Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 40 (2007) 3633–3651

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/pr

# Sharing multiple secrets in visual cryptography

Shyong Jian Shyu[a,*], Shih-Yu Huang[a], Yeuan-Kuen Lee[a], Ran-Zan Wang[b], Kun Chen[a]

[a]*Department of Computer Science and Information Engineering, Ming Chuan University, Gwei-Shan, Taoyuan 333, Taiwan*
[b]*Department of Computer and Communication Engineering, Ming Chuan University, Gwei-Shan, Taoyuan 333, Taiwan*

Received 10 May 2006; received in revised form 6 February 2007; accepted 14 March 2007

**Abstract**

The secret sharing schemes in conventional visual cryptography are characterized by encoding one shared secret into a set of random transparencies which reveal the secret to the human visual system when they are superimposed. In this paper, we propose a visual secret sharing scheme that encodes a set of $x \geqslant 2$ secrets into two circle shares such that none of any single share leaks the secrets and the $x$ secrets can be obtained one by one by stacking the first share and the rotated second shares with $x$ different rotation angles. This is the first true result that discusses the sharing ability in visual cryptography up to any general number of multiple secrets in two circle shares.

*Keywords:* Secret sharing; Visual cryptography; Visual multi-secret sharing

## 1. Introduction

To prevent digital contents from being intercepted by unauthorized parties is a critical demand in information security. With the increasing popularity of the Internet, which makes possible the instant access or distribution of digital contents, such a demand becomes even more significant. Traditional cryptographic skills recommend plenty of solutions by encrypting the digital data into some ciphertext that cannot be recognized by illegal intruders. Yet the decryption of the protected ciphertext needs computations. Generally speaking, the cost or efficiency of the hardware/software performing the decoding computations is mostly proportional to the security of the encryption algorithm. Thus, when we demand a stronger security, the decoding devices/computations become more costly or inefficient.

Visual cryptography proposed by Naor and Shamir [1] discloses the possibility for using human visual ability to perform the decryption process. Specifically, one secret image is encoded into two shares which are seemingly random pictures. By xeroxing them onto transparencies, the dealer distributes the two random transparencies to two participants (one share for each participant). Each participant cannot tell the secret from his own transparency, but when the two participants superimpose their transparencies pixel by pixel, they recognize the secret from the superimposed result by their visual system. Neither computational devices nor cryptographic knowledge is required for the decryption process.

With such an interesting characteristic that the decryption process is by the human visual system only, instead of any computational device, visual cryptography attracts much attention from researchers. In particular, it is much useful in situations where computing devices are not available or not possible to use. Naor and Shamir [1] first presented *k out of n visual secret sharing schemes* which ensure that the secret is concealed from groups of less than *k* participants, while it can be seen by groups of at least *k* participants when they stack their shares altogether. Since this pioneer research, many theoretical results on the *construction* or *contrast* (the relative difference between the reconstructed white and black pixels in the superimposed image) of visual secret sharing schemes for binary images have been proposed in the literature [2–5]. Some studies [6–8] focused on the practical realization of visual cryptographic schemes for gray-level or color images. So far, the above-mentioned results concern sharing “one” secret in a visual sense.

Wu and Chen [9] might be the first researchers to consider the problem of sharing two secret images in two shares in visual cryptography. They concealed two secret binary images into

[*] Corresponding author. Tel.: +886 3 3507001x3402; fax: +886 3 3593874.
*E-mail address:* sjshyu@mcu.edu.tw (S.J. Shyu).

0031-3203/$30.00 © 2007 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.03.012

two seemingly random shares, namely $S_1$ and $S_2$, such that the first secret can be seen by stacking the two shares, denoted by $S_1 \otimes S_2$, and the second secret can be obtained by $S_1^\theta \otimes S_2$ where $\otimes$ denotes the superimposition operation and $S_1^\theta$ is the result of rotating $S_1$ $\theta$ counter-clockwise. $S_1$ and $S_2$ are in the shape of squares of the same size. In order to align the encoded pixels on $S_1$ and $S_2$ as well as on $S_1^\theta$ and $S_2$, they designed the rotation angle $\theta$ to be 90°. Nevertheless, it is easy to obtain that $\theta$ can be 180° or 270°.

Wu and Chang [10] refined the idea of Wu and Chen [9] by consciously designing the encoded shares to be *circles* so that the restrictions to the rotating angles ($\theta = 90°$, 180° or 270°) can be removed. Let the two encoded circle shares in their approach be denoted as $A$ and $B$. The first secret is revealed by $A \otimes B$, while the second secret is obtained by $A^{-\theta} \otimes B$ where $\theta$ may be any angle within (0°, 360°) and $A^{-\theta}$ is the result of rotating $A$ $\theta$ clockwise. Both of the studies successfully share two secrets in two shares in a visual sense.

In this paper, we propose a novel visual secret sharing scheme that encodes a set of $x \geqslant 2$ secrets into two circle shares, namely $A$ and $B$, such that none of $A$ or $B$ individually leaks any secret and the $x$ secrets can be obtained one by one by $A \otimes B$, $A^\theta \otimes B$, $A^{2\theta} \otimes B, \ldots, A^{(x-1)\theta} \otimes B$ where $A^{(i-1)\theta}$ is the result of rotating $A$ $(i-1)\theta$ counter-clockwise for $1 \leqslant i \leqslant x$ for $\theta = 360°/x$ and $A^{0°} = A$. This is the first true result of sharing multiple secrets in visual cryptography for any $x \geqslant 2$ secrets in two shares.

The rest of the paper is organized as follows. In Section 2, we briefly review the visual secret sharing schemes in two shares proposed by Naor and Shamir [1], Wu and Chen [9] and Wu and Chang [10], respectively. The proposed scheme for visual multi-secret sharing is described in Section 3. The experimental results and discussions are presented in Section 4. Section 5 gives some concluding remarks.

## 2. Literature review

The approach of Naor and Shamir [1] is presented in Section 2.1. Wu and Chen's approach [9] is briefly described in Section 2.2. Wu and Chang's scheme [10] is discussed in Section 2.3. It is noticed that among the three schemes, the first one aims at sharing one secret, while the latter two are capable of sharing two secrets.

### 2.1. Naor and Shamir's visual secret sharing scheme

The basic idea of Naor and Shamir's encoding scheme [1] for sharing a single pixel, say $p$, in a binary image $P$ into two shares $s_1$ and $s_2$ is illustrated in Table 1. If $p$ is white, the dealer randomly chooses one of the first two rows of Table 1 to encode $s_1$ and $s_2$. If $p$ is black, the dealer randomly chooses one of the last two rows in Table 1 to encode $s_1$ and $s_2$. The possibilities of the two encoding cases are equally likely to occur, independently of whether the original pixel is black or white. Thus, neither $s_1$ nor $s_2$ exposes any clue about the binary color of $p$. When these two shares are stacked together, i.e. $s_1 \otimes s_2$, two black sub-pixels appear if $p$ is black, while one black sub-pixel and one white sub-pixel appear if $p$ is white as indicated in the rightmost column in Table 1. Based upon the contrast between these two kinds of *reconstructed* pixels, our visual system can tell whether $p$ is black or white by observing $s_1 \otimes s_2$.

Table 1
Encoding a binary pixel $p$ into two shares $s_1$ and $s_2$

| $p$ | Probability | $s_1$ | $s_2$ | $s_1 \otimes s_2$ |
|---|---|---|---|---|
| □ | 1/2 | ◧ | ◧ | ◧ |
| | 1/2 | ◨ | ◨ | ◨ |
| ■ | 1/2 | ◧ | ◨ | ■ |
| | 1/2 | ◨ | ◧ | ■ |

Note that $s_1$ (or $s_2$) in Table 1 is not a single pixel, but two sub-pixels. We call $s_1$ (or $s_2$) an *extended block* and the pair ($s_1$, $s_2$) *the pair of two extended blocks* with respect to $p$. The number of the sub-pixels in each of the two extended blocks ($s_1$, $s_2$) for encoding $p$ is referred to as the *pixel expansion*. In Table 1, the pixel expansion is 2. In realistic implementations, it may be chosen as 4 (=2 × 2) in order to retain the aspect ratio of the original secret image. Since there are six possible patterns for a 2 × 2 extended block, all pairs of two extended blocks ($s_1$, $s_2$)'s for encoding a specific binary pixel $p$ (visual one-secret sharing) are summarized in Table 2.

When $p$ is white (black), the dealer randomly chooses one of the first (last) six rows of Table 2 to encode $p$ into $s_1$ and $s_2$. It is seen from the last column of Table 2 that the reconstructed pixel $r = s_1 \otimes s_2$ may contain two white and two black sub-pixels if $p$ is white, or all four black sub-pixels when $p$ is black. When all pixels in $P$ are encoded in this way, where each $p$'s random choice for encoding alternatives is independent, the encoded shares $S_1$ (containing all $s_1$'s) and $S_2$ (containing all $s_2$'s) are indeed random pictures, respectively. When $S_1$ and $S_2$ are superimposed, all of the four sub-pixels are black in the reconstructed blocks corresponding to each black pixel in $P$, while two sub-pixels are white and the other two are black corresponding to each white pixel in $P$. Based upon such a difference, our visual system recognizes the white and black pixels in $P$ from $S_1 \otimes S_2$. We say that the reconstructed image$S_1 \otimes S_2$ *recovers* $P$.

Fig. 1 shows the implementation results of the encoding scheme in Table 2. Fig. 1(a) is a secret binary image $P$, Fig. 1(b) and (c) are the two encoded shares $S_1$ and $S_2$ which are random pictures revealing no information about $P$ and

Table 2
Implementing the visual one secret sharing scheme with a pixel expansion of 4

| $p$ | Probability | $s_1$ | $s_2$ | $r=s_1\otimes s_2$ |
|---|---|---|---|---|
| □ | 1/6 | ▌ | ▌ | ▌ |
| | 1/6 | ▀ | ▀ | ▀ |
| | 1/6 | ▐ | ▐ | ▐ |
| | 1/6 | ▄ | ▄ | ▄ |
| | 1/6 | ▚ | ▚ | ▚ |
| | 1/6 | ▞ | ▞ | ▞ |
| ■ | 1/6 | ▌ | ▐ | █ |
| | 1/6 | ▀ | ▄ | █ |
| | 1/6 | ▐ | ▌ | █ |
| | 1/6 | ▄ | ▀ | █ |
| | 1/6 | ▚ | ▞ | █ |
| | 1/6 | ▞ | ▚ | █ |

Fig. 1(d) illustrates the reconstructed image $S_1 \otimes S_2$ which recovers $P$ visually.

### 2.2. Visual two-secret sharing schemes

Following the research of Naor and Shamir, Wu and Chen [9] developed a visual secret sharing scheme that encrypts two secrets into two shares. Given two $N \times N$ (square) secret binary images $P_1$ and $P_2$, their scheme produces two shares, namely $S_1$ and $S_2$, which reveal no information about $P_1$ or $P_2$ individually. Yet when stacking $S_1$ and $S_2$, we obtain $P_1$ visually; moreover, when stacking $S_1^{90°}$ and $S_2$, we see $P_2$.

Consider a pair of pixels $p_1=P_1[i, j]$ and $p_2=P_2[u, v]$ in $P_1$ and $P_2$, respectively. We refer to $(p_1, p_2)$ as the *corresponding pixels* of $P_1$ and $P_2$ if and only if $i=u$ and $j=v$. Given a set of corresponding pixels $(p_1, p_2)$, Wu and Chen's encoding scheme for visual two-secret sharing in two shares is summarized in Table 3.

It is seen from Table 3 each pair of corresponding pixels $(p_1, p_2)$ of $(P_1, P_2)$ is encoded into extended blocks $s_1$ (as well as $s_1^{90°}$) and $s_2$ in which the pixel expansion is $m=4$. Note that $s_1^{90°}$ is exactly the result of rotating $s_1$ 90° counterclockwise. We explain how Wu and Chen's encoding scheme works by using a simple example. Assume that the two secret images $P_1$ and $P_2$ are composed in a square of $12 \times 12$ pixels. Then, the two encoded shares $S_1$ and $S_2$ are composed in a square of $48 \times 48$ $(48 = 12 \times 4)$ pixels. They first decompose $S_1$ into four triangle-like areas with an equal size as shown in Fig. 2(a). All of the four areas are composed of an equal amount of extended blocks ($2 \times 2$ pixels each) which are indexed as shown in Fig. 2(b) where each triangle-like area contains 36 blocks. Let block $j$ in area $k$ be denoted as $b_j^k$ for $1 \leqslant k \leqslant 4$ and $1 \leqslant j \leqslant 36$. The extended blocks in area I, $b_j^1$, are randomly selected out of those in Fig. 2(c). Each block, say $b_j^t$, in area II, III, IV is assigned to be the same as $b_j^1$ in area I, that is, $b_j^t = b_j^1$ for $t = 2, 3, 4$ and $1 \leqslant j \leqslant 36$.

