Multi-secrets Visual Secret Sharing

[1]Tzung-Her Chen, [1]Kai-Hsiang Tsao, and [1]Chang-Sian Wu

[1]Department of Computer Science and Information Engineering National Chiayi University, Chiayi City, Taiwan 60004, R.O.C.

Visual secret sharing (VSS) is the well-known technique that encodes a secret image into several share images and, afterward, decodes the secret by superimposing the share images and recognizing by the human visual system. In this paper, a multi-secrets VSS which is extended from traditional VSS is presented. The codebook of traditional (2,2) VSS is adopted to generate share images macroblock by macroblock in such a way that multiple secret images are turned into only two share images and decode all the secrets one by one by stacking two of share images in a way of shifting.

***Keywords*—Visual Secret Sharing (VSS), Multi-Secrets,**

I. INTRODUCTION

As network technology has been greatly advanced, much more information is transmitted via the Internet conveniently and rapidly. At the same time, the security issue is a crucial problem in the transmission process. For example, the information may be intercepted from transmission process. Consequently, the secret sharing schemes are proposed to be one of candidates for solving this security concern.

Since 1979, Shamir [1] and Blakley [2] presented the concept of the (t, r) threshold secret sharing scheme firstly. The meaning of (t, r) secret sharing is encoding the secret information into t shares. When the users want to obtain the secret information, they should gather at least t shares, where $t \leqq r$. In 1995, Noar and Shamir [3] proposed the visual secret sharing scheme (VSS) which encoded the secret image into two share images using the codebook shown in Table 1, for example. For simplicity, (2,2) VSS is taken for example hereafter. According to the color, white or black, of a secret pixel, the white or black group in the codebook is determined. Then, one set of 2×2 sub-pixels should be chosen from six codewords randomly to generate share S^A and share S^B. Therefore, the secret information can be decoded by directly superimposing two share images without any extra computation. Since the traditional VSS can only encode a secret image at a time, there were many schemes [4]-[9] presented by encoding more than one secret images.

In 2003, Wang et al. [5] proposed the shift visual cryptography. Two secret images are encoded into two share images and all the secrets can be revealed by keeping one of two shares and shifting the other share several columns and rows in stacking phase. In 2007, Luo et al. [4] proposed the watermarking-based VSS in which two secret images can be encoded in this scheme and, later, the secrets can be recovered by keeping one of two shares and shifting the other share a half of image to bottom in stacking process. Fang [9] proposed the reversible style VSS. The scheme can encode two secrets into two shares. In decoding phase, the first secret can be recovered by superimposing two share images directly. The second secret is revealed by flipping one of the two shares and superimposing with the other. To encode more secrets in VSS, Wang et al. [8] proposed multilevel VSS which can encode three secrets into four shares, and the number of secrets which can be revealed is proportional to the number of shares engaged in the decoding phase with different level of contrast. Wu and Chang [6] and Shyu et al. [7] proposed the multi-secrets VSS using circular shares which can encode multiple secrets and, later, rotate the shares from various degrees to reveal the secrets. Wu and Chang's scheme [6] can encode two secrets into two circular share images based on the specific codebook redesigned and to recover the secrets by rotating two shares different specific degrees. Shyu et al.'s scheme [7] encoded four secrets into two circular share images and, later, recovered these secrets by keeping one of two share images and rotating the other share images 0, 90, 180, 270 degrees. Note that the proposed schemes in [6]-[9] are based on the redesigned codebooks. Sometimes, the design of a codebook for specific VSS applications is not trivial. Furthermore, all of these schemes are not only have to redesign the codebook before encoding process and, unfortunately, the pixel expansion becomes so dramatically worse that it must influence with the overhead of transmission bandwidth over the networks.

In this paper, the proposed scheme can encode more secret images into two share images based on traditional (2,2) VSS. In particular, it can encode a secret image and n extra secret images into two share images. In order to decode all the secrets, users superimpose two share images according to the specific positions.

