Juan *et al. J Surveill Secur Saf* 2022;3:41-54
DOI: 10.20517/jsss.2021.29

Journal of Surveillance, Security and Safety

Original Article

Open Access

Check for updates

Advanced fault-tolerant visual multi-secret sharing scheme

Justie Su-Tzu Juan, Jen-Yu Lin[1], Bo-Yuan Huang, Yu-Chun Chung

Department of Computer Science and Information Engineering, National Chi Nan University, Nantou 545, Taiwan.

Correspondence to: Prof. Justie Su-Tzu Juan, Department of Computer Science and Information Engineering, National Chi Nan University, No 1 University Road, Puli, Nantou 545, Taiwan. E-mail: jsjuan@ncnu.edu.tw

How to cite this article: Juan JST, Lin JY, Huang BY, Chung YC. Advanced fault-tolerant visual multi-secret sharing scheme. *J Surveill Secur Saf* 2022;3:41-54. http://dx.doi.org/10.20517/jsss.2021.29

Received: 23 Dec 2021 **First Decision:** 25 Feb 2022 **Revised:** 9 Mar 2022 **Accepted:** 24 Apr 2022 **Published:** 20 May 2022

Academic Editors: Guomin Yang, Jian Shen **Copy Editor:** Peng-Juan Wen **Production Editor:** Peng-Juan Wen

Abstract

Aim: In visual cryptography, a secret image is encrypted into two meaningless random images called shares. These two shares can be stacked to recover the secret image without any calculations. However, because of the alignment problem in the decryption phase, risk of poor quality of the restored image exists. Encrypting multiple secrets on two images simultaneously can improve execution efficiency.

Methods: Let 7×7 pixels be a unit; this paper designs a codebook for any unit in the secret images by using a random grid. Besides, this paper shows a general shifting approach that can embed $N(\geq 2)$ secret images simultaneously with adjustable distortion.

Results: This paper provides a visual multi-secret sharing scheme without pixel expansion; the proposed scheme can encrypt more than two secret images into two shares simultaneously. During decoding, aligning the shares precisely is not necessary.

Conclusion: Theoretical analysis and simulation results indicate the effectiveness and practicality of the proposed scheme.

Keywords: Multi-secrets, pixel expansion, random grid, secret sharing scheme, fault-tolerant, visual cryptography

1. INTRODUCTION

A secret-sharing scheme (SSS) is ideal for storing information and is highly essential for ensuring information security. The primary idea behind an SSS is that a secret is distributed among a group of participants, among whom only qualified participants can reconstruct the secret. A visual SSS (VSSS) is also crucial for information security, and its most attractive feature is that the confidential content in a secret image can be deciphered by the human visual system. Compared with an SSS, a VSSS requires no computation in the decoding phase; therefore, a VSSS is widely applicable in many fields. In 1994 , Noar and Shamir[1] first developed the concept of visual cryptography (VC), and they demonstrated a (k, n)-threshold VSSS, where a secret image S is divided into $n(\geq 2)$ image shares; each share does not reveal information about the original image. The secret image can be seen only when any $k(\leq n)$ shares are stacked together.

The performance of the (k, n)-threshold VSSS[1] is limited for two reasons. The first reason is pixel expansion; specifically, the size of shares is larger than the original secret images. The second reason is the necessity of a codebook, which would require additional storage space. Kafri and Keren[2] proposed a random grid (RG)-based VSSS (RG-based VSSS) that uses the random variable mechanism and encodes a binary image into two RGs. The RG-based VSSS is not marred by the above two drawbacks affecting the (k, n)-threshold VSSS; therefore, the RG-based VSSS is the superior approach. Some RG-based schemes have been proposed[3,4]. Thus far, RG-based VSSS is a well-known SSS that provides this novel stacking-to-see property and prevents pixel scaling while requiring no codebook. Furthermore, Yang[5] introduced a probabilistic model to solve the problem of pixel expansion, called probabilistic visual cryptography scheme (PVCS), in 2004. In a PVCS, the reconstruction of the image, however, is probabilistic, meaning that a secret pixel will be correctly reconstructed only with a certain probability. Since then, several studies have considered PVCS[6–8]. However, from the quality of reconstructed image, pixel expansion, recognized region size, and image types to be considered for evaluating PVCSs and RGs, Yang *et al.*[9] pointed out that RG and PVCS have no difference other than the terminology.

Encoding more than one image simultaneously is efficient with respect to time and space costs. Several studies have proposed and discussed schemes for encoding multiple images simultaneously[10–27]. The scheme proposed by Shyu[10] in 2009 is an extension of that proposed by Shyu[3] in 2007, which can encode two images simultaneously. In 2008, Chen *et al.*[11] proposed an RG-based visual multi-SSS (VMSSS) that entails using a rotating RG to encode two images into two shares. Furthermore, Chen *et al.*[12] presented another VMSSS that can embed four secret images into two shares by using four specific angles, namely 0°, 90°, 180°, and 270°. Liu *et al.*[13] adopted the idea presented by Chen *et al.*[12] and proposed a scheme that can encode three images into two meaningful shares. However, in those approaches, a restraint is imposed on the shape of the input images[11–13]; that is, only square images can be input. To ensure more flexibility regarding the size, Chang *et al.*[14,15] proposed RG-based VMSSSs that can encrypt rectangular images. In these schemes, the confidential content can be reconstructed when users stack two shares with a specific offset and roll them into a cylinder. In 2014, Salehi and Balafar[16] proposed several VMSSSs that involve using a cylindrical RG. They provided two recovery operations, namely XOR and OR (the logical XOR, OR operations, or the Boolean XOR, OR operation); compared with OR, XOR produces images with superior quality after restoration[16]. The OR operation does not require any computation during the restoration of secret images, but the XOR operation does. Furthermore, in these schemes, the distortions are fixed according to the number of input secret images. In 2015, Tsao *et al.*[17] proposed an advanced VMSSS for a general access structure. In this scheme, the extraction of secret data can be prevented by stacking specific shares; however, any two shares cannot restore two different secrets. In 2016, Siva Reddy and Prasad[18] proposed an efficient share generation scheme that ensures the recovery of multiple secrets without any losses. However, this scheme is marred by two disadvantages. One of the disadvantages is that it requires the number of shares to be three times the number of secret images. The other disadvantage is that it applies the XOR recovery operation. Some studies have also considered using XOR operation to design their schemes[19,20].