Let us pay attention to the four pixels at the top-right, top-left, bottom-left and bottom-right corners in sequence (counterclockwise) in $P_1$ and $P_2$. Assume that those pixels in $P_1$ ($P_2$) are □, □, ■, ■ (□, ■, □, ■) as shown in Fig. 3(a) (Fig. 3(b)). Assume that corresponding block $b_{26}^1$ at $S_1$ is randomly determined as ▟, then as mentioned $b_{26}^t$ is ▟ for $2 \leqslant t \leqslant 4$ (see Fig. 3(c)). The above-mentioned pixels in $P_1$ and $P_2$ constitute four sets of corresponding pixels: (□, □), (□, ■), (■, □) and (■, ■). Since $b_{26}^k$ in $S_1$ is ▟ for $1 \leqslant k \leqslant 4$, according to Table 3 the four blocks $b_{26}^1$, $b_{26}^2$, $b_{26}^3$ and $b_{26}^4$, in $S_2$ with respect to the four sets of the corresponding pixels are ▐, ▄, ▀ and ▌, respectively (see the 2nd, 6th, 10th and 14th rows in column $s_2$ of Table 3). Fig. 3(d) illustrates the encoding result of $S_2$. As expected, the four corners in the above-mentioned order in $S_1 \otimes S_2$ reveal □, □, ■, ■, respectively (see Fig. 3(e)) to our visual system. When $S_1$ is rotated as $S_1^{90°}$ as indicated in Fig. 3(f), where all blocks are in the form of $s_1^{90°}$, the four corresponding corners in $S_1^{90°} \otimes S_2$ recover □, ■, □, ■, respectively (see Fig. 3(g)). It is not hard to see that by encoding all pixels in $S_1$ and $S_2$ with respect to the corresponding pixels in $P_1$ and $P_2$ according to Table 3, $P_1$ and $P_2$ can be recovered by $S_1 \otimes S_2$ and $S_1^{90°} \otimes S_2$, respectively.

Note that $S_1$ and $S_2$ are in the shape of squares of the same size. $S_1 \otimes S_2$ reveals $P_1$, while $S_1^{\theta} \otimes S_2$ reveals $P_2$. Wu and Chen set $\theta$ to be 90°. It is easy to extend their idea to design $\theta$ as one of 90°, 180° or 270°, but the other degrees are infeasible. This is because the rotated $S_1$ ($S_1^{\theta}$) cannot be aligned to $S_2$ pixel by pixel when $\theta \neq 0°$, 90°,180° or 270°. Except for the fact that $\theta$ is restricted, there is another pitfall in their scheme: since the encoded pixels in each of areas I, II, III and IV in $S_1$ are exactly the same, $S_1$ is not a random picture. In fact, only 1/4 shares of $S_1$ are purely random pictures.

Based upon the idea of Wu and Chen [9], Wu and Chang [10] devised another visual two-secret sharing scheme that allows the rotation angle to be an arbitrary one between 0° and 360° by adopting circle shares. Given an angle $\theta$ and two secret images $P_1$ and $P_2$, their approach produces two circle shares

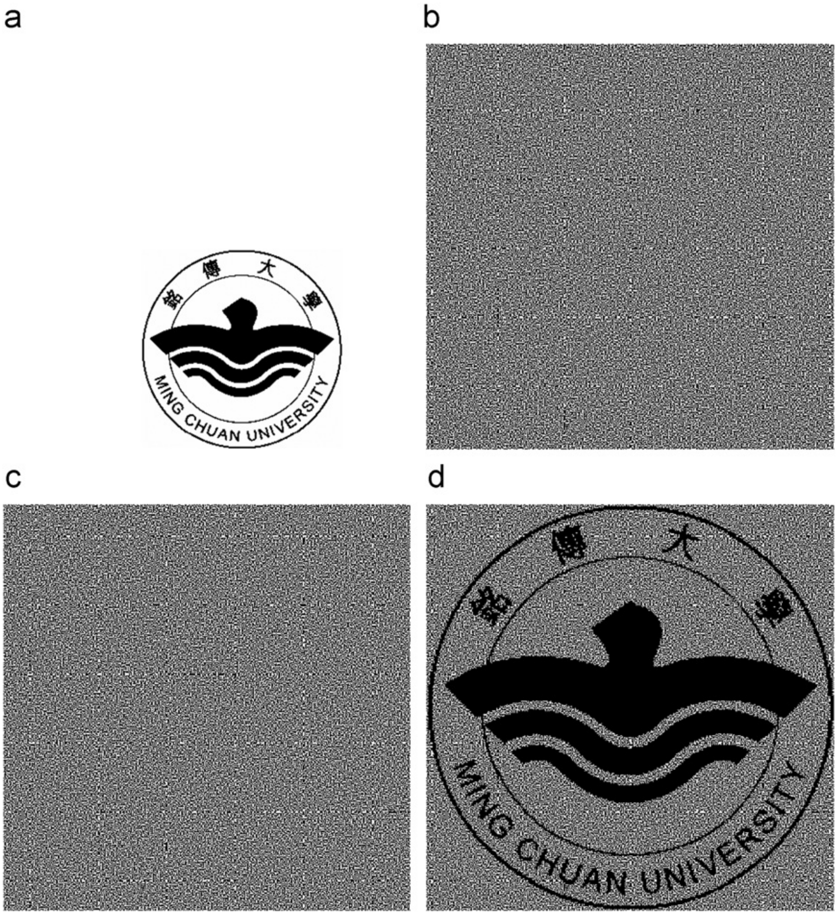


Fig. 1. Implementation results of visual one-secret sharing in two shares: (a) $P$, (b) $S_1$, (c) $S_2$, (d) $S_1 \otimes S_2$.

$A$ and $B$ such that any single $A$ or $B$ is a seemingly random picture which leaks nothing about $P_1$ or $P_2$; while $A \otimes B$ reconstructs $P_1$ and $A^{-\theta} \otimes B$ recovers $P_2$ where $A^{-\theta}$ denotes the result of rotating $A$ $\theta$ clockwise. Note that $A^{-\theta} \otimes B$ is equivalent to $A \otimes B^{\theta}$. Intuitively, it is reasonable to choose circles as the encoded shares since they ease the correct alignments between $A$ and $B$ as well as $A^{-\theta}$ and $B$ pixel by pixel where $0° < \theta < 360°$.

They deliberately decomposed circle share $A$ into $360°/\theta$ areas where each area contains an equal amount of $2 \times 2$ *sector blocks*. Fig. 4(a) shows the four typical patterns for sector blocks, namely $s_1^1$, $s_1^2$, $s_1^3$ and $s_1^4$, used in their approach. That is, the whole circle share $A$ is composed by all these four sector blocks. Note that $s_1^1$ ($s_1^2$, $s_1^3$, $s_1^4$) can be consciously regarded as the result of rotating $s_1^2$ ($s_1^3$, $s_1^4$, $s_1^1$, respectively) 90° counter-clockwise (or $s_1^2$ can be consciously regarded as the result of rotating $s_1^1$ 90° clockwise). We say that $s_1^1$'s ($s_1^2$'s, $s_1^3$'s, $s_1^4$'s) *previous* sector block is $s_1^4$ ($s_1^1$, $s_1^2$, $s_1^3$, respectively) and its *next* sector block is $s_1^2$ ($s_1^3$, $s_1^4$, $s_1^1$, respectively) as summarized in Fig. 4(b).

Let the number of areas in circle share $A$ be $\alpha$ ($=360°/\theta$) and the number of sector blocks in each area be $\beta$. These $\alpha$ areas are indexed clockwise. Let $a_j^k$ be the $j$th sector block in area $k$ in $A$, $1 \leqslant j \leqslant \beta$ and $1 \leqslant k \leqslant \alpha$. At first, the $\beta$ sector blocks in the first area are randomly selected out of those in Fig. 4(a). Then, sector blocks in area $t$ are defined according to those in area $t-1$ by assigning $a_j^t$ as the next sector block of $a_j^{t-1}$, i.e. $a_j^t = next(a_j^{t-1})$ (or $a_j^{t-1} = prev(a_j^t)$) for $1 \leqslant j \leqslant \beta$ and $2 \leqslant t \leqslant \alpha$.

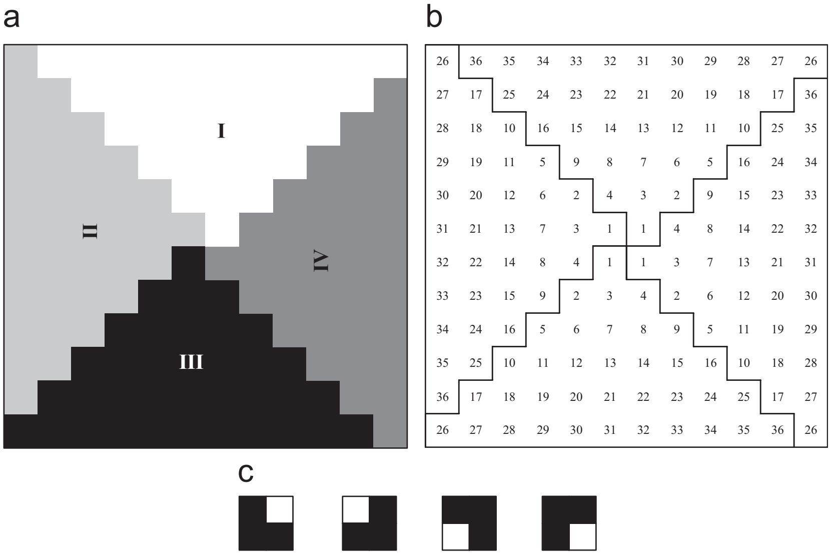
Given a pair of corresponding pixels $p_1$ and $p_2$ in $P_1$ and $P_2$, respectively, each sector block $b_j^k$ in $B$ is determined by $p_1$, $p_2$ and the corresponding block $a_j^k$ in $A$ for $1 \leqslant j \leqslant \beta$ and $1 \leqslant k \leqslant \alpha$. Table 4 summarizes such an encoding scheme.

Note that in Wu and Chen's scheme each extended block $s_2$ in $S_2$ would be superimposed with $s_1$ and $s_1^{90°}$ when $s_1$ is rotated 0° (or fixed) and 90° counter-clockwise, respectively. In Wu and Chang's scheme, each sector block $b_j^k$ in $B$ is superimposed with $a_j^k$ in area $k$ and $a_j^{k-1}$ in $k$'s previous area $k-1$ when $A$ is rotated 0° and $\theta$ clockwise where $a_j^{k-1} = prev(a_j^k)$ (or $a_j^k = next(a_j^{k-1})$). Note that $a_j^{k-1}$ is the result of rotating $a_j^k$ 90° counter-clockwise (or $a_j^k$ is the result of rotating $a_j^{k-1}$ 90° clockwise; see Fig. 4). That means the result of $A^{-\theta} \otimes B$ in Wu and Chang's scheme emulates that of $S_1^{90°} \otimes S_2$ in Wu and Chen's scheme. There is no restriction for $\theta$ to be one of 90°, 180° or 270° merely. Yet there exist some inconsistent situations in some of the areas in $A^{-\theta} \otimes B$ when $\alpha = 360°/\theta > 4$. Interested readers refer to Ref. [10] for details.

Table 3
Wu and Chen's encoding scheme for visual two-secret sharing in two shares

| $p_1$ | $p_2$ | Probability | $s_1$ | $s_1^{90^\circ}$ | $s_2$ | $s_1 \otimes s_2$ | $s_1^{90^\circ} \otimes s_2$ |
|---|---|---|---|---|---|---|---|
| □ | □ | 1/4 | ▙ | ▟ | ▄ | ▙ | ▟ |
| | | 1/4 | ▟ | ▜ | ▐ | ▟ | ▜ |
| | | 1/4 | ▜ | ▛ | ▀ | ▜ | ▛ |
| | | 1/4 | ▛ | ▙ | ▌ | ▛ | ▙ |
| □ | ■ | 1/4 | ▙ | ▟ | ▌ | ▙ | █ |
| | | 1/4 | ▟ | ▜ | ▄ | ▟ | █ |
| | | 1/4 | ▜ | ▛ | ▐ | ▜ | █ |
| | | 1/4 | ▛ | ▙ | ▀ | ▛ | █ |
| ■ | □ | 1/4 | ▙ | ▟ | ▐ | █ | ▟ |
| | | 1/4 | ▟ | ▜ | ▀ | █ | ▜ |
| | | 1/4 | ▜ | ▛ | ▌ | █ | ▛ |
| | | 1/4 | ▛ | ▙ | ▄ | █ | ▙ |
| ■ | ■ | 1/4 | ▙ | ▟ | ▀ | █ | █ |
| | | 1/4 | ▟ | ▜ | ▌ | █ | █ |
| | | 1/4 | ▜ | ▛ | ▄ | █ | █ |
| | | 1/4 | ▛ | ▙ | ▐ | █ | █ |

As mentioned above, square share $S_1$ in Wu and Chen's scheme is not a totally random image. Strictly speaking, neither is circle share $A$ in Wu and Chang's scheme due to the reason that sector block $a_j^t$ is assigned as $next(a_j^{t-1})$ (i.e. the result of rotating $a_j^{t-1}$ 90° clockwise) for $2 \leqslant t \leqslant \alpha$; that is, only sector block $a_j^1$ in the first area is randomly determined from those in Fig. 4(a) for $1 \leqslant j \leqslant \beta$, while the other areas are not. Furthermore, sector blocks in the first area of circle share $A$ (see Fig. 4(a)) in Wu and Chang's scheme (or extended blocks in the first area of square share $S_1$ (see Fig. 2(c)) contain only four patterns, instead of six which is the number of all possible combinations for four



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 26 | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 |
| 27 | 17 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 36 |
| 28 | 18 | 10 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 25 | 35 |
| 29 | 19 | 11 | 5 | 9 | 8 | 7 | 6 | 5 | 16 | 24 | 34 |
| 30 | 20 | 12 | 6 | 2 | 4 | 3 | 2 | 9 | 15 | 23 | 33 |
| 31 | 21 | 13 | 7 | 3 | 1 | 1 | 4 | 8 | 14 | 22 | 32 |
| 32 | 22 | 14 | 8 | 4 | 1 | 1 | 3 | 7 | 13 | 21 | 31 |
| 33 | 23 | 15 | 9 | 2 | 3 | 4 | 2 | 6 | 12 | 20 | 30 |
| 34 | 24 | 16 | 5 | 6 | 7 | 8 | 9 | 5 | 11 | 19 | 29 |
| 35 | 25 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 10 | 18 | 28 |
| 36 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 17 | 27 |
| 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 26 |

Fig. 2. Encoding $S_1$ in Wu and Chen's scheme: (a) Four triangle-like areas. (b) Indexing the blocks in each of the four areas. (c) Blocks to be assigned.

sub-pixels with two white and two black sub-pixels (see Table 2).