Taking practicality into account, in recent years, plenty of VSS technique for halftone images [10], gray-level images [11], [12], and color images [13] have been developed. To meet this trend, the experimental results demonstrate the feasibility of the proposed scheme by means of encoding not only binary secret images but also color secret images.

II. THE PROPOSED SCHEME

In this section, a VSS scheme of encoding multi-secrets is described. The proposed scheme aims at encoding an original secret image *SI* by extending traditional VSS in such a way that certain of extra secret images *ESI*s can be further encoded into the same two share images, say S^A and S^B. In decoding process, all secret images can be disclosed in a way of shifting operations. The proposed scheme encompasses three major phases: decomposition phase, encoding phase, and decoding

phase. Before describing the details of the proposed scheme, the related functions are defined as follows:

***Definition* 1**: $f_{decomp}(\cdot): X_{uv} \leftarrow f_{decomp}(Y, a, b)$

X_{uv} are the outputs of the function $f_{decomp}(\cdot)$ with the inputs Y, a and b, where $f_{decomp}(\cdot)$ is that the function divides an input image Y into $a \times b$ sub-blocks X_{uv} with the same size. Note that u and v are the row index and the column index of the block divided from Y, precisely, u=0,1,…,a-1 and v=0,1,…,b-1.

***Definition* 2**: $f_{VSS}(\cdot): X^{\alpha} \| X^{\beta} \leftarrow f_{VSS}(Y)$

Two subpixel blocks X^{α} and X^{β} are the outputs of the function $f_{VSS}(\cdot)$ with the input Y, where $f_{VSS}(\cdot)$ is the (2,2) VSS process which input a secret pixel and output two share subpixel blocks X^{α} and X^{β} according to (2,2) VSS codebook.

***Definition* 3**: $\overline{f}_{VSS}(,): X \leftarrow \overline{f}_{VSS}(Y, Z)$

A share image X is the output of the function $\overline{f}_{VSS}(,)$ with the inputs Y one share image and Z a secret image, where $\overline{f}_{VSS}(,)$ is based on (2,2) VSS process which input a share subpixel and a secret pixel, and output a share subpixel blocks X^{α} and X^{β} according to (2,2) VSS codebook.

Decomposition phase

Divide an original secret image SI into n sub-blocks with the same size by the function $SI_{uv} \leftarrow f_{decomp}(SI, h, w)$.

Assume the size of SI is $m \times m$ in which SI is divided into h sub-blocks horizontally and w sub-blocks vertically, i.e., $h \times w = n$. The decomposed sub-blocks are $SI_{00}\|SI_{01}\|\ldots\|SI_{(0)(w-1)}\|SI_{10}\|\ldots\|SI_{(h-1)(w-1)}$. Therefore, the size of each sub-block is $(m/w) \times (m/h)$.

For example, the secret image SI with the size of 256×256 is divided into 4 sub-blocks $SI_{00}\|SI_{01}\|SI_{10}\|SI_{11}$ by the function $SI_{uv} \leftarrow f_{decomp}(SI, 2, 2)$ as shown in Fig. 1. The size of these 4 sub-blocks is 128×128.

Encoding phase

According to (2,2) VSS, SI is turned into two share images $S^A = \{S^A_{00} \| S^A_{01} \| \ldots \| S^A_{(0)(w-1)} \| S^A_{10} \| \ldots \| S^A_{(h-1)(w-1)}\}$ and $S^B = \{S^B_{00} \| S^B_{01} \| \ldots \| S^B_{(0)(w-1)} \| S^B_{10} \| \ldots \| S^B_{(h-1)(w-1)}\}$ with the four times larger size of SI.

