-192} \tag{39}$$

For the image sharing part, one efficient attack is to guess a whole true image share with a probability as shown in Eq. (40).

$$Pr[Sharing^{break}] = 256^{-(1+W \cdot L/2)} \tag{40}$$

Hence, the probability of a successful attack to the whole system is calculated in Eq. (41).

$$\begin{aligned} Pr[Success^{break}] &= Pr[Enc^{break}] \cdot Pr[Sharing^{break}] \\ &= 256^{-(1+W \cdot L/2)} * 2^{-192} \\ &= 2^{-4W \cdot L - 200} \end{aligned} \tag{41}$$

For an image with the size of $256 * 256$, the probability of a successful attack will be as shown in Eq. (42). Theoretically, this chance of a successful brute-force attack is too small to be zero. Combining the image encryption and sharing processes, SMIE-SIS can achieve a high level of security in a "$1+1 > 2$" fashion. Beside these theoretical analysis, some experimental security analysis are conducted in Section VI.

$$Pr[Success_{break}] = 2^{-262344} \tag{42}$$

D. Discussion

Different from existing methods of visual cryptography (VC) and polynomial-based secret image sharing (PSIS), SMIE-SIS has at least seven advantages:

(1) SMIE-SIS is a lossless secret image sharing system, namely the original image can be completely reconstructed without any distortion.
(2) SMIE-SIS is a generalized (k, n)-secret image sharing system with arbitrary k values.
(3) SMIE-SIS can be used for different types of images, such as binary, grayscale and color images.
(4) SMIE-SIS is a non-deterministic secret image sharing system. With the same original image and same parameter settings of k and n, SMIE-SIS is able to generate completely different unpredictable and non-repetitive shares in each execution of SMIE-SIS. This will provide a high level of security to withstand different attacks.
(5) SMIE-SIS has much lower pixel expansion than most existing methods. Thus, it has low costs of storage and transmission.
(6) SMIE-SIS has a computation cost much lower than PSIS.
(7) SMIE-SIS has the verification function to detect and locate a fake share.

IV. Simulation Results

To demonstrate the robustness of SMIE-SIS, Figs. 5-6 show several simulation results of binary, grayscale and color images using SMIE-SIS with different (k, n)-sharing matrices. As shown in Figs. 5-6, all the secret shares are noise-like image, protecting from information leakage. And the k can be set to 3, 4, 5, and other any number users want to use. Most importantly, with enough number of shares, the original image will be reconstructed without any data loss. When only less than k shares are available, the reconstructed images are noise-like. These demonstrate SMIE-SIS is able to achieve lossless secret sharing for different types of original images.

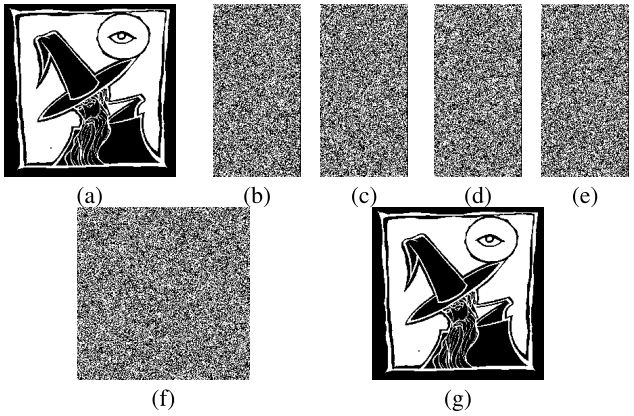


Fig. 5. Binary image security sharing using the proposed SMIE-SIS with the $(3, 4)$-sharing matrix. (a) is the original image with size of 256×256; (b)-(e) are four different Shares with size of 129×256; (f) and (g) are reconstructed images using any 2 and 3 shares, respectively.

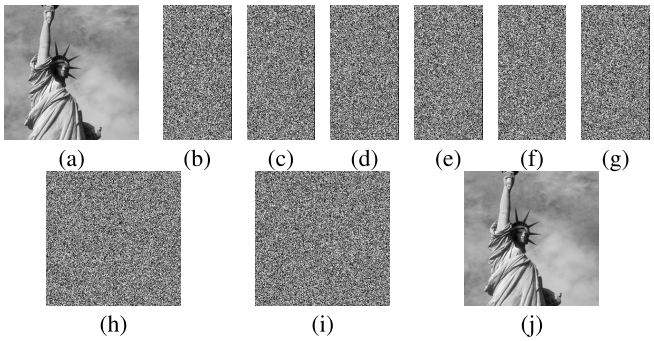


Fig. 6. Grayscale image security sharing using the proposed SMIE-SIS with the $(4, 6)$-sharing matrix. (a) is original image with size of 256×256; (b)-(f) are six different shares with size of 129×256; (h)-(j) are reconstructed images with any 2, 3 and 4 shares, respectively.