Table 1. Stacking results of two pixels

p_1	p_2	$p_1 \otimes p_2$
0	0	0
0	1	1
1	0	1
1	1	1

Table 2. Transmittance in KK1

S	Probability	G_1	G_2	$G_1 \otimes G_2$	$T(G_1 \otimes G_2)$
□	1/2	□	□	□	1/2
	1/2	■	■	■	
■	1/2	□	■	■	0
	1/2	■	□	■	

In VC, each share can be printed on a separate transparent sheet, and each input image can be decrypted by overlapping these shares. However, if the scheme lacks fault tolerance, then the reconstructed image may have severely poor visual quality in the event of misalignment. In 2002, Nakajima and Yamaguchi[21] proposed an enhanced VSS scheme that allows slight misalignment by reserving some space for fault tolerance. On the basis of the idea presented by Nakajima and Yamaguchi[21], Juan *et al.* proposed two schemes with no pixel expansion in 2016[22] and 2018[23], respectively. Al-Tamimi and Gaafar[24] designed a scheme that entails using four-pixel blocks to encrypt a secret. Although no pixel expansion occurred, they did not discuss a security analysis of their scheme. In 2016, Lin and Juan used the ideas described by Chang and Juan[14] in 2012, and Juan *et al.*[22] in 2016 to propose an advanced scheme[25] that can encode two images into two shares, and the images can be reconstructed by stacking two imperfectly aligned shares. In the present paper, we combine the ideas presented by Chang *et al.*[15] in 2018, and Juan and Chen[23] in 2018 and propose a fault-tolerant multi-SSS that encodes more than two secret images into two shares simultaneously without pixel expansion.

The rest of this paper is organized as follows. Related work and the proposed scheme are presented in Sections 2 and 3, respectively. The experimental results are presented in Section 4. Analysis and comparisons are presented in Section 5.

2. RELATED WORK

In 1987, Kafri and Keren[2] first introduced the idea of RG that can be used to classify any pixel in an image as black (opaque) or white (transparent). An imager pixel S is denoted as $S(i, j)$, where (i, j) is the pixel position in the image. We define a black pixel as $S(i, j) = 1$ and a white pixel as $S(i, j) = 0$. The probability of generating a white or black pixel in an RG is the same because of the inherent random mechanism of an RG; that is, the probability is equal to $1/2$.

The *transmittance* of an image G, denoted as $T(G)$, is the ratio of the number of white pixels to all the pixels in G. Therefore, the transmittance of each RG is $1/2$. A secret image S can be encrypted into two shares G_1 and G_2, and the confidential content can be entirely restored according to the stacking rules, as presented in Table 1, where p_i represents a pixel in G_i for $i = 1, 2$ and $p_1 \otimes p_2$ represents the result of superimposing two pixels. On the basis of these definitions, three different RG algorithms were proposed by Kafri and Keren[2], one of which (called KK1) is presented as follows. Table 2 presents the transmittance of $G_1 \otimes G_2$ in KK1.

2.1. KK1 algorithm (proposed in[2])

Input: secret image with size $m \times n$ pixels.
Output: two cipher-shares G_1 and G_2.

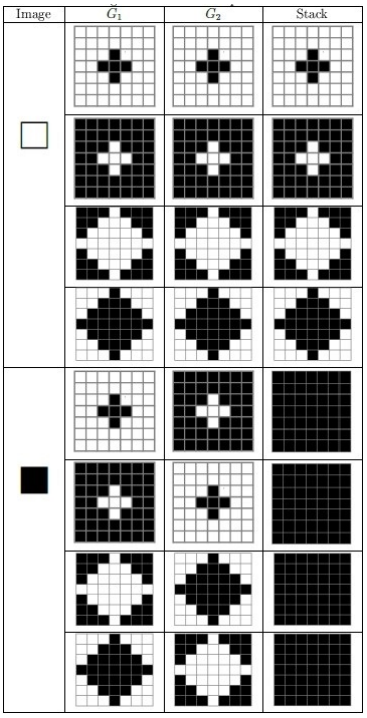


Figure 1. Designed patterns for $n = 7$. This figure is quoted with permission from Juan and Chen[23].

Table 3. Transmittance for $n = 7$

C	Stack	Shift 1 pixel	Shift 2 pixels	Diagonal shift 1 pixel
$T_W(C)$	1/2	73/196	60/196	69/196
$T_B(C)$	0	25/196	38/196	29/196

Generate an $m \times n$ RG G_1
  for (int $i = 0; i < m; i++$)
    for (int $j = 0; j < n; j++$)
      if ($S[i][j] == 0$)
        $G_2[i][j] = G_1[i][j]$
      else
        $G_2[i][j] = 1 - G_1[i][j]$
Return G_1 and G_2

Using the basic model of VC first introduced by Noar and Shamir[1], Nakajima and Yamaguchi[21] proposed a fault-tolerant scheme with pixel expansion in which diamond patterns are designed for encoding; this allows slight deviations during stacking while enabling the classification of the original color of the secret image. In 2016, Juan *et al.*[22] proposed a fault-tolerant scheme without pixel expansion. This scheme considers $u \times u$ pixels as a unit; therefore, the images are divided into several units ($u = 3, 4, 5$, or 6). During the generation of the first share, a unit is selected randomly from the design patterns in the codebook. The unit in the second share is then selected from a specific pattern according to the number of black and white pixels of the secret image. Using the patterns designed by the authors will make the algorithm fault-tolerant. Next, Juan and Chen[23] proposed an improved algorithm to increase the value of u to 7 in 2018. They designed the 7×7 patterns shown in Figure 1, which made their scheme more fault-tolerant. The transmittance of stacking results in terms of the black and white pixels is presented in Table 3. For details of these analyses, please refer to Juan and Chen's scheme[23].