## 3. The proposed algorithm

Both of the above-mentioned schemes accomplish visual secret sharing for only two secrets in two shares. In this section we propose a more generalized visual secret sharing scheme for $x \geqslant 2$ secrets in two shares. Our idea is informally illustrated in Section 3.1. The formal encoding algorithm is proposed in Section 3.2.

### 3.1. Informal description of our idea

Let us start by using a simple example. Assume that the number of secret images to be shared is $x = 3$. Let $P_1$, $P_2$ and $P_3$ be the three binary secret images with the same size $h \times w$. Let $p_1$, $p_2$ and $p_3$ denote the corresponding pixels in $P_1$, $P_2$ and $P_3$, respectively. Let $A$ and $B$ denote the two circle shares encoded by our scheme. Our aim is to assure $A \otimes B$ recovers $P_1$, $A^{120^\circ} \otimes B$ recovers $P_2$ and $A^{240^\circ} \otimes B$ recovers $P_3$.

Since there are three secrets, we decompose circle share $A$ and $B$ into three ($x = 3$) *chord-areas* (*chords* for short), respectively, in which the angle of each chord extends up to 120° (=360°/$x$ = 360°/3). Each chord is divided into a set of $2 \times 3$ ($2 \times x$) chord blocks. Let the number of $2 \times 3$ blocks in each chord be $\beta$. Let $a_j^k$ and $b_j^k$ denote block $j$ of chord $k$ in $A$ and $B$, respectively, $1 \leqslant j \leqslant \beta$ and $1 \leqslant k \leqslant 3 (=x)$. The chords are indexed clockwise and the divided blocks in $A$ and $B$ are indexed as shown in Figs. 5(a) and (b), respectively. We call $a_j^k$ and $b_j^k$ the *corresponding blocks* in $A$ and $B$.

#### 3.1.1. Encoding circle share A

We first define three $2 \times 3$ *elementary blocks*, namely $s_A^1$, $s_A^2$ and $s_A^3$, for circle share $A$ as shown in Fig. 6. That is, these elementary blocks are the basic constituents of $A$ and there are one white and five black sub-pixels in each of the elementary blocks.

In order to guarantee the randomness when using $s_A^k$ as a constituent of $A$ for $1 \leqslant k \leqslant 3$, we permute the sub-pixels within $s_A^k$ before assigning $s_A^k$ as a constituent block in $A$. Let $\Sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)$ be a permutation of $\{1, 2, 3, 4, 5, 6\}$ (in which $6 = 2x = 2 \times 3$). We define a function *permute*$(s, \Sigma)$ to re-arrange the sub-pixels in elementary block $s$ by permutation $\Sigma$. Fig. 7(a) shows a certain typical ordering of the sub-pixels in a $2 \times 3$ elementary block $s$ and Fig. 7(b) shows the result of *permute*$(s, \Sigma)$ with $\Sigma = (3, 5, 1, 6, 2, 4)$. Note that the order of the sub-pixels in the elementary blocks can be defined arbitrarily.

We call the set of three blocks $(a_j^1, a_j^2, a_j^3)$ the *related blocks* of the three chords in $A$ for $1 \leqslant j \leqslant \beta$. Obviously, there are totally $\beta$ sets of the related blocks in $A$. For a certain set of related blocks $(a_j^1, a_j^2, a_j^3)$, we generate one permutation, denoted as $\Sigma_j$, and assign $a_j^k$ to be *permute*$(s_A^k, \Sigma_j)$ for $1 \leqslant k \leqslant 3$ and $1 \leqslant j \leqslant \beta$. That is,

$$(a_j^1, a_j^2, a_j^3) \\ = (\mathit{permute}(s_A^1, \Sigma_j), \mathit{permute}(s_A^2, \Sigma_j), \mathit{permute}(s_A^3, \Sigma_j)) \qquad (1)$$

for $1 \leqslant j \leqslant \beta$.

For the purpose of illustration, we show how the first set of the related blocks $(a_1^1, a_1^2, a_1^3)$ in $A$ is encoded. Assume that $\Sigma_1 = (1, 2, 3, 4, 5, 6)$. Fig. 8(a) exposes the results of encoding $(a_j^1, a_j^2, a_j^3)$ in $A$. Note that for this particular $\Sigma_1$ *permute*$(s_A^k, \Sigma_1) = s_A^k$ for $1 \leqslant k \leqslant 3$. In real implementation, a new random permutation $\Sigma_j$ is adopted when encoding $(a_j^1, a_j^2, a_j^3)$ in $A$ for each $j$, $1 \leqslant j \leqslant \beta$.Figs. 8(b) and (c) show the results of $A^{120^\circ}$ and $A^{240^\circ}$, respectively.

Let $[k, j]$ denote the absolute location with respect to block $j$ of chord $k$ in a circle share (see Fig. 9) and $A^\theta[k, j]$ denote the content of block $[k, j]$ in $A^\theta$ (i.e. the results of rotating $A$

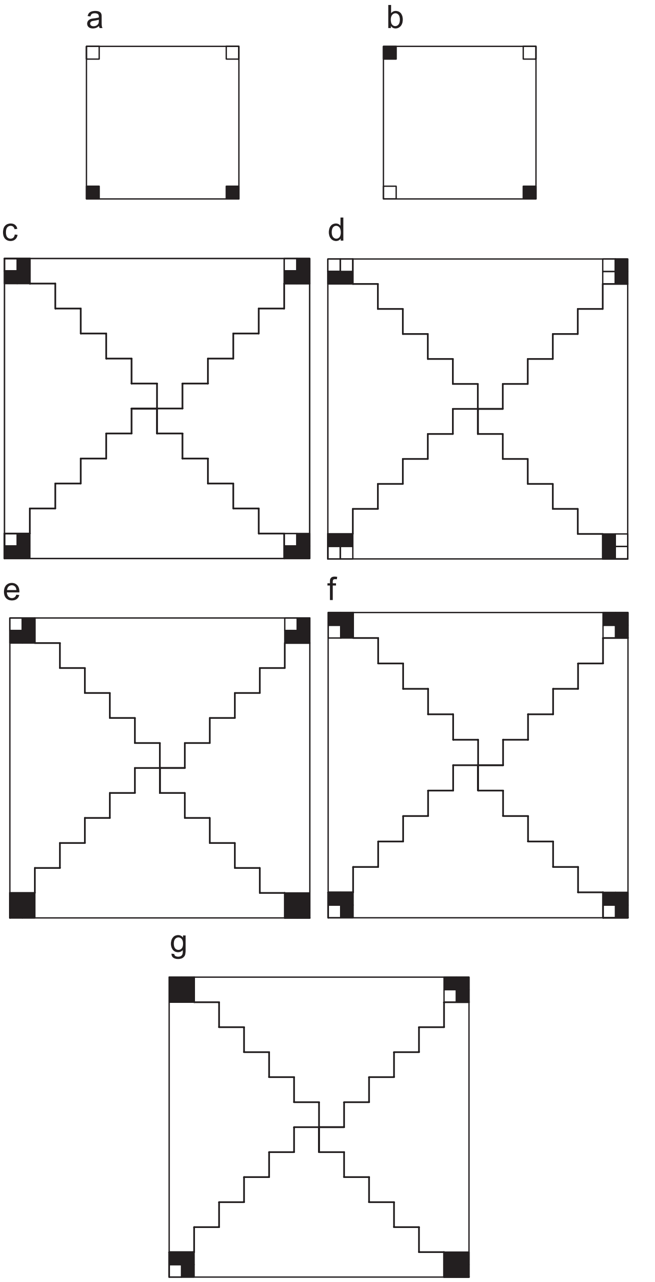


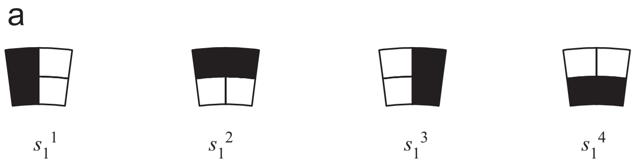
Fig. 3. Example for illustrating the idea of Wu and Chen [9]: (a) $P_1$, (b) $P_2$, (c) $S_1$, (d) $S_2$, (e) $S_1 \otimes S_2$, (f) $S_1^{90^\circ}$, (g) $S_1^{90^\circ} \otimes S_2$.

$\theta$ counter-clockwise) where $1 \leqslant k \leqslant 3(=x)$, $1 \leqslant j \leqslant \beta$ and $\theta = 120^\circ(=360^\circ/3)$ ($A^{0^\circ}[k, j] = A[k, j]$). The relationship among the related blocks is easily seen from Figs. 8 and 9:

$$\begin{aligned} &A[1, j] = a_j^1, \quad A^{120^\circ}[1, j] = a_j^2, \quad A^{240^\circ}[1, j] = a_j^3, \\ &A[2, j] = a_j^2, \quad A^{120^\circ}[2, j] = a_j^3, \quad A^{240^\circ}[2, j] = a_j^1, \\ &A[3, j] = a_j^3, \quad A^{120^\circ}[3, j] = a_j^1, \quad A^{240^\circ}[3, j] = a_j^2 \end{aligned} \tag{2}$$

for $1 \leqslant j \leqslant \beta$.

### 3.1.2. Encoding circle share B



b

| $s$ | $s_1^{\ 1}$ | $s_1^{\ 2}$ | $s_1^{\ 3}$ | $s_1^{\ 4}$ |
|---|---|---|---|---|
| *perv* ($s$) | $s_1^{\ 4}$ | $s_1^{\ 1}$ | $s_1^{\ 2}$ | $s_1^{\ 3}$ |
| *next* ($s$) | $s_1^{\ 2}$ | $s_1^{\ 3}$ | $s_1^{\ 4}$ | $s_1^{\ 1}$ |

Fig. 4. 2 × 2 sector blocks for $A$ in Wu and Chang's approach: (a) 2 × 2 sector blocks for $A$, (b) $prev(s)$ and $next(s)$ of sector block $s$.

First of all, with regard to the three given $h \times w$ secret images $P_1$, $P_2$ and $P_3$, we divide $P_i$ evenly into three $\beta = h \times (w/3) = h \times (w/x)$ *strips*. Let $(p_i)_j^k$ denote the $j$th pixel of strip $k$ in $P_i$ and $(p_1, p_2, p_3)_j^k = ((p_1)_j^k, (p_2)_j^k, (p_3)_j^k)$ be the $j$th corresponding pixels of strip $k$ for $(P_1, P_2, P_3)$ where $1 \leqslant j \leqslant \beta$ and $1 \leqslant i$, $k \leqslant 3$. Each block, say $b_j^k$, in $B$ is determined according to the related blocks $(a_j^1, a_j^2, a_j^3)$ in $A$ and the corresponding pixels $(p_1, p_2, p_3)_j^k$ in $(P_1, P_2, P_3)$ for $1 \leqslant j \leqslant \beta$ and $1 \leqslant k \leqslant 3$.

Consider a particular block $b_j^1$ in the first chord of $B$. Note that the corresponding block $a_j^1$ ($a_j^2$ and $a_j^3$) in $A$ ($A^{120^\circ}$ and $A^{240^\circ}$, respectively) is a permuted result of elementary block $s_A^1$ ($s_A^2$ and $s_A^3$, respectively) according to $\Sigma_j$ which is decided in run time. Our encoding scheme for $b_j^1$ before run time is essentially based upon $(s_A^1, s_A^2, s_A^3)$ and $(p_1, p_2, p_3)_j^1$ as summarized in Table 5.