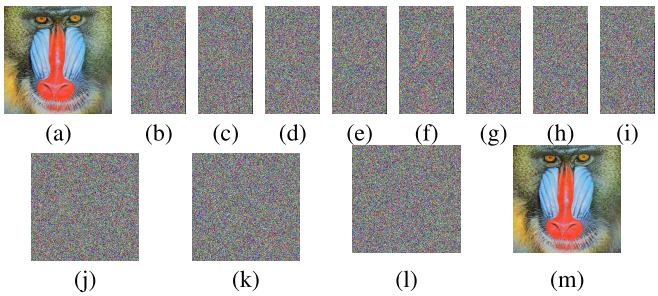


Fig. 7. Color image security sharing using the proposed SMIE-SIS with the $(5, 8)$-sharing matrix. (a) is original image with size of 256×256; (b)-(i) are eight different shares with size of 129×256; (j)-(m) are reconstructed image using 2, 3, 4 and any 5 shares, respectively.

V. Performance Analysis

Compared with existing CV and PSIS methods, the proposed SMIE-SIS has excellent performance with respect to distortion analysis, pixel expansion and computation cost.

A. Distortion Analysis

Distortion analysis aims to evaluate the differences between the original and reconstructed images. The analysis results are shown in Fig. 9. We apply SMIE-SIS and the VC method in [21] to a binary image in Fig. 9(a) and then use a sufficient number of image shares to reconstruct the image as shown in Fig. 9(b) and (c). The reconstructed image of the CV method contains huge background noise as shown in Fig. 9(b).

We then apply SMIE-SIS and an existing PSIS method in [25] to a grayscale image in Fig. 9(d) using the same threshold $k = 4$, and then reconstruct the image using any four image shares. The reconstructed results are shown in Fig. 9(e) and (f). For the PSIS method in [25], all pixel values larger than 255 in image shares are set to 255 when stored in a computer. This results in a data loss in the reconstruction phase as shown in Fig. 9(e). To solve this problem, two lossless schemes have been proposed to achieve no data loss by using quadratic residues [27], [28]. Since these lossless methods were designed only for the $(k = 1, n = 2)$ case, their reconstructed results is presented in Figs. 9 (f) and (g). These results are identical to the original image.

As shown in Figs. 9(c), (f), (g), and (h), the reconstruction results of the Chen's method [27], the Ulutas's method [28],

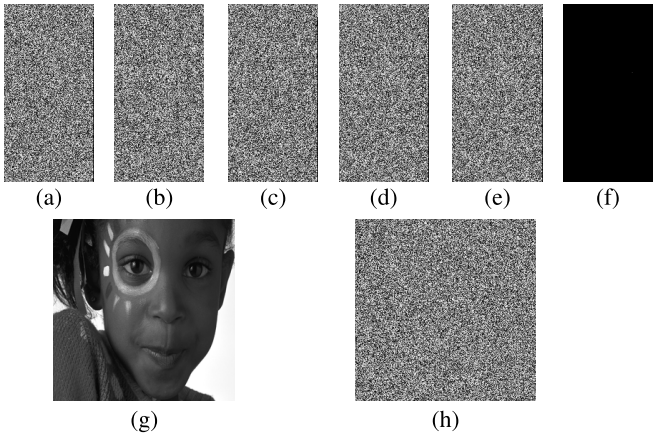


Fig. 8. Security analysis of SMIE-SIS. (a)-(d) show Shares 1-4 with a size of 129×256; (e) is a fake share of Share 4 with only one pixel change in position (100,100); (f) shows differences between Share 4 and the fake share; (g) is the reconstructed image with Shares 1-4 with a size of 256×256; (h) is the reconstructed image with Shares 1-3 and the fake share.

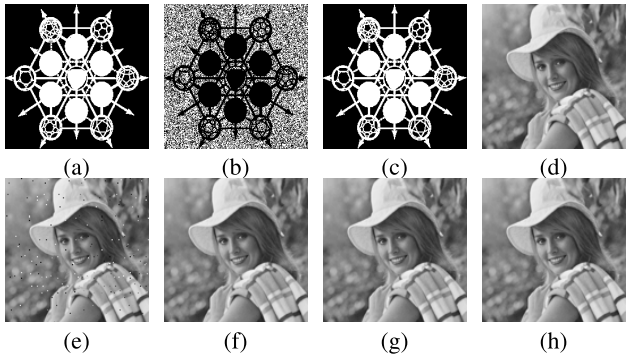


Fig. 9. Distortion Analysis. (a) is the original binary image; (b) is reconstruction of the CV method in [21]; (c) is reconstruction of SMIE-SIS; (d) is the original grayscale image; (e)-(h) are reconstruction results of the lossy PSIS method in [25], Chen's method in [27], Ulutas's method in [28], and SMIE-SIS, respectively.