Step 1: $S^A_{00} \| S^B_{00} \leftarrow f_{VSS}(SI_{00})$. Encoding the sub-block SI_{00} of the original secret image SI to generate the sub-block S^A_{00} and S^B_{00} of the share images S^A and S^B according to traditional (2,2) VSS. That is, a pixel $SI_{00}(i, j)$ is turned into $S^A_{00}(i', j')$ and $S^B_{00}(i', j')$ of S^A and S^B. Note that (i, j) is the position of the pixel of the sub-block with $1 \le i \le (m/w)$ and $1 \le j \le (m/h)$, and (i', j') is the 2×2 sub-pixels with (2i, 2j), (2i+1, 2j), (2i, 2j+1), and (2i+1, 2j+1). For example, the diagram with h = 2 and w = 2 for **Step 1** is shown in Fig. 2(a).

Step 2: $S^B_{01} \leftarrow \overline{f}_{VSS}(S^A_{00}, ESI_1)$. According to the extra secret image EI_1 and the sub-block S^A_{00} of share image S^A, the sub-block S^B_{01} of share image S^B is generated. The diagram for **Step 2** is shown in Fig. 2(b). For example, if the pixel $S_{00}(i, j)$ is white, and one of six codewords in the white group of Table 1 is randomly selected to encode $S_{00}(i, j)$. Assume that selected the set of codewords is {▚,▞} for $S^A_{00}(i', j')$ and $S^B_{00}(i', j')$. And in this step, according to the sub-block S^A_{00}, if the pixel $ESI_1(i, j)$ is black, then 2×2 block of $S^B_{01}(i', j')$ will be obtained by the codeword ▟ in black group of Table 1. Inversly, suppose that the pixel $ESI_1(i, j)$ is white, then 2×2 block of $S^B_{01}(i', j')$ will be obtained by the codeword ▚ in white group of Table 1. Hence, the sub-pixels $S^A_{00}(i', j')$ superimposes with $S^B_{01}(i', j')$ can be revealed the pixel $ESI_1(i, j)$ visually.

Step 3: $S^A_{01} \leftarrow \overline{f}_{VSS}(S^B_{01}, SI_{01})$. According to the next sub-block SI_{01} and the previous share image block S^B_{01} to generate the sub-block S^A_{01} of the share image S^A. The diagram of **Step 3** is shown in Fig. 2(c). For instance, if the pixel $S_{01}(i, j)$ is white and the pixel $S^B_{01}(i, j)$ has been selected as ▚, then the codeword ▚ as 2×2 block of $S^A_{01}(i', j')$ is allocated. When the pixel $S_{01}(i, j)$ is black the sub-pixels $S^B_{01}(i', j')$ has been selected as ▞, then the codeword ▟ as 2× 2 block of $S^A_{01}(i', j')$ is allocated. Consequently, if superimposing $S^A_{01}(i', j')$ and $S^B_{01}(i', j')$, the pixel $S_{01}(i, j)$ can be revealed visually.

Step 4: Repeat the procedures of Step 2 and Step 3 to create the other sub-blocks. The encoding algorithm is shown below. Note that k means the k-th extra secret image, and the colors of the white and black pixel are respectively represented as "0" and "1".

***Algorithm* 1:** Encoding

Input: A binary original secret image SI={$SI(i, j)|SI(i, j)$=0 or 1, $1 \le i \le m$, $1 \le j \le m$} The extra secret images ESI_k={$ESI_k(i, j)|ESI_k(i, j)$=0 or 1, $1 \le i \le m/h$, $1 \le j \le m/w$}, where $1 \le k \le (h \times w)$-1, h and w are the number of horizontal and vertical division.