With respect to RG, Lin and Juan[25] proposed a visual two-secret sharing scheme with no pixel expansion; the scheme has fault-tolerant mechanisms and combines the advantages of Juan *et al.*'s scheme[22] and Chang and Juan's scheme[14]. The encryption phase is divided into three parts, namely scale down, encryption, and tolerance. In the scale down part, an input image of size $m \times n$ pixels is partitioned into $m/6 \times n/6$ units, where the size of each unit is 6×6 pixels ($u = 6$). In the other two parts, the design patterns in Juan *et al.*'s scheme[22] are used for fault tolerance, and a concept similar to that in Chang and Juan's scheme[14] is then used for encoding. In our scheme, we set $u = 7$ and adopt the well-established design patterns in Juan and Chen's scheme[23]to ensure misalignment tolerance. Since the transmittance of stacking results in Juan and Chen's scheme[23] is better than those in Juan *et al.*'s scheme[22], we believe that using the patterns in Juan and Chen's scheme[23] to design our algorithm will yield better results than Lin and Juan's[25].

For multiple-image encryption, Chen *et al.*[12] proposed a scheme that encodes four secret images into two square shares. Through a rotating mechanism, the four confidential secret images can be restored by stacking one share on the other share after rotating by $0°, 90°, 180°,$ or $270°$. Some disadvantages of the mentioned scheme[12] are as follows: the distortions are fixed, the input secret image should be a square, and the number of input secret images should be $<$ or $=$ 4. To address these disadvantages, Chang *et al.*[14,15] proposed some advanced schemes.

We briefly introduce Chang and Juan's scheme[14] that encodes two images into two shares. The main idea of the algorithm[14] is as follows. Two images are divided into p partitions, where p is the positive factor of the width of the image (which can be determined by users). Every $2p$ pixels (in the same position of each subset of the partition) can be considered as a set and can be encrypted together in the next step. Next, an unencrypted pixel from one image is randomly selected to be the basis of the encoding process; subsequently, the corresponding $2p - 2$ pixels can be encrypted (one of the pixels in this set is not encrypted). In addition, two images serve as the basis alternately; therefore, the unencrypted pixel is evenly distributed into two shares in the encoding process. Finally, the steps are repeated until two cipher-shares are generated. Therefore, all the secret data are evenly encrypted into two shares.

In the VMSSS[15], the input images $S_0, S_1, \cdots, S_{N-1}$ of size $m \times n$ pixels are encoded into two shares G_1 and G_2, and integer p is selected according to user requirements (must be a positive factor of the width of the image m). Two consecutive images are randomly selected, and each set ($2p$ pixels) is processed using Chang and Juan's scheme[14]. Thus, N input images are evenly selected, ensuring that each share contains the confidential data of every secret image. Figure 2 shows the main idea of how their scheme forms $2p$ pixels in G_1 and G_2 when one pixel (i, j) and two consecutive secret images S_A and S_{A+1} are randomly selected (here, we let $A = 1$ and $p = 4$).

The study by Chang *et al.*[15] is an extension of Chang and Juan's scheme[14]; the schemes proposed by both studies involve the same recovery operation. During the restoration of secret images S_i for $i = 0, 1, \cdots, N - 1$, G_1 is horizontally shifted by im/p pixels and then stacked with G_2. These two schemes involve shifting instead of rotating the RG to repair the shape problem of the input images and adopting p for adjustable distortion. The *distortion* (DT) is defined as the ratio of the number of pixels that are not used in the encryption phase to the total number of pixels in all secret images. The distortion of the schemes in Chang and Juan's scheme[14] and Chang *et al.*'s scheme[15] can be calculated as $1/(2p)$ and $([N - 2]p + 1)/(Np)$, respectively.

On the basis of the ideas presented in Chang *et al.*'s scheme[15], and Juan and Chen's scheme[23], we propose a more advanced scheme than Lin and Juan's scheme[25]. Specifically, our proposed scheme uses design patterns[23] to ensure fault tolerance and encrypts multiple secret images by shifting the RG. The scheme is introduced in detail in the next section. To make the relationship between the above schemes clearer, we list them in Table 4.

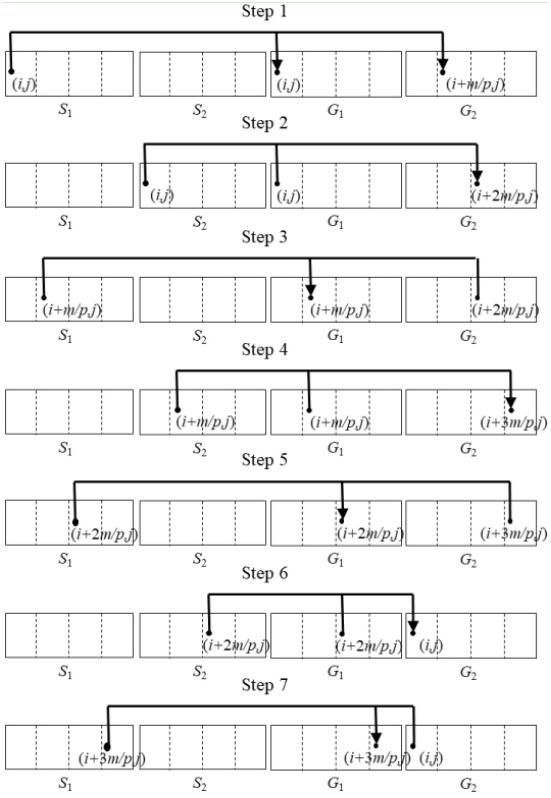


Figure 2. The diagram of the processes in the encryption phase of the VMSSS [15] for $A = 1$ and $p = 4$.

Table 4. The relationship between the proposed scheme and some related schemes

N = the number of secret images	Has no fault-tolerant mechanisms	Has lower fault-tolerant mechanisms ($n = 6$)	Has higher fault-tolerant mechanisms ($n = 7$)
$N = 1$	Kafri and Keren [2] Noar and Shamir [1]	Juan *et al.* [22]	Juan and Chen [23]
$N = 2$	Chang and Juan [14]	Lin and Juan [25]	
$N \geq 2$	Chang *et al.* [15]		This paper

3. OUR SCHEME

In this section, we describe the proposed scheme for encrypting multi-secret images by shifting RGs through a fault-tolerant mechanism. Therefore, the proposed scheme can be denoted as fault-tolerant RG-based VMSSS (*FT-VMSSS*).