TABLE I
PSNR AND SSIM RESULTS OF THE RECONSTRUCTED IMAGES

Original image	Methods	PSNR	SSIM
Fig. 9 (a)	CV [21]	1.3581	-0.1389
	SMIE-SIS	Inf	1
Fig. 9 (d)	PSIS in [25]	27.3974	0.8691
	Chen's [27]	Inf	1
	Ulutas's [28]	Inf	1
	SMIE-SIS	Inf	1

and the proposed SMIE-SIS have no visual distortion. These are also verified by the PSNR and SSIM measure results of the reconstructed images. A higher PSNR or SSIM value means better quality, and thus more similar to the original image. From the PSNR and SSIM results shown in Table I, the results of the CV method are quite low because its reconstructed images contain a lot of noise. The results of the PSIS method in [25] is slightly high because lossy PSIS recovers the original image with little noise. However, the results of the proposed SMIE-SIS, Chen's method [27], and Ulutas's method [28] reach the maximum values of PSNR and SSIM: Inf and 1, respectively. This means that the reconstructed image is the same as the original image. Therefore, SMIE-SIS is a lossless secret image sharing system and outperforms existing CV and PSIS method in [25]. It has the same lossless property as existing lossless PSIS methods.

B. Pixel Expansion

Pixel expansion is an important cost evaluation of data transmission and storage. It is defined by the ratio (μ) between the size of each share (λ_1) and the size of the original image (λ_2) as shown in Eq. (43).

$$\mu = \frac{\lambda_1}{\lambda_2} \tag{43}$$

$\mu > 1$ means that each pixel in the original image will be expanded during secure sharing. When $\mu = 1$, each image share has the same size with the original image. When $\mu < 1$, the original image is compressed by the secure sharing algorithm. Hence, each image share is smaller than the original image. Smaller ratio μ means a higher compression, and thus a lower cost of transmitting and storing image shares.

From Eq. (43), the pixel expansion ratio of the proposed SMIE-SIS can be calculated by

$$\begin{aligned} \mu &= 1 - (1/2)^{\lceil log_2(n/(2k-2))+1 \rceil} + \delta \\ &\approx 1 - (1/2)^{\lceil log_2(n/(2k-2))+1 \rceil} \end{aligned} \tag{44}$$

where δ can be approximately calculated by

$$\delta = \frac{p}{\lambda_2} \tag{45}$$

For the i^{th} share, p is the information to store the i^{th} row of secret matrix $S^{(k,n)}$, including the size of D and B^i as defined in Eq. (23). Because $p = 2 + \lceil \frac{\mathcal{N}_s}{8} \rceil$ pixels in total are needed to store D and B^i, p is quit small compared with the size of image share F^i (see Eq. (23)). For example, for an original image with size of 256×256, when $k = 4$, $n = 6$, p equals to $2 + \lceil \frac{20}{8} \rceil = 5$, and δ is as small as $5/(256 \times 256) = 7.6e-5$. Thus, it is reasonable to approximately calculate the pixel expansion ratio μ without δ.

TABLE II
PIXEL EXPANSION COMPARISON OF DIFFERENT METHODS

k	**Yang's** [38]	**Wu's** [39]	**Hou's** [14]	**Chao's** [40]	**SMIE-SIS**
2	n	1	1	$2 - 3/n$	$1 - (1/2)^{\lceil log_2(n/2+1) \rceil}$
3	$2n-2$	1	1	$2 - 5/n$	$1 - (1/2)^{\lceil log_2(n/4+1) \rceil}$
4	n^2-2n	1	1	$2 - 7/n$	$1 - (1/2)^{\lceil log_2(n/6+1) \rceil}$

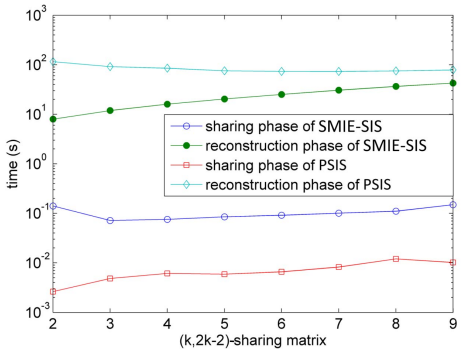


Fig. 10. Computation cost comparison in the sharing and reconstruction phases.

Table II compares the pixel expansion of SMIE-SIS with those of four existing VC methods including the Yang's algorithm [38], Hou's algorithm [14], Chao's algorithm [40] and Wu's algorithm [39]. As we can observe, the pixel expansion of the Hou's [14] and Wu's [39] algorithms is equal to one. The Yang's algorithm [38] has more than one pixel expansion while the Chao's algorithm [40] has the pixel expansion in range of (0, 2). However, SMIE-SIS has pixel expansion in range of (0.5, 1). These demonstrate that SMIE-SIS can reduce the data transmission and storage cost, and outperforms these existing methods.