Output: Two share images S^A={$S^A(i, j)|S^A(i, j)$=0 or 1, $1 \le i \le 2m$, $1 \le j \le 2m$} and S^B={$S^B(i, j)|S^B(i, j)$=0 or 1, $1 \le i \le 2m$, $1 \le j \le 2m$}

//Divide the secret image SI into $h \times w$ blocks with equal size

$SI_{00} \| SI_{01} \| \ldots \| SI_{(0)(w-1)} \| SI_{10} \| \ldots \| SI_{(h-1)(w-1)} \leftarrow f_{decomp}(SI, h, w)$

//Encode the secret images to two share images

$S^A_{00} \| S^B_{00} \leftarrow f_{VSS}(SI_{00})$ //Step 1

k=1

for x=0 to (h-1)
for y=0 to (w-1)
if ($k \leqq (h \times w)-1$){

$S^B_{((\frac{w \cdot x+y+1}{w}) \bmod h)((y+1) \bmod w)} \leftarrow \overline{f}_{VSS}(S^A_{00}, ESI_k)$ //Step 2

$S^A_{((\frac{w \cdot x+y+1}{w}) \bmod h)((y+1) \bmod w)} \leftarrow$
$\overline{f}_{VSS}(S^B_{((\frac{w \cdot x+y+1}{w}) \bmod h)((y+1) \bmod w)}, SI_{((\frac{w \cdot x+y+1}{w}) \bmod h)((y+1) \bmod w)})$ //Step 3

k=k+1
}

Decoding phase

In this phase, the users can stack two share images S^A and S^B regularly to reconstruct the secret image SI, precisely, the reconstructed image SI' with the size of $2m \times 2m$. To reconstruct the extra secret images, the users shifts and stacks S^A and S^B to reveal more secrets. The operations for reconstructing ESIs are shown in ***Algorithm* 2**. And the stacking diagram is shown in Fig. 3. Eventually, all secrets including the information of SI and ESI_k can be recovered.

***Algorithm* 2**: *Decoding*

Input: Two share images $S^A=\{S^A(i, j)|S^A(i, j)=0$ or 1, $1 \leqq i \leqq 2m$, $1 \leqq j \leqq 2m\}$ and $S^B=\{S^B(i, j)|S^B(i, j)=0$ or 1, $1 \leqq i \leqq 2m$, $1 \leqq j \leqq 2m\}$

Output: An original secret image $SI'=\{SI'(i, j)|SI'(i, j)=0$ or 1, $1 \leqq i \leqq 2m$, $1 \leqq j \leqq 2m\}$
The extra secret images $ESI_k'=\{ESI_k'(i, j)|ESI_k'(i, j)=0$ or 1, $1 \leqq i \leqq 2m/h$, $1 \leqq j \leqq 2m/w\}$, where $1 \leqq k \leqq (h \times w)-1$, h and w are the number of horizontal and vertical division.

//Decoding the original secret image//

$SI' = S^A \oplus S^B$ //$\oplus$ is the blockwise OR operation for two matrices

//Decoding the extra secret images

for x=0 to (h-1)
for y=0 to (w-1)
if ($k \leqq (h \times w)-1$){

output $ESI'_k = S^A_{00} \oplus S^B_{((\frac{w \cdot x+y+1}{w}) \bmod h)\ ((y+1)\ \bmod w)}$

k=k+1
}

III. Experimental Results and Further Discussions

The experimental results demonstrate two examples using different numbers of extra secret images. In the first experiment, four secrets were encoded first and, later decoded one by one by stacking two of shares in a way of a series of shifting operations. Fig. 4 shows all the images: (a) the original secret image with the size of 256x256 and (b) - (d) three extra secret images with the size of 128×128. The number of horizontal and vertical divisions are both 2, i.e., h = w = 2. After the encoding process, two share images S^A and S^B shown in Fig. 4(e) and (f) are obtained. At last, when the user stacks the share image directly, the original secret is reconstructed as shown in Fig. 4(g). Besides, the extra secrets shown in Fig. 2(h) - (j) are reconstructed by stacking S^A_{00} with S^B_{01}, S^A_{00} with S^B_{10}, and S^A_{00} with S^B_{11}, respectively.