It is observed from Table 5 that given a set of corresponding pixels $(p_1, p_2, p_3)$, the results of $s_A^1 \otimes s_B$ ($s_A^2 \otimes s_B$, $s_A^3 \otimes s_B$) reveals $p_1$ ($p_2$, $p_3$, respectively) to our visual system. For instance, consider the third row of Table 5 where $(p_1, p_2, p_3) = (\square, \blacksquare, \square)$. When the related blocks in $A$ are in their elementary forms $s_A^1$, $s_A^2$ and $s_A^3$, we design $s_B$ to be ▚so that both $s_A^1 \otimes s_B$ and $s_A^3 \otimes s_B$ reveal one white and five black sub-pixels, while $s_A^2 \otimes s_B$ shows six black sup-pixels. Our eyes recognize $s_A^1 \otimes s_B$ and $s_A^3 \otimes s_B$ as white, while $s_A^2 \otimes s_B$ as black. That means $(p_1, p_2, p_3)$ is recovered by $(s_A^1 \otimes s_B, s_A^2 \otimes s_B, s_A^3 \otimes s_B)$ in a visual sense.

In actual implementation, the set of three related blocks $(a_j^1, a_j^2, a_j^3)$ in $A$ is deliberately assigned as $(permute(s_A^1, \Sigma_j)$, $permute(s_A^2, \Sigma_j)$, $permute(s_A^3, \Sigma_j))$ so that we only need to assign $b_j^1$ to be $permute(s_B, \Sigma_j)$ to preserve the superimposition results designed in Table 5. Then, when we superimpose $a_j^1$ and $b_j^1$, we identify $(p_1)_j^1$ from $a_j^1 \otimes b_j^1$.When we rotate $A$ 120° counter-clockwise, $b_j^1$'s corresponding block in $A^{120^\circ}$ turns out to be $a_j^2$ (i.e. $A^{120^\circ}[1, j]$, see formula (2) and Fig. 9) and $a_j^2 \otimes b_j^1$ reveals $(p_2)_j^1$ in a visual sense. Likewise, when rotating $A$ 240° counter-clockwise, $b_j^1$'s corresponding block in $A^{240^\circ}$ is $a_j^3$ and $a_j^3 \otimes b_j^1$ reveals $(p_3)_j^1$.In general, when rotating $A$ $(i-1)\theta$ counter-clockwise we recognize $a_j^i \otimes b_j^1$ as

Table 4
Wu and Chang's encoding scheme for visual two-secret sharing in two shares

| $p_1$ | $p_2$ | Probability | $s_1$ | $s_1^{-\theta}$ | $s_2$ | $s_1 \otimes s_2$ | $s_1^{-\theta} \otimes s_2$ |
|---|---|---|---|---|---|---|---|
| □ | □ | 1/4 | ■□<br>■□ | ■■<br>□□ | ■■<br>■□ | ■■<br>■□ | ■■<br>■□ |
| | | 1/4 | ■■<br>□□ | □■<br>□■ | ■■<br>□■ | ■■<br>□■ | ■■<br>□■ |
| | | 1/4 | □■<br>□■ | □□<br>■■ | □■<br>■■ | □■<br>■■ | □■<br>■■ |
| | | 1/4 | □□<br>■■ | ■□<br>■□ | ■□<br>■■ | ■□<br>■■ | ■□<br>■■ |
| □ | ■ | 1/4 | ■□<br>■□ | ■■<br>□□ | ■□<br>■■ | ■□<br>■■ | ■■<br>■■ |
| | | 1/4 | ■■<br>□□ | □■<br>□■ | ■■<br>■□ | ■■<br>■□ | ■■<br>■■ |
| | | 1/4 | □■<br>□■ | □□<br>■■ | ■■<br>□■ | ■■<br>□■ | ■■<br>■■ |
| | | 1/4 | □□<br>■■ | ■□<br>■□ | □■<br>■■ | □■<br>■■ | ■■<br>■■ |
| ■ | □ | 1/4 | ■□<br>■□ | ■■<br>□□ | ■■<br>□■ | ■■<br>■■ | ■■<br>□■ |
| | | 1/4 | ■■<br>□□ | □■<br>□■ | □■<br>■■ | ■■<br>■■ | □■<br>■■ |
| | | 1/4 | □■<br>□■ | □□<br>■■ | ■□<br>■■ | ■■<br>■■ | ■□<br>■■ |
| | | 1/4 | □□<br>■■ | ■□<br>■□ | ■■<br>■□ | ■■<br>■■ | ■■<br>■□ |
| ■ | ■ | 1/4 | ■□<br>■□ | ■■<br>□□ | □■<br>■■ | ■■<br>■■ | ■■<br>■■ |
| | | 1/4 | ■■<br>□□ | □■<br>□■ | ■□<br>■■ | ■■<br>■■ | ■■<br>■■ |
| | | 1/4 | □■<br>□■ | □□<br>■■ | ■■<br>■□ | ■■<br>■■ | ■■<br>■■ |
| | | 1/4 | □□<br>■■ | ■□<br>■□ | ■■<br>□■ | ■■<br>■■ | ■■<br>■■ |

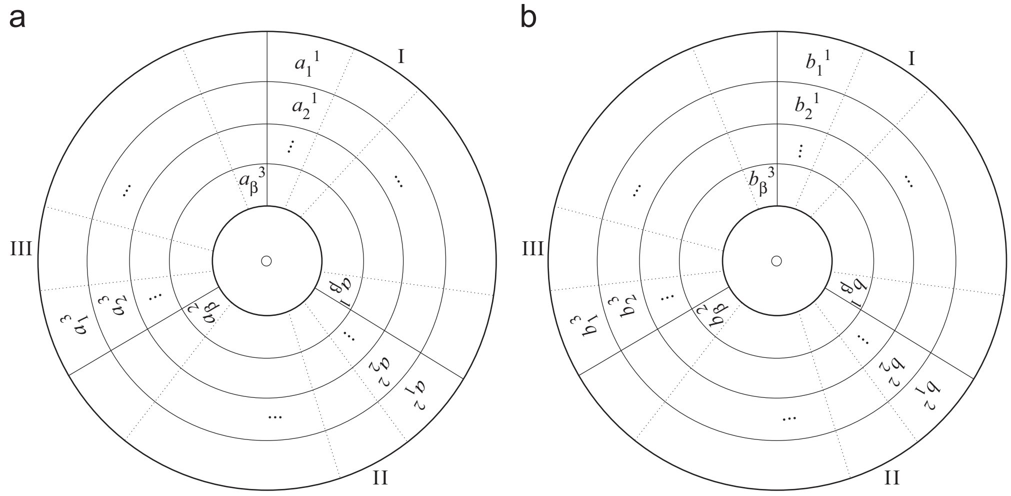


Fig. 5. Decomposing circle shares $A$ and $B$ into chords which are further divided into blocks: (a) $A$, (b) $B$.

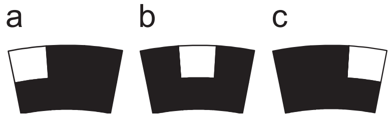


Fig. 6. Elementary blocks for circle share $A$ for sharing 3 secrets: (a) ${S_A}^1$, (b) ${S_A}^2$, (c) ${S_A}^3$.

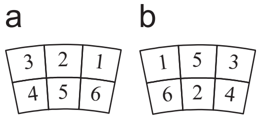


Fig. 7. Sub-pixels in 2 × 3 elementary block $s$ and $permute(s, \Sigma)$: (a) a certain ordering of the sub-pixels in $s$, (b) the ordering of those in $permute(s, \Sigma)$ where $\Sigma = (3, 5, 1, 6, 2, 4)$.

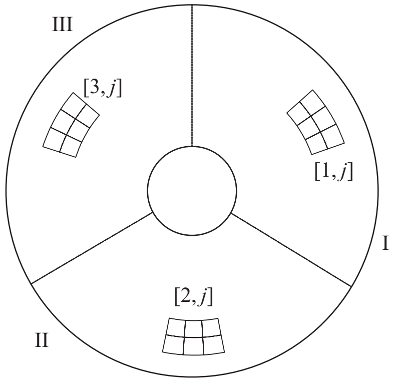


Fig. 9. Absolute location of block [1, $j$], [2, $j$] and [3, $j$].

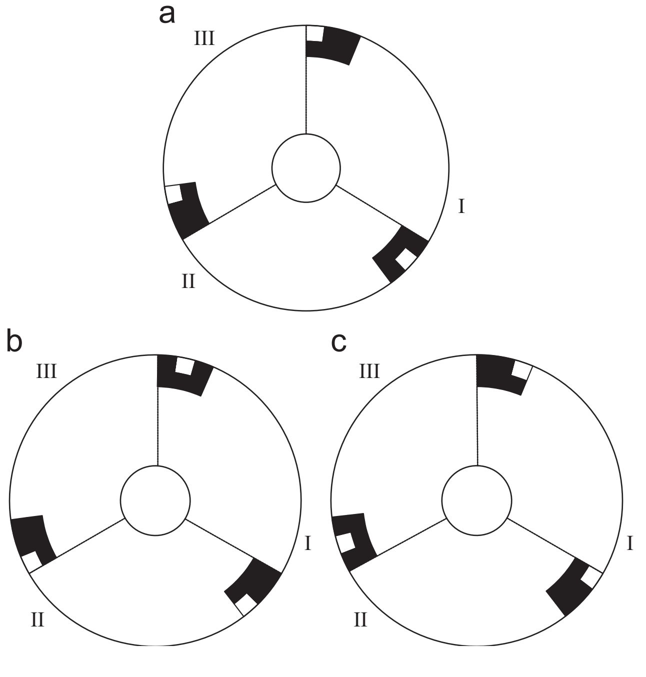


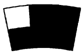
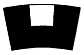
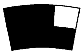
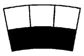
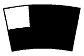
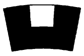
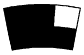
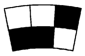
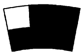
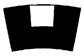
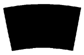
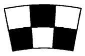
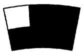
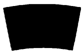
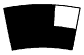
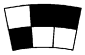
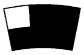
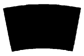
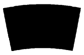
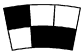
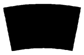
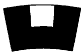
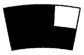
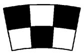
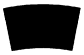
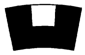
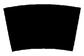
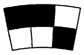
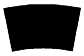
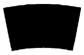
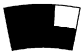
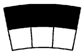
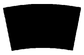
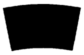
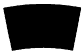
Fig. 8. Encoding the first three blocks in each of the three chords by $\Sigma_1$ in $A$: (a) $A$, (b) $A^{120°}$, (c) $A^{240°}$.

$(p_i)_j^1$ in the first chord of $A^{(i-1)\theta} \otimes B$ by our visual system for $1 \leqslant i \leqslant 3(=x)$ and $\theta = 120°(=360°/x)$.

We call the blocks in column $s_B$ of Table 5 the *elementary blocks circle share B* for sharing 3 secrets, which consists of three white and three black sub-pixels. They are named by $s_B^0, s_B^1, \ldots, s_B^7$ in sequence as indicated in Fig. 10. When we denote □ as 0 and ■ as 1, the superscript $l$ of $s_B^l$ is equal to the code formed by $p_1p_2p_3$ in binary, i.e. $l = btod(p_1p_2p_3)$ where $btod(b)$ is a function that returns the decimal representation of a binary number $b$. It means that based upon Table 5,once $(a_j^1, a_j^2, a_j^3)$ is assigned to be $(s_A^1, s_A^2, s_A^3)$ and $b_j^1$ is encoded to be $s_B^{btod(p_1p_2p_3)}$ (specifically, $(a_j^1, a_j^2, a_j^3) = (permute(s_A^1, \Sigma_j), permute(s_A^2, \Sigma_j), permute(s_A^3, \Sigma_j))$ and $b_j^1 = permute(s_B^{btod(p_1p_2p_3)}, \Sigma_j)$in practical implementation) with respect to the given $(p_1, p_2, p_3)_j^1$, $(a_j^1 \otimes b_j^1, a_j^2 \otimes b_j^1, a_j^3 \otimes b_j^1)$ recovers $(p_1, p_2, p_3)_j^1$.

Now, we take the instances in Fig. 11, in which the first three pixels of the three divided strips in $P_i$ are depicted for $1 \leqslant i \leqslant 3$, as an example to show how the

Table 5
Encoding a set of corresponding pixels $(p_1, p_2, p_3)_j^1$ into $a_j^1$ ($a_j^2$ and $a_j^3$) and $b_j^1$ in terms of $s_A^1$ ($s_A^2$, $s_A^3$, respectively) and $s_B$ in the first chords of $A$ and $B$, respectively for visual 3-secret sharing

| $p_1$ | $p_2$ | $p_3$ | $s_A^1$ | $s_A^2$ | $s_A^3$ | $s_B$ | $s_A^1 \otimes s_B$ | $s_A^2 \otimes s_B$ | $s_A^3 \otimes s_B$ |
|---|---|---|---|---|---|---|---|---|---|
| □ | □ | □ |  |  |  |  |  |  |  |
| □ | □ | ■ | | | |  |  |  |  |
| □ | ■ | □ | | | |  |  |  |  |
| □ | ■ | ■ | | | |  |  |  |  |
| ■ | □ | □ | | | |  |  |  |  |
| ■ | □ | ■ | | | |  |  |  |  |
| ■ | ■ | □ | | | |  |  |  |  |
| ■ | ■ | ■ | | | |  |  |  |  |

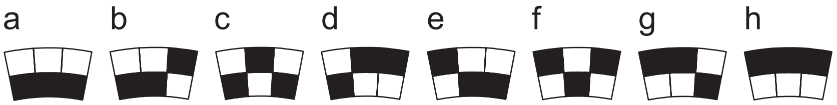


Fig. 10. Elementary blocks of share $B$ for sharing 3 secrets: (a) $s_B^0$, (b) $s_B^1$, (c) $s_B^2$, (d) $s_B^3$, (e) $s_B^4$, (f) $s_B^5$, (g) $s_B^6$, (h) $s_B^7$.