C. Computation Cost

Although CV methods have almost no computation cost in the reconstruction phase, they have a high pixel expansion ratio and quite low-quality reconstruction. PSIS needs a significantly large computation cost in the reconstruction phase due to the Lagrange interpolation.

Fig. 10 shows a computation cost comparison between the PSIS method in [25] and our SMIE-SIS using MATLAB R2013a in a computer with the Windows 8.1 operating system, Intel(R) Core(TM) i5-4460 CPU@3.20GHz and 12 GB RAM. As can be seen, with the increase of k and n, the computational cost of SMIE-SIS becomes larger. SMIE-SIS requires a slightly larger computation cost than the PSIS method in the sharing phase. However, SMIE-SIS significantly reduces the computation cost in the reconstruction phase while achieving pixel compression and lossless reconstruction, especially when the value of k is small. This is

TABLE III
PERFORMANCE COMPARISONS

	PSIS	VC	SMIE-SIS
Expansion ratio	$\geqslant 1$	$\geqslant 1$	(0.5, 1)
Data loss	small/no	large	No
Reconstruction cost	Large	No	Small
Original image	all types	mainly binary	all types

extremely important for users who have devices with limited capability of data processing. Thus, SMIE-SIS is more suitable for applications.

D. Performance Comparison

Performance comparisons among SMIE-SIS, VC and PSIS from different aspects are shown in Table III. The VC method outperforms other methods in no reconstruction cost, but it has the limitations of a high expansion ratio, a large data loss and mainly working for binary images. PSIS can be divided into two categories: lossy and lossless, according to data loss in reconstructed images. Except for the high expansion ratio (like $\mu = 1$ for [27] and $\mu > 1$ for [28]) and high computation cost, lossy PSIS suffers from small distortion, while lossless PSIS suffers from limited settings of k and n (like $k = 1, n = 2$ only in [27] and [28]). Hence, the proposed SMIE-SIS has excellent advantages, including low reconstruction cost, small expansion ratio to save storage and transmission costs, no data loss, and suitable for all types of original images.

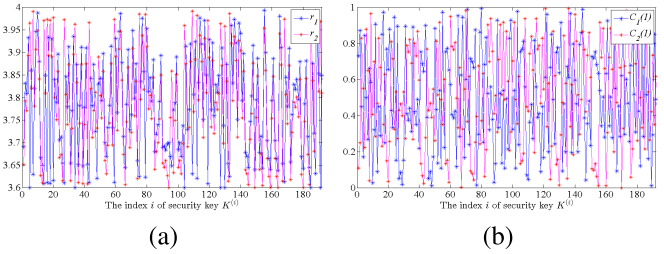
VI. EXPERIMENTAL SECURITY ANALYSIS

Security of existing methods rely only on the secret sharing process, while the proposed SMIE-SIS depends on both the secret sharing and encryption processes. In Section III-C, we theoretically analyze the security of proposed SMIE-SIS. In this section, we experimentally analyze its security and demonstrate that SMIE-SIS has the ability to resist several common attacks such as the brute-force attack, differential attack and fake share attack.

A. Brute-Force Attack

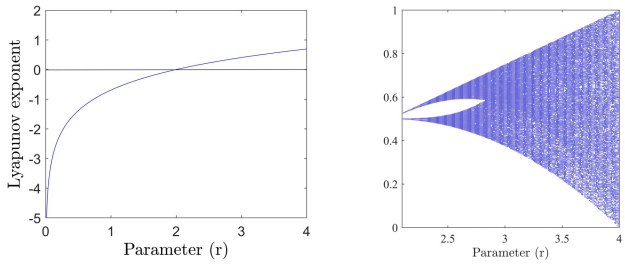
In addition to a large key space analyzed in Section III-C, SMIE-SIS is able to defend the brute-force attack due to its high sensitivity to the security key, ciphertext and plaintext. These high sensitivity mainly results from the utilized image encryption's high key sensitivity to initial conditions, and high chaotic behaviors of the utilized dynamical system. They also ensure the security of the chaotic sequence generated by the Tent map and key.

Since the chaotic system is sensitive to its initial conditions, different security keys yield different chaotic sequences [41]–[43]. In order to test the key sensitivity of SMIE-SIS, we first randomly generate a 192-bit security key denoted by $K^{(0)}$. Based on $K^{(0)}$, we generate other 192 keys denoted by $K^{(1)}$, $K^{(2)}$,..., $K^{(192)}$, where $K^{(i)} (1 \leq i \leq 192)$ is generated from $K^{(0)}$ by flipping the i^{th} bit in $K^{(0)}$. Thus



(a) (b)

Fig. 11. Distributions of initial values (a) r_1, r_2 and (b) $C_1(1)$, $C_2(1)$ that are generated by 193 binary sequences.