In our scheme, there are four functions and one procedure that help us build the main algorithm. The first three functions are defined as follows. They are used below in the procedure and main algorithm.

Definition 1. $f_{RSP}(.) : Y \leftarrow f_{RSP}(X), Y$ is the output of the function $f_{RSP}(.)$ with the inputs X, where $f_{RSP}(.)$ is randomly selecting a pixel of X.
Definition 2. $f_{RG}(.) : Y \| Z \leftarrow f_{RG}(X), Y$ and Z are the outputs of the function $f_{RG}(.)$ with the input X, where $f_{RG}(.)$ is KK1 algorithm that inputs a pixel of the secret image, and then outputs two cipher-pixels for two shares.
Definition 3. $f_{\overline{RG}}(.) : Z \leftarrow f_{\overline{RG}}(X, Y), Z$ is the output of the function $f_{\overline{RG}}(.)$ with the inputs X and Y, where $f_{\overline{RG}}(.)$ is the function according to $f_{RG}(.)$ (as in Definition 2) which inputs a cipher-pixel of one share Y and a pixel of the secret image X, and then outputs the other cipher-pixel.

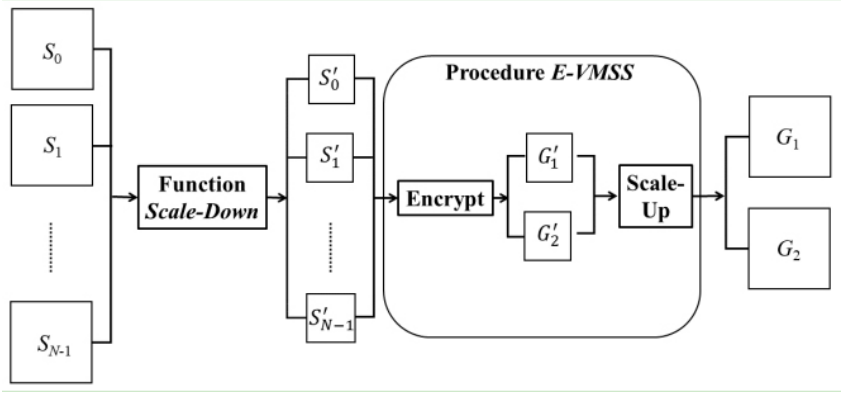


Figure 3. The model of proposed encrypt method.

Table 5. Notations used in the algorithm *FT-VMSS* and procedure *E-VMSS*

N	: Number of secret images.
S_i	: Secret image i, $0 \le i \le N-1$, the size of S_i is $m \times n$ pixels.
p	: A positive integer that must be a positive factor of m. User-determined and related to distortion.
S'_i	: Scaled-down image i, $0 \le i \le N-1$, the size of S'_i is $m/7 \times n/7$ pixels.
A	: A random number among $\{0, 1, 2, \cdots, N-1\}$.
(i, j)	: A random position determined by $f_{RSP}(.)$.
$S(i, j)$	: Pixel value of image S in position (i, j).
%	: Modulus operation.
G'_i	: Reduced share i, $1 \le i \le 2$, the size of G'_i is $m/7 \times n/7$ pixels.
X	: A random pattern X in Figure 1, the size of X is 7×7 pixels.
G_i	: Share i, $1 \le i \le 2$, the size of G_i is $m \times n$ pixels.

Figure 3 shows the model of the proposed encrypt method. The main concept of our approach is as follows: first, we divide each secret image into several blocks with pixels of size 7×7, thus resulting in 49 pixels per block. Second, we label each block as "black" or "white" according to the number of pixels in the block. If > 24 (half of 49) pixels in a block are black, then the block is labeled as black; otherwise, it is labeled as white. These two steps are what the function *Scale-Down* in the algorithm does. Then, similar to Chang *et al*.'s VMSSS[15], we repeat the encryption process until two cipher-shares are generated randomly using the aforementioned step and Figure 1 (from Juan and Chen's scheme[23]). Thus, when the images are restored, they consist of several design patterns, as presented in Figure 1. This process ensures that secret data are decipherable because of the sufficient transmittance difference, as presented in Table 3; for example, the transmittance differences are 73/196 and 25/196 in the case of "shift 1 pixel". This important step is done by Procedure *E-VMSS*.

Therefore, using the *Scale-Down* function, we obtain N resized images $S'_0, S'_1, \cdots, S'_{N-1}$ with pixels of size $m/7 \times n/7$ from the input N secret images $S_0, S_1, \cdots, S_{N-1}$ with pixels of size $m \times n$. We separate the primary procedure *E-VMSS* into two parts. In the first part (Encrypt part), we encrypt the scaled-down images S'_i, S'_{i+1} into encoded contents G'_1 and G'_2 with pixels of size $m/7 \times n/7$ according to the *E-VMSS*. In the second part (Scale-Up part), each pixel in G'_1 and G'_2 is restored to the corresponding pattern in Figure 1 to obtain G_1 and G_2, according to the pattern X randomly selected at the beginning. Then, G_1 and G_2 are obtained, which preserve the size of the input secret images, and the problem of pixel expansion is avoided. In other words, in procedure *E-VMSS*, we first encrypt the reduced secret images into two reduced shares (Encrypt part), and then restore the reduced shares back to the size we want, namely the size of the original secret images (Scale-Up part). Since such steps are performed pixel by pixel, the instructions of these two parts are not consecutive in procedure *E-VMSS*. For each pair of encrypted pixels in S'_i, S'_{i+1}, we first generate a pair of relative pixels in reduced shares G'_1 and G'_2, and then use this pair of pixels to immediately restore 7×7 pixels in shares G_1 and G_2 (with the same size as secret images S_i). Therefore, the algorithm *FT-VMSSS* can be performed using four basic functions and one procedure; the algorithmic codes are well constructed as described in the following subsection. The related notations are listed in Table 5.