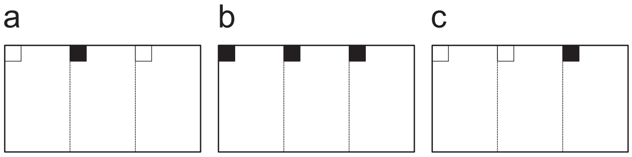


Fig. 11. Instances of the first three pixels of the three strips in (a) $P_1$, (b) $P_2$ and (c) $P_3$.

corresponding blocks in $B$ are encoded. From Fig. 11, we have $(p_1, p_2, p_3)_1^1 = (\square, ■, \square)$. According to Table 5, the elementary block for $b_1^1$ is chosen to be $s_B^{btod(p_1p_2p_3)} = s_B^{btod(010)} = s_B^2$. Since in practical implementation, $b_1^1$'s corresponding block $a_1^1$ ($a_1^2, a_1^3$) in $A$ ($A^{120°}$, $A^{240°}$, respectively) has been encoded as $permute(s_A^1, \Sigma_1)$ ($permute(s_A^2, \Sigma_1)$, $permute(s_A^3, \Sigma_1)$, respectively), the same permutation $\Sigma_1$ should be adopted for encoding $b_1^1$ to preserve the superimposition result of $s_A^1 \otimes s_B$ ($s_A^2 \otimes s_B$ and $s_A^2 \otimes s_B$, respectively). Let us simply set $\Sigma_1 = \{1, 2, 3, 4, 5, 6\}$. Therefore, $b_1^1$ is encoded as $permute(s_B^2, \Sigma_1)$ ( ) as shown in Fig. 12. It is easily seen that

$$a_1^1 \otimes b_1^1 = permute(s_A^1, \Sigma_1) \otimes permute(s_B^2, \Sigma_1)$$

$= s_A^1 \otimes s_B^2 =$ $\otimes$ $=$ ,

$$a_1^2 \otimes b_1^1 = permute(s_A^2, \Sigma_1) \otimes permute(s_B^2, \Sigma_1)$$

$= s_A^2 \otimes s_B^2 =$ $\otimes$ $=$ ,

$$a_1^3 \otimes b_1^1 = permute(s_A^3, \Sigma_1) \otimes permute(s_B^2, \Sigma_1)$$

$= s_A^3 \otimes s_B^2 =$ $\otimes$ $=$ .

Therefore, the first blocks in the first chords of $A \otimes B$, $A^{120°} \otimes B$ and $A^{240°} \otimes B$ reconstruct $(p_1)_1^1$ (□), $(p_2)_1^1$ (■) and $(p_3)_1^1$) (□), respectively.

In summary, given a certain $(p_1, p_2, p_3)_j^1$ in the first strips of $(P_1, P_2, P_3)$, $b_j^1$ is encoded as $permute(s_B^{btod(p_1p_2p_3)}, \Sigma_j)$ where $\Sigma_j$ is a random permutation for $1 \leqslant j \leqslant \beta$. It is noted that for a specific block $b_j^1$ in the first chord of $B$, when $A$ is rotated 0°, 120° and 240° counter-clockwise, the blocks that are superimposed onto $b_j^1$ are $a_j^1$, $a_j^2$and $a_j^3$, respectively where $a_j^k = permute(s_A^k, \Sigma_j)$ for $1 \leqslant j \leqslant \beta$ and $1 \leqslant k \leqslant 3$.

Now, consider a certain block $b_j^2$ in the second chord of $B$. When $A$ is rotated 0°, 120° and 240° counter-clockwise, the blocks that are superimposed onto $b_j^2$ are $a_j^2$, $a_j^3$ and $a_j^1$ accordingly (see Fig. 9 and formula (2)). Thus, to recover a given set of $(p_1, p_2, p_3)_j^2$, we should assure that $s_A^2 \otimes s_B$, $s_A^3 \otimes s_B$ and $s_A^1 \otimes s_B$ (or more precisely $permute(s_A^2,\Sigma_j) \otimes permute(s_B, \Sigma_j)$, $permute(s_A^3, \Sigma_j) \otimes permute(s_B, \Sigma_j)$ and $permute(s_A^1, \Sigma_j) \otimes permute(s_B, \Sigma_j)$) reconstruct $(p_1)_j^2$, $(p_2)_j^2$ and $(p_3)_j^2$, respectively. Table 6 is designed for this principle.

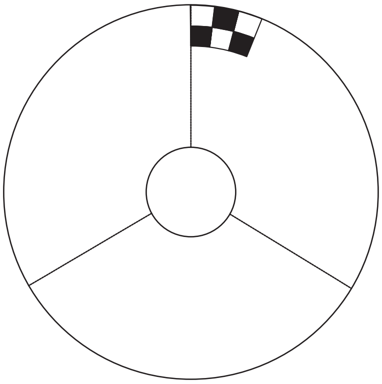


Fig. 12. Encoding $b_1^1$ in $B$.

In fact, we can re-arrange Table 6 to make the columns 4–10 exactly the same as those in Table 5. Table 7 is such a consequence. Note that Tables 5 and 7 are the same except for the headings of columns 1–3. From Table 7, we observe that given a set of corresponding pixels $(p_1, p_2, p_3)_j^2$, the elementary block of $b_j^2$ can be easily determined by $s_B^{btod(p_3p_1p_2)}$.

Following the above example shown in Fig. 11, we have $(p_1, p_2, p_3)_1^2 = (■, ■, □)$ and $\Sigma_1 = (1, 2, 3, 4, 5, 6)$). Since $btod(p_3p_1p_2) = btod(011) = 3$, $b_1^2$ is encoded as $permute(s_B^3, \Sigma_1)$ ( ) as shown in Fig. 13. It is easily seen that

$$a_1^2 \otimes b_1^2 = permute(s_A^2, \Sigma_1) \otimes permute(s_B^3, \Sigma_1)$$

$= s_A^2 \otimes s_B^3 =$ $\otimes$ $=$ ,

$$a_1^3 \otimes b_1^2 = permute(s_A^3, \Sigma_1) \otimes permute(s_B^3, \Sigma_1)$$

$= s_A^3 \otimes s_B^3 =$ $\otimes$ $=$ ,

$$a_1^1 \otimes b_1^2 = permute(s_A^1, \Sigma_1) \otimes permute(s_B^3, \Sigma_1)$$

$= s_A^1 \otimes s_B^3 =$ $\otimes$ $=$ .

That is, $a_1^2 \otimes b_1^2$, $a_1^3 \otimes b_1^2$ and $a_1^1 \otimes b_1^2$ (the first blocks in the second chords of $A \otimes B$, $A^{120°} \otimes B$ and$A^{240°} \otimes B$) reconstruct $(p_1)_1^2$ (■), $(p_2)_1^2$ (■) and $(p_3)_1^2$ (□), respectively.

Based upon the experience above, Table 8 summarizes the encoding scheme for the blocks in the third chord of $B$ for

Table 6
Encoding a set of corresponding pixels $(p_1, p_2, p_3)_j^2$ into $a_j^2$ ($a_j^3$ and $a_j^1$) and $b_j^2$ in terms of $s_A^2$ ($s_A^3$,$s_A^1$, respectively) and $s_B$ in the second chords of $A$ and $B$, respectively for visual 3-secret sharing

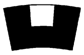
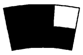
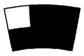
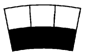
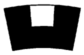
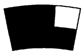
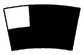
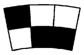
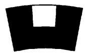
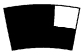
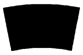
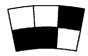
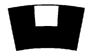
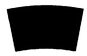
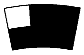
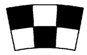
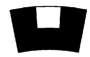
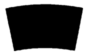
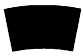
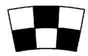
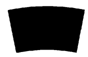
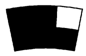
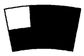
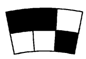
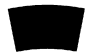
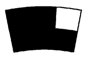
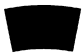
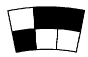
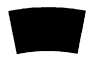
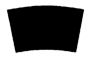
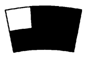
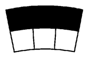
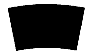
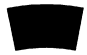
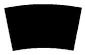
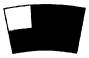
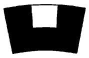
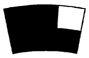
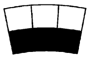
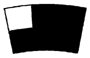
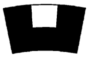
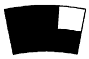
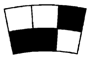
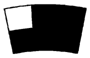
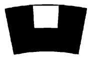
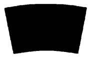
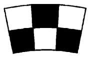
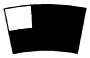
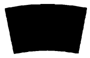
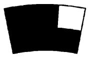
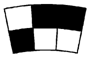
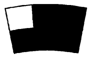
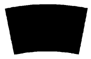
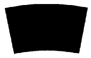
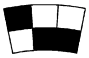
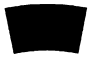
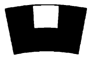
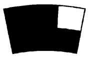
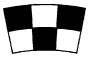
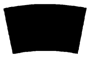
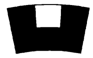
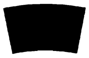
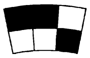
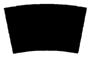
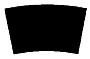
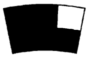
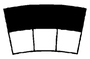
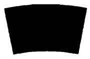
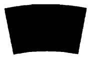
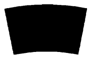
| $p_1$ | $p_2$ | $p_3$ | $s_A^2$ | $s_A^3$ | $s_A^1$ | $s_B$ | $s_A^2\otimes s_B$ | $s_A^3\otimes s_B$ | $s_A^1\otimes s_B$ |
|---|---|---|---|---|---|---|---|---|---|
| □ | □ | □ |  |  |  |  |  |  |  |
| □ | □ | ■ | | | |  |  |  |  |
| □ | ■ | □ | | | |  |  |  |  |
| □ | ■ | ■ | | | |  |  |  |  |
| ■ | □ | □ | | | |  |  |  |  |
| ■ | □ | ■ | | | |  |  |  |  |
| ■ | ■ | □ | | | |  |  |  |  |
| ■ | ■ | ■ | | | |  |  |  |  |

Table 7
Encoding scheme equivalent to Table 6 for the second chords of $A$ and $B$ for visual 3-secret sharing

| $p_3$ | $p_1$ | $p_2$ | $s_A^1$ | $s_A^2$ | $s_A^3$ | $s_B$ | $s_A^1 \otimes s_B$ | $s_A^2 \otimes s_B$ | $s_A^3 \otimes s_B$ |
|---|---|---|---|---|---|---|---|---|---|
| □ | □ | □ |  |  |  |  |  |  |  |
| □ | □ | ■ | | | |  |  |  |  |
| □ | ■ | □ | | | |  |  |  |  |
| □ | ■ | ■ | | | |  |  |  |  |
| ■ | □ | □ | | | |  |  |  |  |
| ■ | □ | ■ | | | |  |  |  |  |
| ■ | ■ | □ | | | |  |  |  |  |
| ■ | ■ | ■ | | | |  |  |  |  |

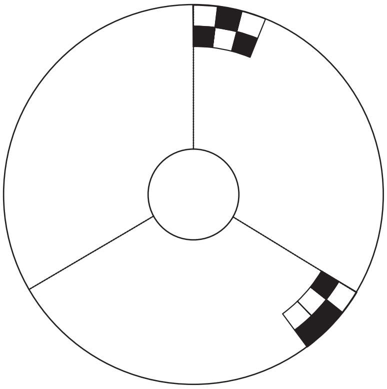


Fig. 13. Encoding $b_1^2$ in $B$.

sharing 3 secrets. We can see from Table 8 that given a set of corresponding pixels $(p_1, p_2, p_3)_j^3$, the elementary block of $b_j^3$ is chosen to be $s_B^{btod(p_2 p_3 p_1)}$.

Following the previous example in Fig. 11, consider the particular case $(p_1, p_2, p_3)_1^3 = $ (□, ■, ■). Since $btod(p_2 p_3 p_1) = btod(110) = 6$, we encode $b_1^3$ as $permute(s_B^6, \Sigma_1)$ (▀) as shown in Fig. 14. It is clearly seen that

$a_1^3 \otimes b_1^2 = permute(s_A^3, \Sigma_1) \otimes permute(s_B^6, \Sigma_1)$
$= s_A^3 \otimes s_B^6 =$ ▀ $\otimes$ ▀ $=$ ▀,
$a_1^1 \otimes b_1^2 = permute(s_A^1, \Sigma_1) \otimes permute(s_B^6, \Sigma_1)$
$= s_A^1 \otimes s_B^6 =$ ▀ $\otimes$ ▀ $=$ ▀,
$a_1^2 \otimes b_1^2 = permute(s_A^2, \Sigma_1) \otimes permute(s_B^6, \Sigma_1)$
$= s_A^2 \otimes s_B^6 =$ ▀ $\otimes$ ▀ $=$ ▀.