In the second experiment, there are six secret images to encode into two share images with h = 3 and w = 2. The results are shown in Fig. 5 including (a) the original secret image with the size of 256x256, and (b) - (f) five extra secret images with the size of 128x85. After the encoding process, two share images S^A and S^B shown in Fig. 5(g) and (h) are obtained. When the user stacks the share image directly, the first secret is reconstructed as shown in Fig. 5(i). Besides, the extra secrets shown in Fig. 5(j) - (n) are reconstructed by stacking S^A_{00} with S^B_{01}, S^A_{00} with S^B_{10}, S^A_{00} with S^B_{11}, S^A_{00} with S^B_{20}, and S^A_{00} with S^B_{21}, respectively.

Discussion

The performance of the proposed scheme is further discussed as below. The comparison table is shown in Table 2 to demonstrate the advantages of the proposed scheme.

Property* 1**: ***No extra codebook redesigned

The proposed scheme does not redesign a new codebook prior to encoding process. Precisely, the basic codebook used in traditional VSS can be used directly.

Property* 2**: ***No extra pixel expansion introduced

Carry on ***Property* 1**, no extra pixel expansion occurs such that the proposed scheme does not worsen the drawback of traditional VSS.

Property* 3**: ***Multiple secrets encoded

Obviously, the proposed scheme can encode more secrets than traditional VSS. The larger $n = h \times w$; the more secrets can be encoded in the same carriers, i.e., the quantity of share images.

Property* 4**: ***Bandwidth and storage saving

The bandwidth and storage complexity depends on the size of share images, precisely pixel expansion. In this scheme, it also benefits from encoding more secrets into the secret images. Carry on ***Property* 1**, **2** and **3** the performance in terms of bandwidth and storage is efficient.

Property* 5**: ***Wide image format

Since, the proposed scheme can be used to encode not only binary images but also gray-level or color images that the present scheme is, thus, of wide use.

IV. Conclusions

In this paper, a multi-secrets VSS scheme based on traditional (2,2) VSS is proposed. Taking bandwidth and storage into consideration, how to turn more secret images into the same share images is critical for the VSS scheme. With the codebook of traditional VSS, the proposed scheme does not need to redesign a new codebook which often introduces more pixel expansion to meet the new demand.

Compared to the related works, the present VSS scheme with the capability of encoding more than one secrets at once does work efficiently.

ACKNOWLEDGMENT

This research was partly supported by National Science Council under contract NSC 95-2221-E-415-009-MY3.

REFERENCES

[1] A. Shamir, "How to share a secret," *Communications of the ACM*, Vol. 22, pp. 612 - 613, 1979.
[2] G. R. Blakley, "Safeguarding cryptography keys," *Proceedings of AFIPS 1979 National Computer Conference*, Vol. 48, New York, USA, pp. 313 - 317, 1979.
[3] M. Naor and A. Shamir, "Visual cryptography," *Proceedings of Advances in Cryptology: Eurocrypt94, Lecture Notes in Computer Science*, Vol. 950, pp. 1 - 12, 1995.
[4] H. Luo, J. S. Pan, Z. M. Lu, and B. Y. Liao, "Watermarking-based transparency authentication in visual cryptography," *Proceedings of Seventh International Conference on Intelligent Systems Design and Applications*, pp. 609 - 616, 2007.
[5] D. Wang, P. Luo, L. Yang, D. Qi, and Y. Dai, "Shift visual cryptography scheme of two secret images," *Progress in Natural Science*, Vol. 13, No. 6, pp. 457 - 463, 2003.
[6] H. C. Wu and C. C. Chang, "Sharing visual multi-secrets using circle shares," *Computer Standards & Interfaces*, Vol. 28, Issue 1, pp. 123 - 135, 2005.
[7] S. J. Shyu, S. Y. Huanga,Y. K. Lee, R. Z. Wang, and K. Chen, "Sharing multiple secrets in visual cryptography," Pattern Recognition, Vol. 40, Issue 12, pp. 3633 - 3651, 2007.
[8] R. Z. Wang, Y. K. Lee, S. Y. Huang, and T. L. Chia, "Multilevel visual secret sharing," *Proceedings of the Second International Conference on Innovative Computing, Information and Control, Kumamoto, Japan*, pp. 283 - 283, 2007.
[9] W. P. Fang, "Visual cryptography in reversible style," *Proceedings of the Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kaohsiung, Taiwan*, pp.519 - 524, 2007.
[10] Z. Zhi, G. R. Arce, and G. D. Crescenzo, "Halftone visual cryptography," *IEEE Transactions on Image Processing*, Vol. 15, No. 8, pp. 2441 - 2453, 2006.
[11] C. C. Lin and W. H. Tsai, "Visual cryptography for graylevel images by dithering techniques", *Pattern Recognition Letters*, Vol. 24, pp. 349 - 358, 2003.
[12] C. Blundo, A. D. Santis, and M. Naor, "Visual cryptography for grey level images," *Information Processing Letters*, Vol. 75, Issue 6, pp. 255 - 259, 2000.
[13] D. Jin, W.Q. Yan, and M.S. Kankanhalli, "Progressive color visual cryptography", *Journal of Electronic Imaging*, Vol. 14, Issue 3, pp. 033019-1 - 033019-13, 2005.