3.1. Encryption phase

Function 1. (f_{RSP})
Input: a secret image S with pixels of size $m \times n$
Output: one pixel of the input secret image $S(i, j)$
i = random$(0, m-1)$
j = random$(0, n-1)$
Return (i, j)

Function 2. (f_{RG})
Input: a pixel of secret image $S(i, j)$
Output: a pixel of shares $G_1(i, j)$ and $G_2(i, j)$
$G_1(i, j)$ = random$(0, 1)$
if $(S(i, j) == 0)$
$G_2(i, j) = G_1(i, j)$
else $G_2(i, j) = 1 - G_1(i, j)$
Return $(G_1(i, j), G_2(i, j))$

Function 3. ($f_{\overline{RG}}$)
Input: a pixel of secret image $S(i, j)$, and a pixel of one share $G_1(i, j)$
Output: a pixel of the other share $G_2(i, j)$
if $(S(i, j) == 0)$
$G_2(i, j) = G_1(i, j)$
else $G_2(i, j) = 1 - G_1(i, j)$
Return $G_2(i, j)$

Function 4. (*Scale-Down*)
Input: the original secret image S with pixels of size $m \times n$
Output: the scaled down image S' with pixels of size $m/7 \times n/7$

```
for (int a = 0; a < m/7; a++)
    for (int b = 0; b < n/7; b++)
        count = 0;
        for (int i = 7a; i < 7(a + 1); i++)
            for (int j = 7b; j < 7(b + 1); j++)
                if (S(i, j) == 0)
                    count++
        if (count> 24)
            S'(a, b) = 0
        else
            S'(a, b) = 1
Return S'
```

Procedure *E-VMSS* $(S_A(i, j), S_B(i, j), p, A)$
$G'_1(i, j)||G'_2((i + A * m/p)\%m, j) \leftarrow f_{RG}(S_A(i, j))$
One pattern X with pixels of size 7×7 is randomly selected from the G_1 column in Figure 1.

```
for (int a = 0; a < 7; a++)
    for (int b = 0; b < 7; b++)
        G1(7i + a, 7j + b) ← X(a, b)
```

if $(G'_2((i + A * m/p)\%m, j) == G'_1(i, j))$
$G_2(7(i + A * m/p)\%m + a, 7j + b) \leftarrow X(a, b)$
else
$G_2(7(i + A * m/p)\%m + a, 7j + b) \leftarrow 1 - X(a, b)$
for (int $k = 0; k < p - 1; k++$)
if $(A <> N - 1)$
int $A' = (A + k + 1)\%p;$
$G'_2((i + A' * m/p)\%m, j) \leftarrow f_{\overline{RG}}(S_B((i + k * m/p)\%m, j), G'_1((i + k * m/p)\%m, j))$
$G'_1((i + (k + 1) * m/p)\%m, j) \leftarrow f_{\overline{RG}}(S_A((i + (k + 1) * m/p)\%m, j), G'_2((i + A' * m/p)\%m, j))$
for (int $a = 0; a < 7; a++$)
for (int $b = 0; b < 7; b++$)
if $(G'_2((i + A' * m/p)\%m, j) == G'_1(i, j)))$
$G_2(7((i + A' * m/p)\%m) + a, 7j + b) \leftarrow X(a, b)$
else
$G_2(7((i + A' * m/p)\%m) + a, 7j + b) \leftarrow 1 - X(a, b)$
if $(G'_1((i + (k + 1) * m/p)\%m, j) == G'_1(i, j)$
$G_1(7((i + (k + 1) * m/p)\%m) + a, 7j + b) \leftarrow X(a, b)$
else
$G_1(7((i + (k + 1) * m/p)\%m) + a, 7j + b) \leftarrow 1 - X(a, b)$
else
$G'_2((i - kA * m/p)\%m, j) \leftarrow f_{\overline{RG}}(S_0((i - kA * m/p)\%m, j), G'_1((i - kA * m/p)\%m, j))$
$G'_1((i - (k + 1)A * m/p)\%m, j) \leftarrow f_{\overline{RG}}(S_A((i - (k + 1)A * m/p)\%m, j), G'_2((i - kA * m/p)\%m, j))$
for (int $a = 0; a < 7; a++$)
for (int $b = 0; b < 7; b++$)
if $(G'_2((i - kA * m/p)\%m, j) == G'_1(i, j)))$
$G_2(7((i - kA * m/p)\%m) + a, 7j + b) \leftarrow X(a, b)$
else
$G_2(7((i - kA * m/p)\%m) + a, 7j + b) \leftarrow 1 - X(a, b)$
if $(G'_1((i - (k + 1)A * m/p)\%m, j) == G'_1(i, j)$
$G_1(7((i - (k + 1)A * m/p)\%m) + a, 7j + b) \leftarrow X(a, b)$
else
$G_1(7((i - (k + 1)A * m/p)\%m) + a, 7j + b) \leftarrow 1 - X(a, b)$

Algorithm *FT-VMSS*
Input: Secret images $S_0, S_1, \cdots, S_{N-1}$ and positive integer p (a positive factor of m)
Output: Shares G_1 and G_2
for$(i = 0; i < N; i++)$
S'_i =*Scale-Down*(S_i);
Repeat
Randomly select A from $\{0, 1, 2, \cdots, N - 1\}$;
$(i, j) \leftarrow f_{RSP}(S_A)$;
Procedure E-VMSS $(S'_A(i, j), S'_{(A+1)\%N}(i, j), p, A)$;
Until all the pixels of G_1 and G_2 are generated

3.2. Decryption phase

After collecting the two cipher-grids G_1 and G_2, users can easily restore N secret images. The first secret image can be reconstructed by directly stacking G_1 and G_2 together. The secret image S_i for $i = 1, 2, \cdots, N - 1$ can be restored by superposing G_1 and G_i, where G_i is obtained from G_2 through horizontal shifting by a width

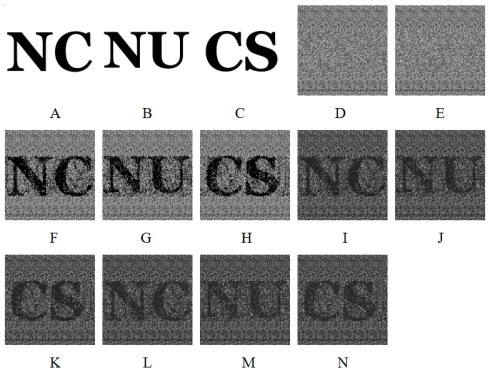


Figure 4. Results of Experiment 1: (A-C) secret images; (D) share G_1; (E) share G_2; (F-H) restored images; (I-K) restored images (one-pixel right shift); and (L-N) restored images (one-pixel diagonal shift).