(a) (b)

Fig. 12. Lyapunov exponent (a) and bifurcation diagram (b) of the Tent map used in Eq. (19).

$K^{(i)}$ is only one bit different from $K^{(0)}$ in the i^{th} bit. These 193 keys are then utilized to generate 193 sets of initial conditions $(r_1, C_1(1), r_2, C_2(1))$ using Eqs. (17) and (18). The results are plotted in Fig. 11. We can observe that, even with a tiny difference in the security keys, the values of $r_1, C_1(1), r_2$ and $C_2(1)$ change dramatically within their corresponding data ranges. Thus, it is extremely difficult for the attackers to obtain the chaotic sequences by analyzing the security keys.

Different from other chaotic systems suffering from the inadequacy problem [44], Tent map shows a good chaotic map across a large parameter range [45], which can be demonstrated by Lyapunov exponent and bifurcation diagram. Lyapunov exponent (LE) is a measure to describe chaotic behaviors of the dynamical system [46]. Here, we use LE to measure chaotic behaviors of the Tent map in Eq. (19). From the LE results plotted in Fig. 12, we can observe that, when the initial parameter is set to the range of $[3.6, 4]$, the LE results are larger than 0. This means the Tent map in Eq. (19) has good chaotic behaviors in this range. This also can be demonstrated by its bifurcation diagram, which shows uniform distribution of generated chaotic values. Considering our settings for r_1 and r_2 in the range of $[3.6, 4]$, the chaotic sequence shows a good randomness property.

A simulation test is conducted to visually show the process and result of a brute-force attack. Unauthorized users may intend to break a secret sharing system using the brute-force attack to guess the secret share for illegal reconstruction. Fig. 8 shows the results of the brute-force attack when applying a $(4, 6)$-sharing matrix to a grayscale image with size of 256×256. A fake share in Fig. 8(e) is generated by changing only one pixel value of Share 4 in location (100,100). We can observe that, when combining this fake share with three correct secret shares in Fig. 8(a)-(c), the reconstructed

TABLE IV
NPCR, UACI, SRCC AND KRCC RESULTS OF DIFFERENT METHODS WITH (4, 6)-SHARING SCHEME

		Yang's [38]	**Wu's** [39]	**Hou's** [14]	**chao's** [40]	**SMIE-SIS**
NPCR	Share 1,2,3,4,5,6	0,0,0,0,0,0	0,0,0,0,0,0	0,0,0,0,0,0	0,0,0,0,0,0	0.9887,0.9891,0.9890,0.9889,0.9885,0.9888
UACI	Share 1,2,3,4,5,6	0,0,0,0,0,0	0,0,0,0,0,0	0,0,0,0,0,0	0,0,0,0,0,0	0.3318,0.3309,0.3320,0.3300,0.3330,0.3309
SRCC	Share 1,2,3,4,5,6	1,1,1,1,1,1	1,1,1,1,1,1	1,1,1,1,1,1	1,1,1,1,1,1	0.0215,0.0236,0.0216,0.0299,0.0187,0.0212
KRCC	Share 1,2,3,4,5,6	1,1,1,1,1,1	1,1,1,1,1,1	1,1,1,1,1,1	1,1,1,1,1,1	0.0144,0.0158,0.0145,0.0201,0.0125,0.0142

TABLE V
SHANNON ENTROPY ANALYSIS (AVERAGE)

Original image	Share image	Reconstructed image ($< k$ shares)
Fig 5.(a): 0.9995	Fig 5.(b)-(e): 0.9999	Fig 5.(f): 1.0000
Fig 6.(a): 7.1551	Fig 6.(b)-(g): 7.9885	Fig 6.(h)-(i): 7.9973
Fig 7.(a): 7.6780	Fig 7.(b)-(i): 0.9924	Fig 7.(j)-(l): 7.9990

image is noise-like as shown in Fig. 8(h). If and only if four or more secret shares are correctly received, the receiver is able to successfully reconstruct the original image without any error. The possibility of correctly guessing all pixel values in a share is $2^{-256\times256}$. It is almost impossible, showing that the proposed SMIE-SIS can resist the brute-force attack.

B. Shannon Entropy Analysis

Shannon Entropy is introduced to measure the uncertainty [47]. From its mathematical definition, a larger value of Shannon Entropy means larger uncertainty, and also means more randomness. From the Shannon Entropy values of three previously introduced simulation experiments in Figs. 5-7, no matter what the original Shannon Entropy value is, the average Shannon Entropy values of generated shares are close to their theoretical maximum values (1 for binary images and 8 for grayscale and color images), as listed in Table V. This demonstrates good random distributions of pixel values in each share, and verifies good diffusion and confusion property of the proposed system. Table V also includes the Shannon Entropy values of unsuccessfully reconstructed images with less than k shares. Their Shannon Entropy values (close to their theoretical maximum values) show the random distribution of the pixel values in unsuccessful reconstruction results. Hence, from this statistical perspective, there is no original secret information leakage in the unsuccessful reconstructions.