TABLE I
(2,2) VSS CODEBOOK

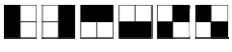
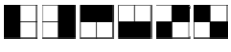
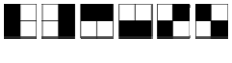
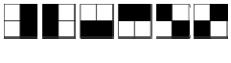
Secret pixel	white group	black group
Share S^A		
Share S^B		
Stacked result		

TABLE II
COMPARISON BETWEEN THE RELATED WORKS AND THE PROPOSED SCHEME

Proposed scheme	Codebook redesign	Pixel expansion	Number of secret images	Type of share images	Image format
Naor and Shamir [3]	Yes	4	1	Square	Binary
Wang et al. [5]	No	4	2	Square	Binary
Luo et al. [4]	No	4	2	Square	Binary
Fang [9]	Yes	9	2	Square	Binary
Wang et al. [8]	Yes	9	3	Square	Binary
Wu and Chang [6]	Yes	4	2	Circular	Binary
Shyu et al. [7]	Yes	8	4	Circular	Binary
Ours	No	4	$n(n \geqq 2)$	Square	Binary Gray Color

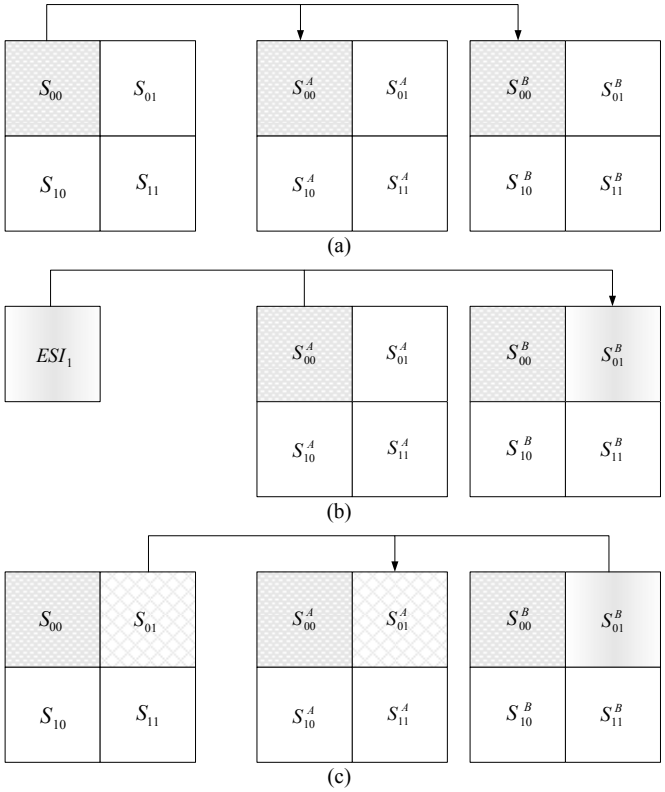


Fig. 2. The diagrams in the encoding phase for (a) Step 1, (b) Step 2, and (c) Step 3.