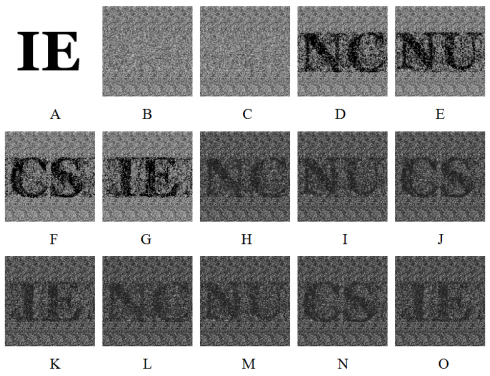


Figure 5. Results of Experiment 2: (A) secret image; (B) share G_1; (C) share G_2; (D-G) restored images; (H-K) restored images (one-pixel right shift); and (L-O) restored images (one-pixel diagonal shift).

of i/p.

4. EXPERIMENTAL RESULTS

This section describes three experiments performed in this study. In the first experiment, we encoded three images into two shares, where the size of the input images was 980×980 pixels, as presented in Figure 4A-C. By setting N to 3 and p to 7 in the FT-VMSSS algorithm, we generated two cipher-shares G_1 and G_2, as presented in Figure 4D and E. The first secret was restored by stacking G_1 and G_2 directly. The second (third) secret was obtained by superimposing G_2 with G_1 that had being horizontally shifted by a width of 1/7(2/7) to the left, as presented in Figure 4F-H. The results of imperfect stacking and one-pixel right shift are presented in Figure 4I-K. For example, when G_2 was stacked with G_1 that had shifted to the right by one pixel, the first restored image could still be obtained, as depicted in Figure 4I. The results for the other stacking cases involving misalignment and one-pixel diagonal shift are presented in Figure 4L-N.

In the second experiment, four images with pixels of size 980×980, as presented in Figure 5A and Figure 4A-C, were used, N was set to 4, and p was set to 7. The two shares G_1 and G_2 were generated after the *FT-VMSS* algorithm was applied, as depicted in Figure 5B and C. Perfect stacking of the reconstructed four secret images is presented in Figure 5D-G. Figure 5H-K presents the restored images for imperfect stacking (with a one-pixel right shift). Restored images for one-pixel diagonal shift are presented in Figure 5L-O.

In the third experiment, images with pixels of size 1960 × 1400 were used, as presented in Figure 6A-D. For

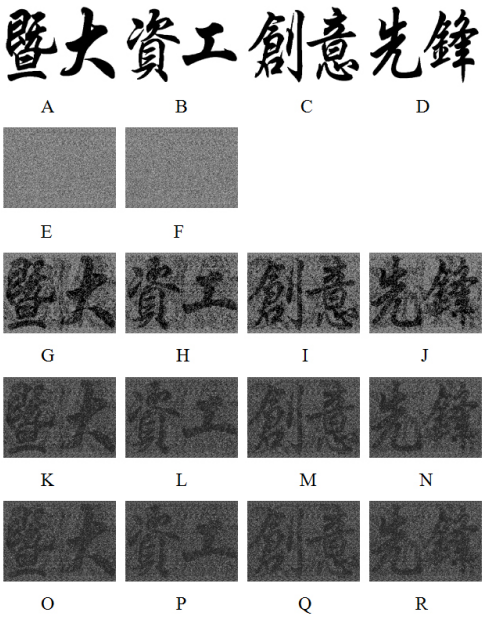


Figure 6. Results of Experiment 3: (A-D) secret images; (E) share G_1; (F) share G_2; (G-J) restored images; (K-N) restored images (one-pixel right shift); and (O-R) restored images (one-pixel diagonal shift).

this simulation, N was set to 4 and p was set to 20. Figure 6E and F presents the two cipher-shares generated. The first (second, third, and fourth) secrets were obtained using the decryption process described in Section 3, as indicated in Figure 6G (Figure 6H-J, respectively). Experimental results for the case of imperfect alignment and one-pixel right shift are displayed in Figure 6K-N. The results for the case of misaligned stacking and one-pixel diagonal shift are presented in Figure 6O-R.

5. ANALYSIS AND CONCLUSIONS

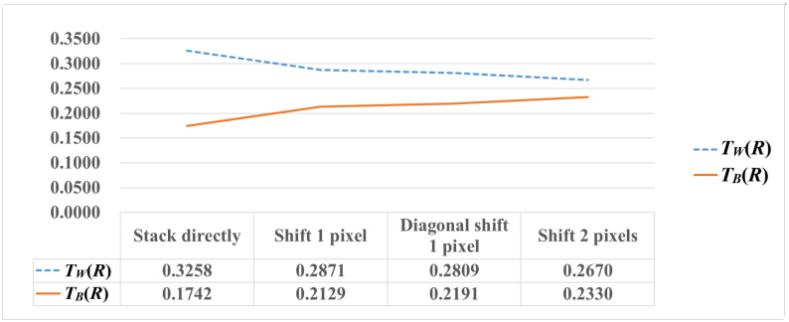
In this paper, we present the design of an *FT-VMSSS* for sharing multiple secret images simultaneously, as described in Section 3. In the proposed scheme, multiple secrets can be embedded into two cipher shares, and the distortion is evenly distributed across the two shares.