C. Differential Attack

Differential attack is to build a relationship of the differences in input and output to guess the original input using the knowledge of the output. To resist the differential attack, we have to ensure that a small change in input will result in a significant difference in output. Because the proposed SMIE-SIS has non-duplicate property, even using the same original image and same parameter settings, SMIE-SIS produces completely different shares in each execution of SMIE-SIS.

To evaluate the differences of two outputs of SMIE-SIS in two executions with the same input, we use the Unified Average Changing Intensity (UACI) and Number of Pixel Change Rate (NPCR) as defined in Eqs. (46)-(47), where E_1 and E_2 are two output images with size of $M \times N$.

$$UACI = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left(\frac{|E_1(m,n) - E_2(m,n)|}{255}\right) \times 100\% \tag{46}$$

$$NPCR = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\mathcal{B}(m,n)}{MN} \times 100\% \tag{47}$$

where

$$\mathcal{B}(m,n) = \begin{cases} 1 & \text{For } E_1(m,n) \neq E_2(m,n) \\ 0 & \text{Otherwise} \end{cases}$$

A higher value of UACI or NPCR means a huger difference between two output images. A zero value of UACI and NPCR means two measured image are the same.

To investigate the correlation among image shares, we use the Spearman's Rank Correlation Coefficient (SRCC) and Kendall Rank Correlation Coefficient (KRCC) to measure the monotonicity of image shares generated by two executions of SMIE-SIS. The definitions are provided in Eqs. (48) and (49), where $\mathbf{N}$ is the totally number of pixels in each share.

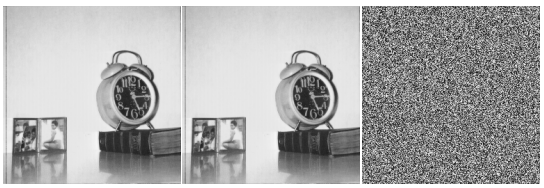
$$\text{SRCC} = 1 - \frac{6\sum_{i=1}^{\mathbf{N}} d_i^2}{\mathbf{N}(\mathbf{N}^2 - 1)} \tag{48}$$

where d_i is the difference between ranks of the i^{th} image share in subjective and objective evaluations.

$$\text{KRCC} = \frac{\mathbf{N}_c - \mathbf{N}_d}{\frac{1}{2}\mathbf{N}(\mathbf{N} - 1)} \tag{49}$$

where $\mathbf{N}_c$ and $\mathbf{N}_d$ are the numbers of concordant and discordant pairs in two shares, respectively. The values of SRCC and KRCC are within the range of $[-1, 1]$. The larger absolute value of SRCC (or KRCC) means a higher correlation between two images.

Table IV shows the UACI, NPCR, SRCC and KRCC results of each share generated by the proposed SMIE-SIS and existing methods. As can be observed, existing methods obtain "0" values in the UACI and NPCR measures and "1" values in the SRCC and KRCC measures. This means that existing methods are vulnerable for differential attacks. For the proposed SMIE-SIS, the values of UACI and NPCR are extremely close to their theoretical values, 33.464% and



(a) (b) (c)

Fig. 13. Fake share attack ($k = 7$). The (a) original image; (b) reconstructed image with 7 true shares; and (c) reconstructed image using 7 true shares and one fake share.

99.609%, which were proved in [48]. And, the values of SRCC and KRCC are close to 0. These demonstrate that SMIE-SIS generates completely different secret shares in two different executions of SMIE-SIS with the same input image and parameters. These ensure a high security level of the proposed SMIE-SIS to withstand different attacks.

D. Fake Share Attack

In real applications, attackers intend to use a fake share to detect information when there are more than k shares involved in the reconstruction phase. Existing methods fail to resist this attack. The proposed SMIE-SIS has a special verification function to detect which share is a fake one. Fig. 13 shows the reconstruction results using $k = 7$ true shares and one fake share. Once a fake share is involved in reconstruction, even the number of true share is larger than k, the proposed SMIE-SIS cannot recover any information of the original image. If and only if all involved shares are true shares and are not less than k, SMIE-SIS can recover the original image. The proposed SMIE-SIS has a high level of security for real applications.