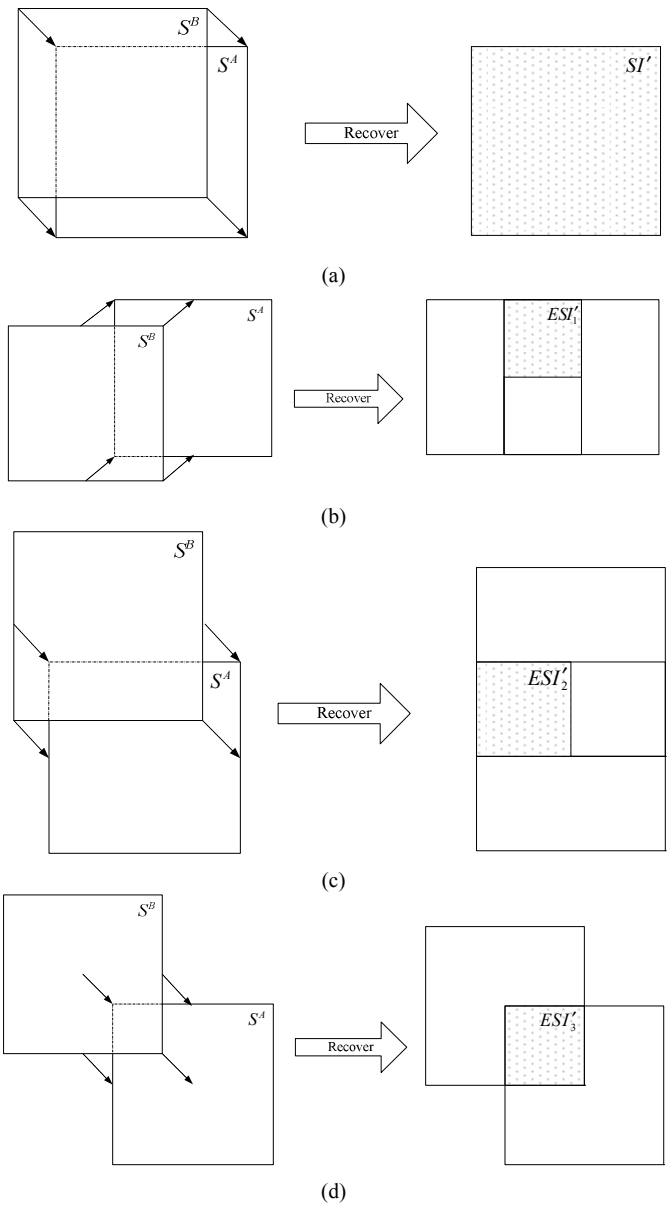
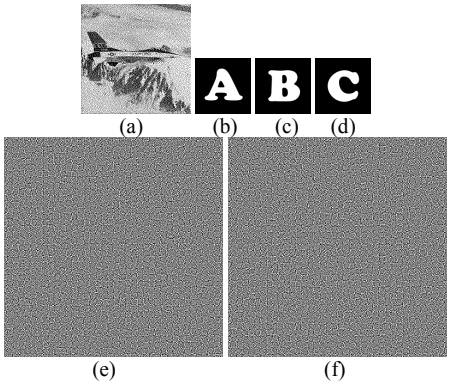


Fig. 3. The diagrams of recovering secrets: (a) Recovering an original secret image SI' (b) – (d) Recovering the extra secret images ESI_1', ESI_2', ESI_3'.