Let $T_W(R)$ be the *transmittance* of the area in the restored image R, which corresponds to the white area in the secret image and $T_B(R)$ be the *transmittance* of the area in the restored image R, which corresponds to the black area in the secret images. We then recalculate the problem of transmittance because of the effect of distortion. The transmittance of the successfully encrypted part is not affected. However, the distortion part $[(N-2)p+1)/Np]$ is not encrypted. The probability of the distortion part being black (or white) is 1/2. Therefore, the transmittance of the recovered image R can be derived as follows, where $T_W(C)$ and $T_B(C)$ are described in Table 3 .

$$T_B(R) = \frac{T_B(C) \times (48p-24) + ((49N-48)\,p+24) \times \frac{1}{2} \times (T_B(C) + T_W(C))}{49Np}, \tag{1}$$

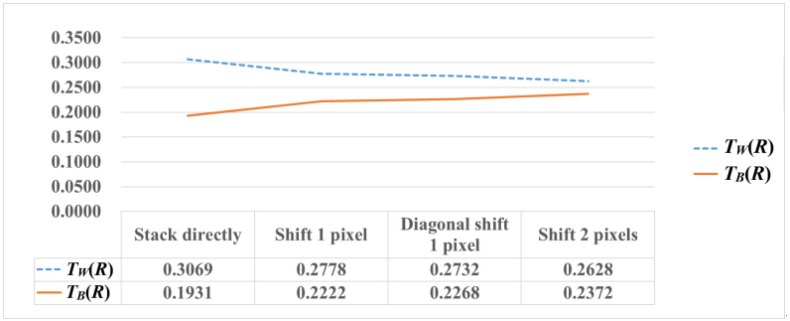
$$T_W(R) = \frac{T_W(C) \times (48p-24) + ((49N-48)\,p+24) \times \frac{1}{2} \times (T_B(C) + T_W(C))}{49Np} \tag{2}$$

Therefore, when $N, p \geq 1, T_W(R) > T_B(R)$ because $T_W(C) > T_B(C)$ in Table 3 (from Juan and Chen's scheme[23]) for any stacked restored image (perfect alignment or imperfect alignment with one-pixel shift, two-pixel shift, or diagonal one-pixel shift). Thus, the white and black pixel areas in each secret image can



	Stack directly	Shift 1 pixel	Diagonal shift 1 pixel	Shift 2 pixels
$T_W(R)$	0.3258	0.2871	0.2809	0.2670
$T_B(R)$	0.1742	0.2129	0.2191	0.2330

Figure 7. Transmittance analysis for $N = 3$ and $p = 7$.



	Stack directly	Shift 1 pixel	Diagonal shift 1 pixel	Shift 2 pixels
$T_W(R)$	0.3069	0.2778	0.2732	0.2628
$T_B(R)$	0.1931	0.2222	0.2268	0.2372

Figure 8. Transmittance analysis for $N = 4$ and $p = 7$.

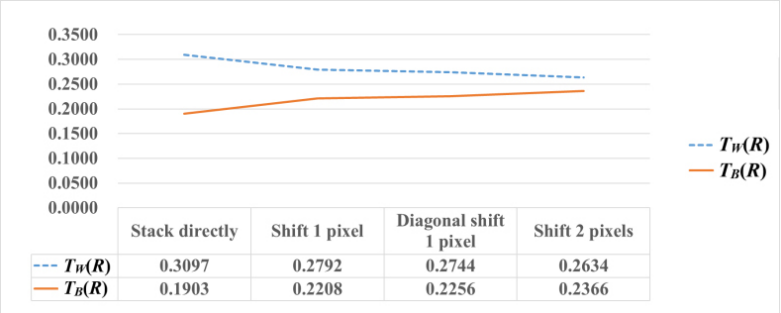
be visually distinguished (or the information of secret images can be observed) from the restored image R. Conversely, $T(G_1) = T(G_2) = 1/2$, since any unit is randomly selected in Figure 1 for any (i, j) in G'_1 to construct the corresponding units of G_1 and G_2. Thus, no information of any S_i for $i = 0, 1, \cdots, N-1$ can be obtained from G_1 or G_2 individually. That means that our algorithm is correct and secure.

Through the encryption process described in Section 3 and Chang *et al.*'s scheme[15], we can calculate the distortion of the restored images R as follows, where N is the number of input secret images and p is the specified positive factor of the width m of the input image.

$$DT(R) = 1 - \frac{25}{49} \times \frac{2p-1}{Np} = \frac{(49N-50)\,p+25}{49Np} \tag{3}$$

According to our experimental results, fault-tolerant performance can permit a shift of only one pixel. Figures 7-9 indicate that, for the transmittance analysis for sharing three or four secret images, $p = 7$ or 20, the difference between white and black transmittance decreased gradually from the case of perfect stacking to that of imperfect stacking (but $T_W(R) > T_B(R)$ in any case). Therefore, to address the fault caused by more pixel shifting, the input secret images must be set in a larger size.

Table 6 presents a comparison of the performance of our scheme and those of schemes proposed in related studies. Among the schemes proposed in related studies, those with a fault-tolerant mechanism can encrypt at most two secret images at a time, and those that can encrypt any multiple secret images lack the mechanism of fault tolerance. Our scheme is the first MVSSS to be fault-tolerant and can really encrypt any number of secret images. Therefore, compared with these schemes, our scheme is more practical.



	Stack directly	Shift 1 pixel	Diagonal shift 1 pixel	Shift 2 pixels
$T_W(R)$	0.3097	0.2792	0.2744	0.2634
$T_B(R)$	0.1903	0.2208	0.2256	0.2366

Figure 9. Transmittance analysis for $N = 4$ and $p = 20$.

Table 6. Comparison of related schemes to the proposed scheme

	Fault tolerance	Without pixel expansion	By random grid	Flexible	The number of secrets
The proposed scheme	Yes	Yes	Yes	Yes	≥ 2
[15]	No	Yes	Yes	Yes	≥ 2
[21]	Yes	No	No	No	1
[22,23]	Yes	Yes	Yes	No	1
[25]	Yes	Yes	Yes	Yes	2
[26-28]	Yes	No	No	No	1
[29]	Yes	Yes	No	No	1
[30]	Yes	Yes	Yes	No	1

DECLARATIONS

Authors' contributions

Conceived the proposed idea, provided the main plan and supervised the entire project: Juan JS-T
Contributed in performing the experiments and analyzed the results: Lin JY
Participated in the experiment and drafting the manuscript: Huang BY
Corrected some deficiencies in the main scheme and assisted in revising the article: Chung YC
All authors discussed the main idea and scientific contribution.