VII. Conclusion

In this paper, we first introduced the mathematical definition, generation algorithm and mathematical analysis of the (k, n)-sharing matrix $S^{(k,n)}$. Combining sharing matrix $S^{(k,n)}$ and chaotic-based encryption, we have further proposed an SMIE-SIS for lossless verifiable (k, n)-secret image sharing. It utilizes a (k, n)-sharing matrix to perform the sharing coding and chaotic-based encryption to obtain n noise-like secret shares. Mathematical analysis has shown that $S^{(k,n)}$ is suitable for image sharing and the combination of chaotic-based encryption and sharing matrix is able to provide high security level. Simulation results have shown that the proposed SMIE-SIS is robust to protect different types of images, including binary, grayscale and color images. The performance analysis demonstrated that SMIE-SIS outperforms several existing visual cryptography and PSIS methods in: 1) lossless reconstruction of the original images, 2) a low pixel expansion ratio to reduce the storage and transmission costs, 3) a low computation cost. The security analysis including theory analysis and experimental demonstration show SMIE-SIS has a high level of security to withstand the brute-force attack, differential attacks, and a verification function to detect the fake shares.

References

[1] S. J. Shyu, "Visual cryptograms of random grids for general access structures," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 3, pp. 414–424, Mar. 2013.

[2] W. Luh and D. Kundur, "Distributed secret sharing for discrete memoryless networks," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 3, pp. 1–7, Sep. 2008.

[3] C.-W. Lee and W.-H. Tsai, "A secret-sharing-based method for authentication of grayscale document images via the use of the PNG image with a data repair capability," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 207–218, Jan. 2012.

[4] L. Harn, "Group authentication," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1893–1898, Sep. 2013.

[5] C.-K. Chu, S. S. M. Chow, W.-G. Tzeng, J. Zhou, and R. H. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 468–477, Feb. 2014.

[6] I. Kang, G. Arce, and H.-K. Lee, "Color extended visual cryptography using error diffusion," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 132–145, Jan. 2011.

[7] Z. Zhou, G. R. Arce, and G. D. Crescenzo, "Halftone visual cryptography," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2441–2453, Aug. 2006.

[8] K.-H. Lee and P.-L. Chiu, "Image size invariant visual cryptography for general access structures subject to display quality constraints," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3830–3841, Oct. 2013.

[9] M. Iwamoto, "A weak security notion for visual secret sharing schemes," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 372–382, Apr. 2012.

[10] S. J. Shyu and H.-W. Jiang, "General constructions for threshold multiple-secret visual cryptographic schemes," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 5, pp. 733–743, May 2013.

[11] M. Naor and A. Shamir, *Visual Cryptography* (Lecture Notes in Computer Science), vol. 950. Berlin, Germany: Springer, 1995, ch. 1, pp. 1–12.

[12] K.-H. Lee and P.-L. Chiu, "Sharing visual secrets in single image random dot stereograms," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4336–4347, Oct. 2014.

[13] H. Wang and D. S. Wong, "On secret reconstruction in secret sharing schemes," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 473–480, Jan. 2008.

[14] Y.-C. Hou and Z.-Y. Quan, "Progressive visual cryptography with unexpanded shares," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1760–1764, Nov. 2011.

[15] K. Kurosawa, "General error decodable secret sharing scheme and its application," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 6304–6309, Sep. 2011.

[16] F. Liu and C. Wu, "Embedded extended visual cryptography schemes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 307–322, Jun. 2011.

[17] Z. Wang, G. R. Arce, and G. D. Crescenzo, "Halftone visual cryptography via error diffusion," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 3, pp. 383–396, Sep. 2009.

[18] K.-H. Lee and P.-L. Chiu, "Digital image sharing by diverse image media," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 1, pp. 88–98, Jan. 2014.

[19] D.-S. Wang, T. Song, L. Dong, and C.-N. Yang, "Optimal contrast grayscale visual cryptography schemes with reversing," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 2059–2072, Dec. 2013.

[20] S.-J. Lin and W.-H. Chung, "A probabilistic model of (t, n) visual cryptography scheme with dynamic group," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 197–207, Feb. 2012.

[21] Y.-C. Hou, S.-C. Wei, and C.-Y. Lin, "Random-grid-based visual cryptography schemes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 5, pp. 733–744, May 2014.

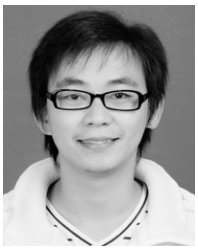
[22] R. De Prisco and A. De Santis, "On the relation of random grid and deterministic visual cryptography," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 4, pp. 653–665, Apr. 2014.

[23] X. Wu and W. Sun, "Generalized random grid and its applications in visual cryptography," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1541–1553, Sep. 2013.

[24] F. Liu, C. Wu, and X. Lin, "Step construction of visual cryptography schemes," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 27–38, Mar. 2010.