We have that $a_1^3 \otimes b_1^3$, $a_1^1 \otimes b_1^3$ and $a_1^2 \otimes b_1^3$ (the first blocks in the third chords of $A \otimes B$, $A^{120°} \otimes B$ and $A^{240°} \otimes B$) recover $(p_1)_1^3$ (□), $(p_2)_1^3$ (■), and $(p_3)_1^3$ (■), respectively.

Fig. 15 depicts the results of the first three blocks in the three chords of $A \otimes B$, $A^{120°} \otimes B$ and $A^{240°} \otimes B$ which reconstruct $(p_1)_1^k$ in $P_1$ (□, ■, □) (Fig. 15(a) vs. Fig. 11(a)), $(p_2)_1^k$ in $P_2$ (■, ■, ■)(Fig. 15(b) vs. Fig. 11(b)) and $(p_3)_1^k$ in $P_3$ (□, □, ■)) (Fig. 15(c) vs. Fig. 11(c)) respectively for $1 \leqslant k \leqslant 3$.

Based upon the above discussions, we obtain

$$b_j^k = permute(s_B^l, \Sigma_j)$$

where

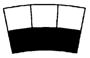
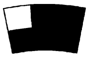
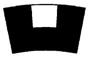
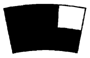
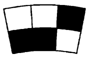
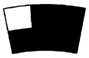
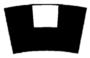
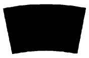
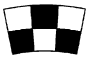
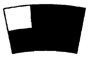
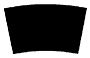
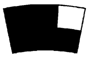
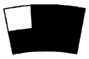
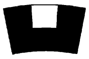
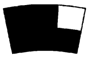
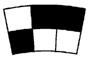
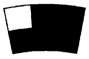
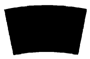
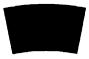
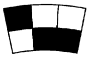
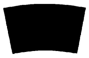
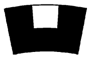
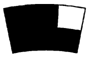
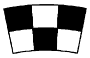
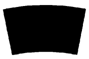
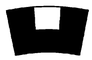
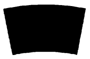
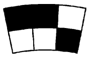
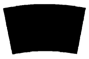
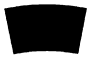
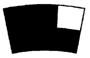
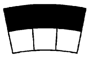
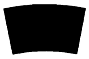
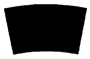
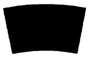
$$l = \begin{cases} btod(p_1 p_2 p_3) & \text{if } k = 1; \\ btod(p_3 p_1 p_2) & \text{if } k = 2; \\ btod(p_2 p_3 p_1) & \text{otherwise,} \end{cases}$$

for the given set of corresponding pixels $(p_1, p_2, p_3)_j^k$ with respect to pixel $j$ of strip $k$ in $(P_1, P_2, P_3)$ where $1 \leqslant j \leqslant \beta$ and $1 \leqslant k \leqslant 3$.

Let $rotate(r_1 r_2 \ldots r_x, d)$ denote a function that rotates $r_1 r_2 \ldots r_x$ right $d$ bits where $r_i \in \{0, 1\}$, $1 \leqslant i \leqslant x$ and $0 \leqslant d \leqslant x - 1$; that is,

$$rotate(r_1 r_2 \ldots r_x, d) = \begin{cases} r_1 r_2 \ldots r_x \\ \quad \text{if } d = 0; \\ r_{x-d+1} r_{x-d+2} \ldots r_x r_1 r_2 \ldots r_{x-d} \\ \quad \text{otherwise } (1 \leqslant d \leqslant x - 1). \end{cases} \qquad (3)$$

Table 8
Encoding a set of corresponding pixels $(p_1, p_2, p_3)_j^3$ into $a_j^3$ ($a_j^1$ and $a_j^2$) and $b_j^3$ in terms of $s_A^3$ ($s_A^1$,$s_A^2$, respectively) and $s_B$ in the third chords of $A$ and $B$, respectively for visual 3-secret sharing

| $p_2$ | $p_3$ | $p_1$ | $s_A^1$ | $s_A^2$ | $s_A^3$ | $s_B$ | $s_A^1 \otimes s_B$ | $s_A^2 \otimes s_B$ | $s_A^3 \otimes s_B$ |
|---|---|---|---|---|---|---|---|---|---|
| □ | □ | □ | | | |  |  |  |  |
| □ | □ | ■ | | | |  |  |  |  |
| □ | ■ | □ | | | |  |  |  |  |
| □ | ■ | ■ |  |  |  |  |  |  |  |
| ■ | □ | □ | | | |  |  |  |  |
| ■ | □ | ■ | | | |  |  |  |  |
| ■ | ■ | □ | | | |  |  |  |  |
| ■ | ■ | ■ | | | |  |  |  |  |

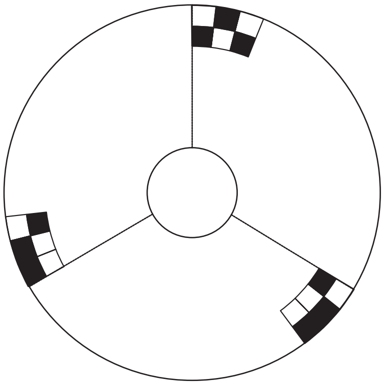


Fig. 14. Encoding $b_1^3$ in $B$.

Then the above formula can be simplified as

$$b_j^k = permute(s_B^{btod(rotate(p_1 p_2 p_3, k-1))}, \Sigma_j) \tag{4}$$

with respect to the set of corresponding pixels $(p_1, p_2, p_3)_j^k$ where $1 \leqslant j \leqslant \beta$ and $1 \leqslant k \leqslant 3$.

Since the number of sub-pixels in both of the elementary blocks of $A$ and $B$ is $6(=2x)$ in the above case, the *pixel expansion* (i.e. the number of sup-pixels in the shares needed to encode a set of corresponding pixels in the secret images) in our algorithm is 6 for $x=3$. The visual $x$-secret sharing scheme for any general number $x \geqslant 2$ will be formalized in the following section.

### 3.2. Proposed visual multi-secret sharing scheme

The definitions of the elementary blocks for circle shares $A$ (formula (1)) and $B$ (formula (4)) and the encoding scheme in Tables 5, 7 and 8 for visual 3-secret sharing can be generalized to accomplish the visual multi-secret sharing for $x \geqslant 1$ (including $x=1$) secrets. Furthermore, there is no need to store any codebook like Tables 5, 7 and 8. Thus our scheme formally presented in the following is not only general but also efficient for physical implementation.

Assume that there are $x$ secrets to be shared by two participants. The two circle shares $A$ and $B$ are evenly decomposed into $x$ chords, respectively. Let $\theta$ denote the degree expanded in each chord of $A$ and $B$. It is computed as

$$\theta = 360°/x.$$

We refer to the elementary block of the $x$ secrets as a block with $2x$ ordered sub-pixels as shown in Fig. 16. It is noted that the pixel expansion in our scheme is $2x$ when $x$ secrets are shared. The width and height of the elementary block can be any combination as long as their multiplication is $2x$ (or even any number larger than $2x$ for some special purposes, such as retaining aspect ratios, to ease the production of the circle Shares, and so on). The order of the $2x$ sub-pixels in the elementary block can also be arbitrarily defined. In the following discussions, we follow the shape and order of the elementary block as shown in Fig. 16.

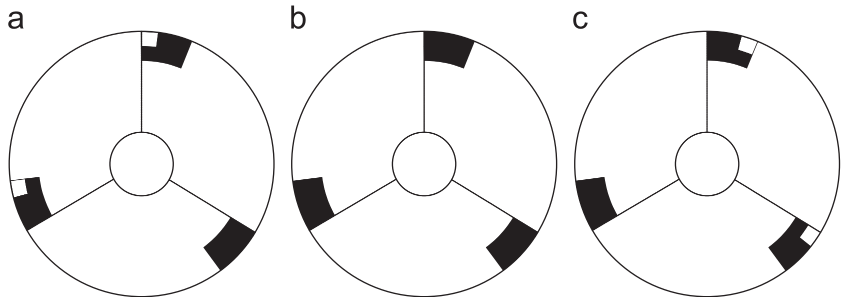


Fig. 15. Results of (a) $A \otimes B$, (b) $A^{120^\circ} \otimes B$ and (c) $A^{240^\circ} \otimes B$.

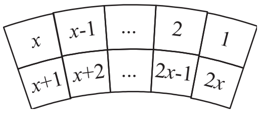


Fig. 16. Elementary block for $x$ secrets.

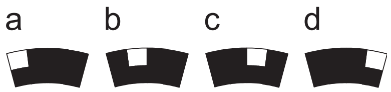


Fig. 17. Elementary blocks in $E_A^4$: (a) $s_A^1$, (b) $s_A^2$, (c) $s_A^3$, (d) $s_A^4$.

We define the set of the elementary blocks for share $A$ as follows:

$$E_A^x = \{s_A^k | 1 \leqslant k \leqslant x\},$$

where $s_A^k$ is an elementary block consisting of one white and $2x-1$ black sub-pixels in which the $j$th sub-pixel, denoted as $s_A^k[j]$, is defined by

$$s_A^k[j] = \begin{cases} 0 & \text{if } j = x+1-k; \\ 1 & \text{otherwise,} \end{cases} \tag{5}$$

for $1 \leqslant j \leqslant 2x$ and $1 \leqslant k \leqslant x$.

Fig. 17 shows the elementary blocks of $A$ for encoding $x=4$ secrets. As an example, we show how the sub-pixels in $s_A^2$ are computed by formula (5). Since $k=2$ and $x+1-k=4+1-2=3$, thus $s_A^k[3]=0$ and $s_A^k[j']=1$ for $1 \leqslant j' \neq 3 \leqslant 8(=2x)$ as shown in Fig. 17(b).

We define the set of the elementary blocks for share $B$ as follows:

$$E_B^x = \{s_B^\gamma | 0 \leqslant \gamma \leqslant 2^x - 1\},$$

where $s_B^\gamma$ is also an elementary block containing $x$ white and $x$ black sub-pixels in which the $j$th sub-pixel, denoted as $s_B^\gamma[j]$, is defined by

$$s_B^\gamma[j] = \begin{cases} r_j & 1 \leqslant j \leqslant x; \\ \bar{r}_{2x+1-j} & \text{otherwise,} \end{cases} \tag{6}$$

where $\gamma = btod(r_x r_{x-1} \ldots r_2 r_1)$, that is, $r_t$ is the $t$th least significant bit of $\gamma$ when $\gamma$ is represented in binary ($x$-bit) in which $1 \leqslant t \leqslant x$ and $0 \leqslant \gamma \leqslant 2^x - 1$, for $1 \leqslant j \leqslant 2x$ and $\bar{r}_t$ is the inverse of $r_t$.

Fig. 18 illustrates the elementary blocks of $B$ for $x=4$. Consider $s_B^4$. Since $\gamma = 4 = btod(r_4 r_3 r_2 r_1) = btod(0100)_2$, we have $(s_B^4[1], s_B^4[2], s_B^4[3], s_B^4[4]) = (r_1, r_2, r_3, r_4) = (0,0,1,0)$ and $(s_B^4[5], s_B^4[6], s_B^4[7], s_B^4[8]) = (\bar{r}_{2\times4+1-5}, \bar{r}_{2\times4+1-6}, \bar{r}_{2\times4+1-7}, \bar{r}_{2\times4+1-8}) = (\bar{r}_4, \bar{r}_3, \bar{r}_2, \bar{r}_1) = (1,0,1,1)$. Thus, $s_B^4$ is as shown in Fig. 18(e).

Formulae (1) and (4) about the encoding of blocks in $A$ and $B$, respectively, for $x=3$ can now be formulated in a more generalized form as follows. The blocks in $A$ are encoded by

$$(a_j^1, a_j^2, \ldots, a_j^x) = permute(s_A^1, \Sigma_j), permute(s_A^2, \Sigma_j), \ldots, permute(s_A^x, \Sigma_j)), \tag{7}$$

where $\Sigma_j$ is a random permutation of $\{1, 2, \ldots, 2x\}$ for $1 \leqslant j \leqslant \beta$.

Given a set of corresponding pixels $(p_1, p_2, \ldots, p_x)_j^k$ in block $j$ of strip $k$ in $(P_1, P_2, \ldots, P_x)$, $b_j^k$ (i.e. block $j$ of chord $k$ in $B$) is encoded by

$$b_j^k = permute(s_B^{btod(rotate(p_1 p_2 \ldots p_x, k-1))}, \Sigma_j) \tag{8}$$

for $1 \leqslant j \leqslant \beta$ and $1 \leqslant k \leqslant x$ where function $ratote(p_1 p_2 \ldots p_x, k-1)$ is defined by formula (3).

Based upon the above definitions and formulae (5)–(8), our visual multi-secrets sharing scheme is formally presented in Algorithm 1.