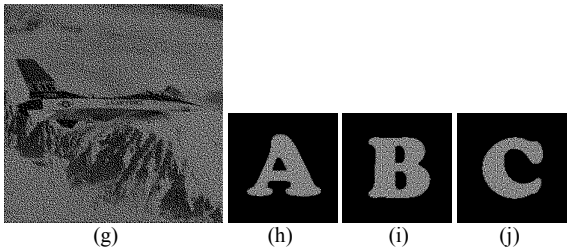


Fig. 4. (a) The original secret image (256×256), (b) - (d) The extra secret images (128×128), (e) - (f) Share image S^A and S^B (512×512), (g) Reconstructed original secret (512×512), and (h) - (j) Reconstructed extra secrets (256×256).

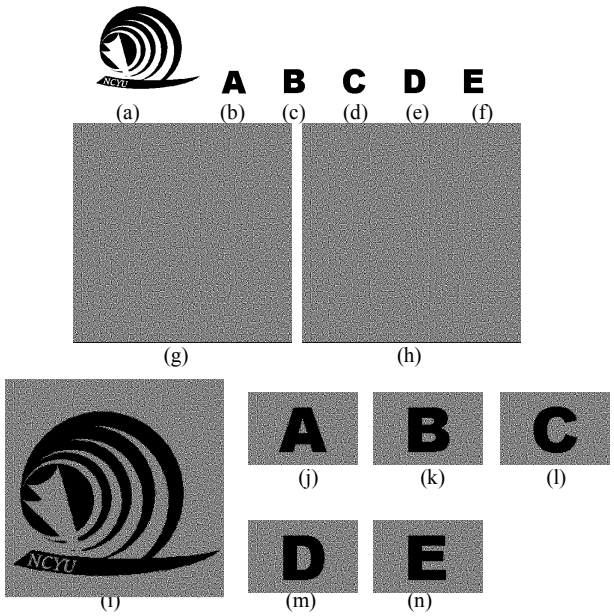


Fig. 5. (a) The original secret image (256×256), (b) - (f) The additional secret images (128×85), (g) - (h) Share image S^A and S^B (512×512), (i) Reconstructed original secret (512×512), and (j) - (n) Reconstructed extra secrets (256×170).

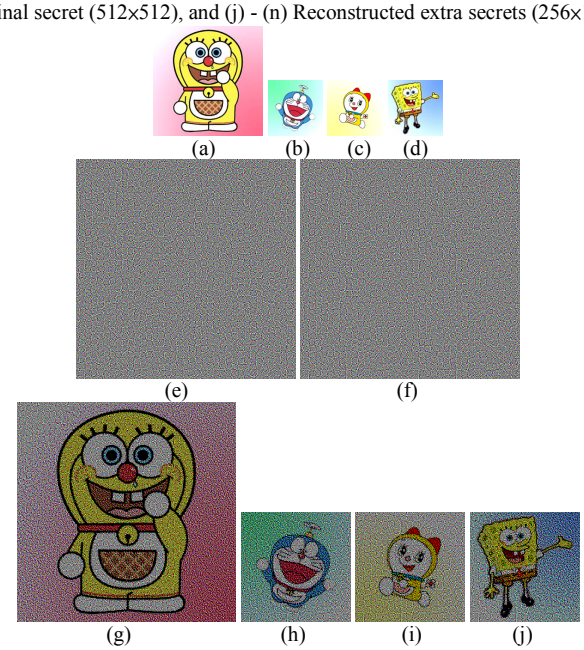


Fig. 6. (a) The original color secret image (256×256), (b) - (d) The extra color secret images (128×128), (e) - (f) Color share image S^A and S^B (512×512), (g) Reconstructed original color secret (512×512), and (h) - (j) Reconstructed extra color secrets (256×256).