[25] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[26] P. Li, P.-J. Ma, X.-H. Su, and C.-N. Yang, "Improvements of a two-in-one image secret sharing scheme based on gray mixing model," *J. Vis. Commun. Image Represent.*, vol. 23, no. 3, pp. 441–453, 2012.
[27] C.-C. Chen and C.-C. Chang, "Secret image sharing using quadratic residues," in *Proc. 3rd Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, vol. 1. Nov. 2007, pp. 515–518.
[28] G. Ulutas, V. V. Nabiyev, and M. Ulutas, "Polynomial approach in a secret image sharing using quadratic residue," in *Proc. 24th Int. Symp. Comput. Inf. Sci.*, Sep. 2009, pp. 586–591.
[29] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.
[30] A. Akhavan, A. Samsudin, and A. Akhshani, "Cryptanalysis of 'an improvement over an image encryption method based on total shuffling,'" *Opt. Commun.*, vol. 350, pp. 77–82, Sep. 2015.
[31] C. Li, "Cracking a hierarchical chaotic image encryption algorithm based on permutation," *Signal Process.*, vol. 118, pp. 203–210, Jan. 2016.
[32] S. Li and X. Zheng, "Cryptanalysis of a chaotic image encryption method," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2. May 2002, pp. II-708–II-711.
[33] E. Y. Xie, C. Li, S. Yu, and J. Lü, "On the cryptanalysis of fridrich's chaotic image encryption scheme," *Signal Process.*, vol. 132, pp. 150–154, Mar. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168416302651
[34] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. London, U.K.: Chapman & Hall, 2007.
[35] Z. Hua, S. Yi, Y. Zhou, C. Li, and Y. Wu, "Designing hyperchaotic cat maps with any desired number of positive Lyapunov exponents," *IEEE Trans. Cybern.* [Online]. Available: https://doi.org/10.1109/TCYB.2016.2642166
[36] S. Yi and Y. Zhou, "Binary-block embedding for reversible data hiding in encrypted images," *Signal Process.*, vol. 133, pp. 40–51, Apr. 2017.
[37] Y. Zhou, Z. Hua, C. M. Pun, and C. L. P. Chen, "Cascade chaotic system with applications," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2001–2012, Sep. 2015.
[38] C.-N. Yang and D.-S. Wang, "Property analysis of XOR-based visual cryptography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 189–197, Feb. 2014.
[39] X. Wu and W. Sun, "Extended capabilities for XOR-based visual cryptography," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1592–1605, Oct. 2014.
[40] K. Chao and J. Lin, "Secret image sharing: A Boolean-operations-based approach combining benefits of polynomial-based and fast approaches," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 2, pp. 263–285, 2009.
[41] Z. Hua and Y. Zhou, "Dynamic parameter-control chaotic system," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3330–3341, Dec. 2016.
[42] L. Bao and Y. Zhou, "Image encryption: Generating visually meaningful encrypted images," *Inf. Sci.*, vol. 324, pp. 197–207, Dec. 2015.
[43] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Inf. Sci.*, vol. 297, pp. 80–94, Mar. 2015.
[44] D. Arroyo, G. Alvarez, and V. Fernandez. (May 2008). "On the inadequacy of the logistic map for cryptographic applications." [Online]. Available: https://arxiv.org/abs/0805.4355
[45] Y. Zhou, L. Bao, and C. L. P. Chen, "Image encryption using a new parametric switching chaotic system," *Signal Process.*, vol. 93, no. 11, pp. 3039–3052, 2013.
[46] J. M. Amigo, L. Kocarev, and J. Szczepanski, "Discrete Lyapunov exponent and resistance to differential cryptanalysis," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 10, pp. 882–886, Oct. 2007.
[47] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, pp. 172–182, Apr. 2014.
[48] C. Fu, J.-J. Chen, H. Zou, W.-H. Meng, Y.-F. Zhan, and Y.-W. Yu, "A chaos-based digital image encryption scheme with an improved diffusion strategy," *Opt. Exp.*, vol. 20, no. 3, pp. 2363–2378, 2012.



Long Bao (S'11) received the B.S. and M.S. degrees in electrical engineering from Hunan University, Changsha, China, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with Tufts University, Medford, MA, USA. He was a Research Assistant with the Department of Computer and Information Science at University of Macau from 2012 to 2014. His research interests are machine learning, image quality assessment, image denoising, image coloring, and multimedia security. He is a Student Member of the IEEE-Eta Kappa Nu.

Shuang Yi received the B.S. degree in software engineering from Chongqing University, Chongqing, China, in 2011. She is currently pursuing the Ph.D. degree with the Department of Computer and Information Science, University of Macau. Her research interests are multimedia security and signal/image processing.

Yicong Zhou (M'07–SM'14) received the B.S. degree from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees from Tufts University, MA, USA, all in electrical engineering.

He is currently an Associate Professor and the Director of the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau, Macau, China. His research interests include chaotic systems, multimedia security, image processing and understanding, and machine learning.

Dr. Zhou was a recipient of the Third Price of the Macau Natural Science Award in 2014. He serves as a Leading Co-Chair of the Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society. He is an Associate Editor of *Neurocomputing* and the *Journal of Visual Communication and Image Representation*.