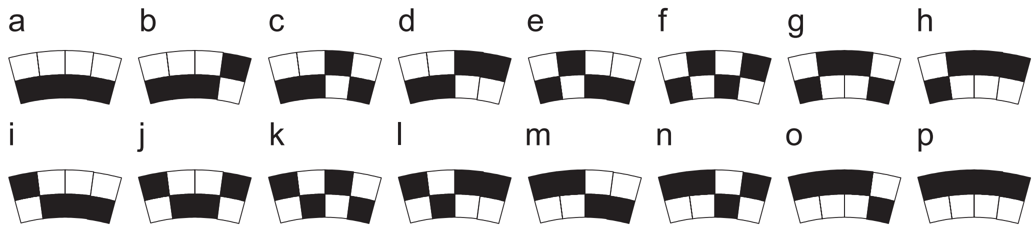


Fig. 18. Elementary blocks in $E_B^4$: (a) $s_B^0$, (b) $s_B^1$, (c) $s_B^2$, (d) $s_B^3$, (e) $s_B^4$, (e) $s_B^5$, (g) $s_B^6$, (h) $s_B^7$, (i) $s_B^8$, (j) $s_B^9$, (k) $s_B^{10}$, (l) $s_B^{11}$, (m) $s_B^{12}$, (n) $s_B^{13}$, (o) $s_B^{14}$, (p) $s_B^{15}$.

**Algorithm 1.** *Encoding x secret images into two circle shares*

```
Input:   $x$ $h \times w$ binary secret image $P_1, P_2, \ldots, P_x$
Output:  two circle shares $A$ and $B$ such that any single $A$ or $B$ leaks no information about any one of the secret images, while
         $A^{(i-1)\theta} \otimes B$ recovers $P_i$ for $1 \leqslant i \leqslant x$ in the human visual system where $\theta = 360°/x$ and $A^{0°} = A$
1.       Create $A$ and $B$ as circle shares which are decomposed into $x$ chords where each chord is composed by $\beta = h \times (w/x)$
         chord-shaped blocks referred to as $a_j^k$ and $b_j^k$, $1 \leqslant k \leqslant x$ and $1 \leqslant j \leqslant \beta$, respectively and each block contains $2x$ sub-pixels.
2.       Generate $E_A^x$ and $E_B^x$ according to formulae (5) and (6) respectively.
3.       for (each block $j$, $1 \leqslant j \leqslant \beta$) do
3.1      {      Determine $\Sigma_j = (\sigma_1, \sigma_2, \ldots, \sigma_{2x})$, a random permutation of $\{1, 2, \ldots, 2x\}$
3.2             for (each chord $k$, $1 \leqslant k \leqslant x$) do
                {      // all $\beta$ blocks in $x$ chords of $A$ and $B$ adopt the same permutation $\Sigma_j$
3.2.1                  $a_j^k = permute(s_A^k, \Sigma_j)$
3.2.2                  for (each secret image $i$, $1 \leqslant i \leqslant x$) do $q_i = (p_i)_j^k$
3.2.3                  $\gamma = btod(rotate(q_1 q_2 \ldots q_x, k-1))$
3.2.4                  $b_j^k = permute(s_B^\gamma, \Sigma_j)$
                }
         }
4.       Output($A$, $B$) // $A$ and $B$ are composed by all $a_j^k$'s and $b_j^k$'s respectively
```

Note that a new permutation $\Sigma_j$ is determined to permute the sub-pixels in each pair of $a_j^k$ and $b_j^k$ for $1 \leqslant j \leqslant \beta$ and $1 \leqslant k \leqslant x$ to ensure the entire randomness that the sub-pixels in $a_j^k$ and $b_j^k$ can provide. Further, $a_j^k$ and $b_j^k$ are encoded by using the same permutation $\Sigma_j$ so that the numbers of the white and black sub-pixels in $a_j^k \otimes b_j^k$ and $s_A^k \otimes s_B^\gamma$ are exactly the same for $1 \leqslant k \leqslant x$.

The pixel expansion of our scheme is $2x$ when $x$ secrets are shared. In the case of $x = 2$, our pixel expansion is $2x = 4$ which is the same as that of Wu and Chen [9] as well as Wu and Chang [10]. The number of all possible patterns in an extended block in $S_1$ [9] (see Fig. 2(c)) or any sector block in $A$ [10] (see Fig. 4(a)) is 4, which contains two white and two black sub-pixels, while that in $b_j^k$ of our scheme is $(4!)/(2! \times 2!) = 6$. The randomness of our scheme, in the case of $x = 2$, is surely better than that of Wu and Chen as well as Wu and Chang.

It is seen from Algorithm 1 that we do not physically store any information about Tables 5, 7 and 8 in memory. The elementary blocks $s_A^k$'s and $s_B^\gamma$'s are generated in the run time (Step 2 in Algorithm 1) according to formulae (5) and (6). The encoding process is guaranteed by formulae (7) and (8) (Step 3 in Algorithm 1).

## 4. Experimental results and discussions

We implemented the proposed visual multi-secret sharing scheme by using Borland C++ Builder 6 (BCB) in a personal computer running MS Windows. Since the blocks are in the shape of chords, we called the embedded functions in BCB such as circle drawing, line drawing, flood-filling a closed area, and so on, to build the chord-shaped blocks in our scheme.

We designed three experiments to explore the feasibility and applicability of our visual multi-secret sharing scheme. Experiment 1 verifies the correctness of our scheme for $x=3$ where the starting position for encoding on the circle shares are fixed as above-mentioned. Experiment 2 demonstrates that our scheme can be easily extended in such a way that the starting position for encoding can be arbitrarily assigned. This increases the secrecy of the proposed scheme. Experiment 3 gives the implementation results of our visual 4-secret sharing scheme. Then, we examine the pixel expansion and contrast of our scheme.

*Experiment* 1: Fig. 19 illustrates the results of a computer implementation of the proposed scheme for sharing three secret images. Figs. 19(a)–(c) are the three secrets to be shared,

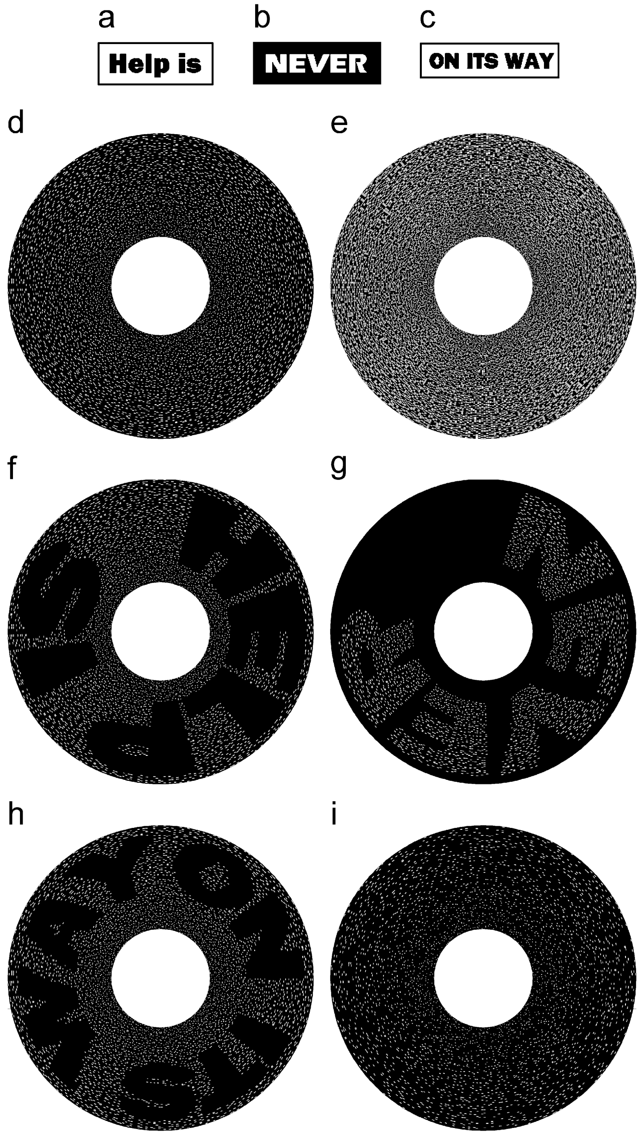


Fig. 19. Implementation results for the proposed visual 3-secret sharing scheme: (a) $P_1$, (b) $P_2$, (c) $P_3$, (d) $A$, (e) $B$, (f) $A \otimes B$, (g) $A^{120^\circ} \otimes B$, (h) $A^{240^\circ} \otimes B$, (i) $A^{85^\circ} \otimes B$.

namely $P_1$, $P_2$ and$P_3$, respectively. Figs. 19(d) and (e) show the circle shares $A$ and $B$ encoded by Algorithm 1, which expose no information about $P_1$, $P_2$ and $P_3$ individually. Figs. 19(f)–(h) reveal the superimposed results of $A \otimes B$, $A^{120^\circ} \otimes B$ and $A^{240^\circ} \otimes B$ which reconstruct $P_1$, $P_2$ and $P_3$ in our visual system, respectively. Fig. 19(i) gives another superimposed result, e.g. $A^{85^\circ} \otimes B$, which leaks no information about any of the three secrets. In fact, any result of $A^{\theta} \otimes B$, for $\theta \neq 0°, 120°, 240°$, is merely a seemingly random picture.

*Experiment* 2: The encoding processes of $A$ and $B$ in our algorithm start from the 0° position and move on in a clockwise direction (see Fig. 5). However, the starting position for encoding in $A$ (or $B$) can be pre-defined arbitrarily.

Fig. 20 shows the implementation results of using the same example as in Experiment 1 with a different starting position in $B$; that is, we encoded $B$ by starting from the 85° position (85° counter clock-wise to the 0° position) while we encoded $A$ by starting from the 0° position as mentioned. The three secret images are the same as those in Figs. 19(a)–(c). Figs. 20(a) and (b) are the circle shares $A'$ and $B'$ encoded by Algorithm 1. Fig. 20(c) shows the result of $A' \otimes B'$ which reveals nothing about the secrets, while Figs. 20(d)–(f) display the superimposed results of $(A')^{85^\circ} \otimes B'$, $(A')^{205^\circ} \otimes B'$, and $(A')^{325^\circ} \otimes B'$ which reconstruct $P_1$, $P_2$ and $P_3$, respectively, in our visual system. Note that both $A \otimes B$ (Fig. 19(f)) and $(A')^{85^\circ} \otimes B'$ (Fig. 20(c)) disclose $P_1$, yet, the reconstructed$P_1$ in $(A')^{85^\circ} \otimes B'$ is 85° counter-clockwise away from that in $A \otimes B$.

*Experiment* 3: Fig. 21 gives the implementation results of the proposed scheme for sharing four secrets. Figs. 21(a)–(d) are the four secrets to be shared, namely $P_1$, $P_2$, $P_3$ and $P_4$, respectively. Figs. 21(e) and (f) are the encoded circle shares $A$ and $B$. Figs. 21(g)–(j) show the superimposed results of $A \otimes B$, $A^{90^\circ} \otimes B$, $A^{180^\circ} \otimes B$ and $A^{270^\circ} \otimes B$ which recover $P_1$, $P_2$, $P_3$ and $P_4$ in our visual system, respectively.

The results in Experiments 1–3, as expected, demonstrate the feasibility and applicability of our visual multi-secret sharing scheme. When we deal with $x$ secrets, the pixel expansion of our scheme is $2x$. The contrast (i.e. the relative difference between the reconstructed white and black pixels in the superimposed image) of our scheme is $1/(2x)$ since all $2x$ subpixels in a reconstructed black pixel are black, while those in a reconstructed white pixel are $2x - 1$ black and 1 white. Note that when $x = 2$, the pixel expansions (contrasts) in Wu and Chen's [9], Wu and Chang's [10] and our schemes are all 4 (1/4).

As a matter of fact, the sharing of multiple secrets visually brings forth new problems to be considered. For instance, with regard to the "starting position for encoding" in $A$ or/and $B$ in Experiment 2, we may design such a concern to be some kind of *private key* which is only accessible between the dealer and authorized participant(s). Without the correct starting positions in $A$ or/and $B$, the alignment of $A$ and $B$ cannot recover the secret yet. In addition, the second secret of the three secrets in Experiment 1 might be designed to be fake for the purpose of diffusion. That is to say whether the whole secret message is "Help is never on its way" or "Help is on its way" may be treated to be another private key between the dealer and authorized participant(s). Mainly, the number of secrets, the degree of the starting position for encoding, the combination of the true or fake reconstructed secrets, and so on, can be designed as private keys to increase the level of security in the visual multi-secret sharing system.