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work was supported by the Ministry of Science and Technology of the Republic of China under Contract (MOST 108-2221-E-260-008- and MOST 110-2221-E-260-003).

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Naor M, Shamir A. Visual cryptography. In: De Santis A, editor. Advances in cryptology-EUROCRYPT'94. Berlin: Springer Berlin Heidelberg; 1995. p. 1-12. DOI
2. Kafri O, Keren E. Encryption of pictures and shapes by random grids. *Opt Lett* 1987;12:377-9. DOI
3. Shyu SJ. Image encryption by random grids. *Pattern Recognition* 2007;40:1014-31. DOI
4. Lin SJ, Lin JC, Fang WP. Visual cryptography (VC) with non-expanded shadow images: hilbert-curve approach. In Proceedings of the IEEE International Conference on Intelligence and Security Informatics, 2008. p. 271-2. DOI
5. Yang CN. New visual secret sharing schemes using probabilistic method. *Pattern Recognition Letters* 2004;25:481-94. DOI
6. Cimato S, De Prisco R, De Santis A. Probabilistic visual cryptography schemes. *The Computer Journal* 2006;49:97-107. DOI
7. Lin SJ, Chung WH. A probabilistic model of (t, n) visual cryptography scheme with dynamic group. IEEE Trans Inform Forensic Secur 2012;7:197-207. DOI
8. Wu X, Yang CN. Probabilistic color visual cryptography schemes for black and white secret images. *Journal of Visual Communication and Image Representation* 2020;70:102793. DOI
9. Yang CN, Wu CC, Wang DS. A discussion on the relationship between probabilistic visual cryptography and random grid. *Information Sciences* 2014;278:141-73. DOI
10. Shyu SJ. Image encryption by multiple random grids. *Pattern Recognition* 2009;42:1582-96. DOI
11. Chen TH, Tsao KH, Wei KC. Multiple-image encryption by rotating random grids. In Proceedings of Eighth International Conference on Intelligent Systems Design and Applications, 2008. p. 252-6. DOI
12. Chen TH, Tsao KH, Lee YS. Yet another multiple-image encryption by rotating random grids. Signal Processing 2012;92:2229-37. DOI
13. Liu CL, Tsai WJ, Chang TY, Peng CC, Wong PS. Meaningful share generation for (2, 2)-multiple visual secret sharing scheme without pixel expansion. *The Computer Journal* 2015;58:1598-606. DOI
14. Chang JJY, Juan JST. Multi-VSS scheme by shifting random grids. In Proceedings of World Academy of Science, Engineering and Technology. *International Journal of Computer and Information Engineering* 2012;6:1-7. DOI
15. Chang JJY, Huang BY, Juan JST. A new visual multi-secrets sharing scheme by random grids. *Cryptography* 2018;2:24. DOI
16. Salehi S, Balafar MA. Visual multi secret sharing by cylindrical random grid. *Journal of Information Security and Applications* 2014;19:245-55. DOI
17. Tsao KH, Shyu SJ, Lin CH, Lee YS, Chen TH. Visual multiple-secret sharing for flexible general access structure by random grids. *Displays* 2015;39:80-92. DOI
18. Siva Reddy L, Prasad MVNK. Extended visual cryptography scheme for multi-secret sharing. In: Nagar A, Mohapatra D, Chaki N, editors. Proceedings of 3rd international conference on advanced computing, networking and informatics. New Delhi: Springer; 2016. p. 249-57. DOI
19. Wu X, Sun W. Random grid-based visual secret sharing with abilities of OR and XOR decryptions. *Journal of Visual Communication and Image Representation* 2013;24:48-62. DOI
20. Nag A, Biswas S, Sarkar D, Sarka PP. Secret image sharing scheme based on a boolean operation. *Cybernetics and Information Technologies* 2014;14:98-113. DOI
21. Nakajima M, Yamaguchi Y. Extended visual cryptography for natural images. *Journal of WSCG* 2002;10:303-10. Available from: https://www.researchgate.net/publication/221546377_Extended_Visual_Cryptography_for_Natural_Images [Last accessed on 27 Apr 2022]
22. Juan JST, Chen YC, Guo S. Fault-tolerant visual secret sharing schemes without pixel expansion. Appl Sci 2016;6:18. DOI
23. Juan JST, Chen YC. Extended fault-tolerant visual secret sharing scheme without pixel expansion. In Proceedings of International Conference on Security and Management (SAM), 2018. Available from: https://www.proquest.com/openview/23eef0355811fe65b1164b61bbbb5b46/1?pq-origsite=gscholar&cbl=1976342 [Last accessed on 27 Apr 2022]
24. Al-Tamimi AGT, Gaafar A. A new simple non-expansion algorithm for (2,2)-visual secret sharing scheme. *International Journal of Computer Applications* 2015;113:1-5. DOI
25. Lin JY, Juan JST. Fault-tolerant visual 2-secret sharing scheme. In Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering, 2017. p. 517-25. DOI
26. Nakajima M, Yamaguchi Y. Enhancing registration tolerance of extended visual cryptography for natural images. *J Electron Imag* 2004;13:654. DOI
27. Wang D, Dong L, Li X. Towards shift tolerant visual secret sharing schemes. *IEEE Trans Inform Forensic Secur* 2011;6:323-37. DOI
28. Yang CN, Peng AG, Chen TS. MTVSS: (M)isalignment (T)olerant (V)isual (S)ecret (S)haring on resolving alignment difficulty. *Signal Processing* 2009;89:1602-24. DOI
29. Chen SK, Lin JC. Fault-tolerant and progressive transmission of images. *Pattern Recognition* 2005;38:2466-71. DOI
30. Chung YC, Ou JH, Juan JST. Fault-tolerant visual secret sharing scheme using meaningful shares. In Proceedings of 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), 2019. p. 1-6. DOI