## 5. Concluding remarks

By adopting circle shares, we designed a general visual secret sharing scheme for $x \geqslant 1$ (indeed, our scheme works well for $x = 1$) secrets in two shares in this paper. This is the first research that achieves true multi-secret sharing in two shares in visual cryptography. The previous studies considered sharing

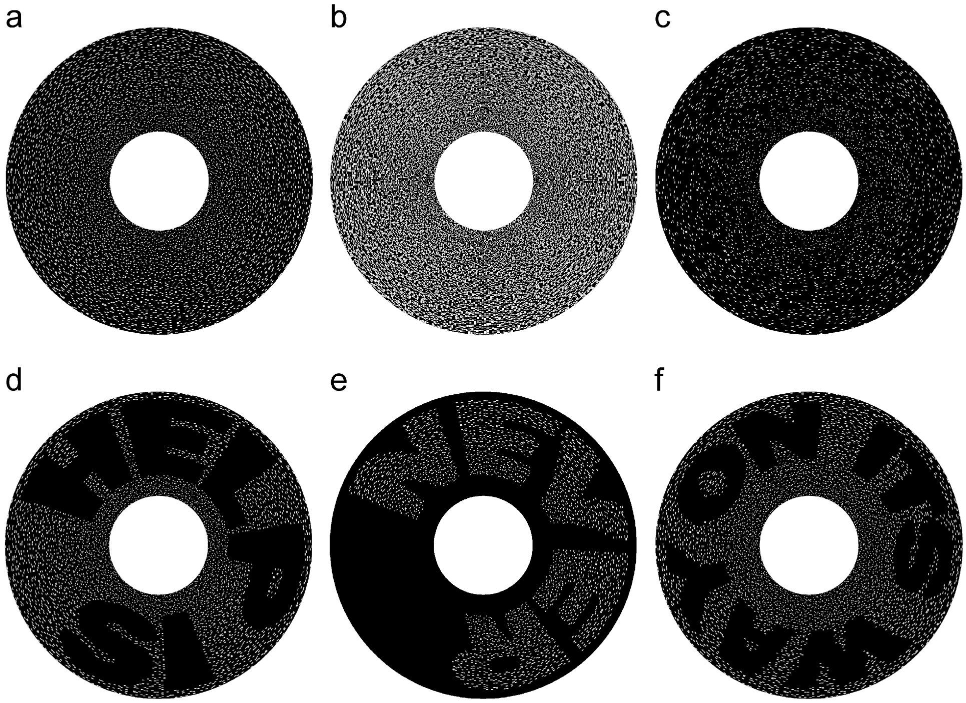


Fig. 20. Implementation results for the proposed visual 3-secret sharing scheme with a different starting encoding position: (a) $A'$, (b) $B'$, (c) $A' \otimes B'$, (d) $(A')^{85^\circ} \otimes B'$, (e) $(A')^{205^\circ} \otimes B'$, (f) $(A')^{325^\circ} \otimes B'$.

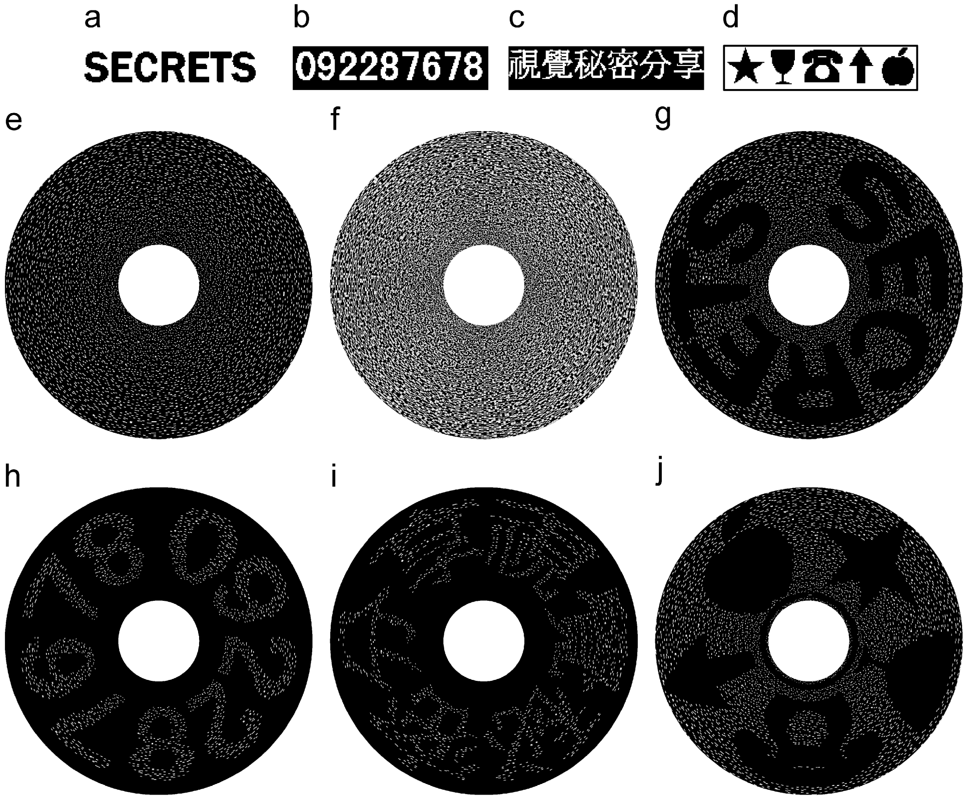


Fig. 21. Results of computer implementation for visual 4-secret sharing: (a) $P_1$, (b) $P_2$, (c) $P_3$, (d) $P_4$, (e) $A$, (f) $B$, (g) $A \otimes B$, (h) $A^{90^\circ} \otimes B$, (i) $A^{180^\circ} \otimes B$, (j) $A^{270^\circ} \otimes B$.

only two secrets in two shares [9,10]. The proposed scheme can be implemented easily and it takes only some constant working space. All encoding information can be determined in run time. By introducing an independent random permutation (i.e. $\Sigma_j$, see formulae (1) and (4)) when encoding each pair of the corresponding blocks (i.e. $a_j^k$ and $b_j^k$, see Step 3.2.1 and 3.2.4

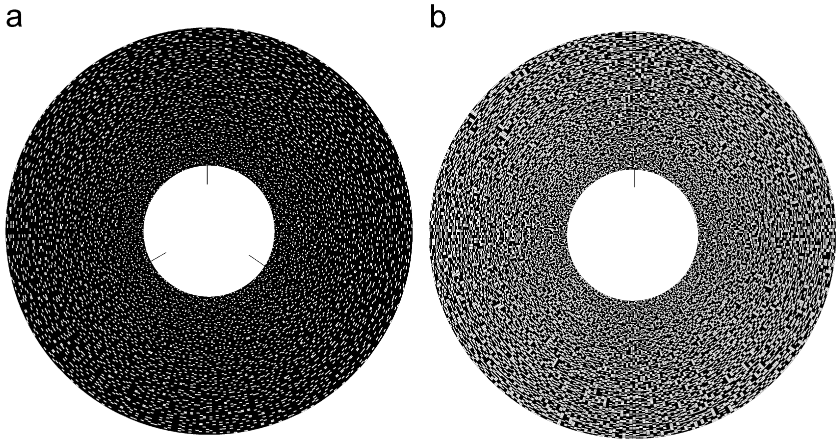


Fig. 22. Shares (based upon Experiment 1) with supplementary lines to ease the alignments: (a) $A$ with three markers, (b) $B$ with one marker.

in Algorithm 1), our scheme ensures the maximum randomness that the sub-pixels in an encoded block may possibly provide. For the transmitter, one machine capable of running our encoding scheme is needed, while for the receivers, no computing device is required and the decryption process is simply by human visual system. The proposed scheme can be easily extended to gray-level images by adopting the halftone technology [6] or even color images by exploiting color decomposition [6] or color composition [8].

In traditional visual secret sharing schemes, rectangle shares are encoded to conceal one shared secret. They are easily superimposed by aligning the rectangular corners. As compared to the rectangle shares, the circle shares in our visual multi-secret sharing system are relatively hard to superimpose since there are no reference points to align with. Basically, the usage of the circle shares increases the complexity in decoding the secrets. In practical applications, the dealer might add additional information in the circle shares, such as some supplementary points, lines or markers, to ease the superimposition (decoding process) for the participants. Fig. 22 shows one possible arrangement in the case of sharing three secrets as in Experiment 1. Note that circle share $A$ has three markers (see Fig. 22(a)), while $B$ has only one (see Fig. 22(a)) based upon which $A$, $A^{120^\circ}$ and $A^{240^\circ}$ can be superimposed with $B$ easily. Or, the dealer can deliberately organize such information as private key(s) such that only the legal receivers are informed how to obtain the key(s). Generally speaking, the use of circle shares to convey several secrets discloses some new issues that have not yet been considered in traditional visual cryptography, such as “How many secrets are there?”, “How to superimpose the shares (where to align with or in what rotation angles)?”, “Is there any fake secret(s) for diffusion?”, and so on. These concerns can be designed as a set of private keys. The consideration and distribution of these private keys can be further discussed.

The pixel expansion of our scheme is $2x$ when $x$ secrets are shared. It would be challenging to prove whether or not it is optimal. Is there any algorithm that improves the contrast in our scheme is surely worthy of further study. How to extend our scheme such that multiple secrets can be shared by more than two shares is also an interesting topic. Potentially, sharing multiple secrets may have more flexibilities and applications than sharing only one secret. Visual identification and visual authentication are some typical applications in visual cryptography [11]. It would be of much significance to re-examine these topics from a viewpoint of sharing multiple secrets.

## Acknowledgements

This research was partially supported by National Science Council of the Republic of China under grants NSC 94-2213-E-130-024 and NSC 93-2213-E-130-001.

## References

[1] M. Naor, A. Shamir, Visual cryptography, in: A. De Santis (Ed.), Advances in Cryptology: Eurpocrypt’94, Lecture Notes in Computer Science, vol. 950, 1995, pp. 1–12.

[2] G. Ateniese, C. Blundo, A. De Santis, D.R. Stinson, Visual cryptography for general access structures, Inf. Comput. 129 (1996) 86–106.

[3] D.R. Stinson, An introduction to visual cryptography, Presented at Public Key Solutions ’97, Toronto, Canada, April 28–30, 1997.

[4] E.R. Verheul, H.C.A. Van Tilborg, Constructions and properties of $k$ out of $n$ visual secret sharing schemes, Des. Codes Cryptogr. 11 (1997) 179–196.

[5] C. Blundo, A. De Santis, D.R. Stinson, On the contrast in visual cryptography schemes. J. Cryptology 12 (1999) 261–289.

[6] Y.-C. Hou, Visual cryptography for color images, Pattern Recognition 36 (2003) 1619–1629.

[7] C.-C. Lin, W.-H. Tsai, Visual cryptography for grey-level images by dithering techniques, Pattern Recognition Lett. 24 (2003) 349–358.

[8] S.J. Shyu, Efficient visual secret sharing scheme for color images, Pattern Recognition 39 (2006) 866–880.

[9] C.C. Wu, L.H. Chen, A study on visual cryptography, Master Thesis, Institute of Computer and Information Science, National Chiao Tung University, Taiwan, R.O.C., 1998.

[10] H.-C. Wu, C.-C. Chang, Sharing visual multi-secrets using circle shares, Comput. Stand. Interfaces 134 (28) (2005) 123–135.

[11] M. Naor, B. Pinkas, Visual authentication and identification, in: B.S. Kaliski, Jr. (Ed.), Advances in Cryptology: CRYPTO’97, Lecture Notes in Computer Science, vol. 1294, 1997, pp. 322–336.

**About the Author**—SHYONG JIAN SHYU received his B.S. degree in Computer Engineering from the Nationa Chiao Tung University in 1985, the M.S. degree in Computer and Decision Sciences in 1987 and the Ph.D. degree in Computer Science in 1991 from the National Tsing Hua University, Taiwan, ROC. From 1993 to 1994, he worked as a researcher at Academia Sinica Computer Centre, Taiwan. Currently, he is a professor of the Department of Computer Science and Information Engineering at Ming Chuan University, Taiwan. His research interests include the design and anlaysis of algorithms, parallel computing, visual cryptography and computational biology.

**About the Author**—SHIH-YU HUANG received his B.S. degree in information engineering from Tatung Institute of Technology, Taipei Taiwan, Republic of China, in 1988, and his M.S. and Ph.D. degrees from Department of Computer Sciences, National Tsing Hua University Taiwan, Republic of China, in 1990 and 1995, respectively. From 1995 to 1999, he worked in telecommunication laboratories of Chunghwa Telecom. Co., LTD., Taiwan. In October 1999, he joined the Department of Computer Science and Information Engineering, Ming Chuan University, Taiwan. His current interests are image compression, visual communication, and digital library.

**About the Author**—YEUAN-KUEN LEE received his B.S., M.S., and Ph.D. degrees in computer and information science from the National Chiao-Tung University, Hsinchu, Taiwan, in 1989, 1991, and 2002, respectively. From 1993 to 1995, he was a lecturer at the Aletheia University, Taipei, Taiwan. He is currently an assistant professor of computer science and information engineering at Ming-Chuan University, Taiwan. His research interests are in the areas of interactive media, media security, digital steganography, and steganalysis.

**About the Author**—RAN-ZAN WANG received his B.S. degree in computer engineering and science in 1994 and M.S. degree in electrical engineering and computer science in 1996, both from Yuan-Ze University. In 2001, he received his Ph.D. degree in Computer and Information Science from National Chiao Tung University. In 2001–2002, he was an assistant professor at the department of computer engineering in Van Nung University. He joined the department of computer and communication engineering at Ming Chuan University in August 2002, and is currently an associate professor there. His recent research interests include media security, pattern recognition, and image processing. Dr. Wang is a member of the Phi-Tau-Phi Scholastic Honor Society.

**About the Author**—KUN CHEN received his M.S. degree in computer science and information engineering in 2007. He is currently a Ph.D. student in the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan. His recent research interests include cryptography and object-oriented programming.