J. Vis. Commun. Image R. 21 (2010) 900–916

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Flip visual cryptography (FVC) with perfect security, conditionally-optimal contrast, and no expansion

Sian-Jheng Lin [a], Shang-Kuan Chen [b,*], Ja-Chen Lin [a]

[a] Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
[b] Department of Computer Science and Information Engineering, Yuanpei University, Hsinchu, Taiwan

ARTICLE INFO

Article history:
Received 31 March 2010
Accepted 17 August 2010
Available online 26 August 2010

Keywords:
Visual cryptography
Flipping
Conditionally optimal contrast
Secret image sharing
Visual secret sharing
Non-expansible
Perfect security
Random grids

ABSTRACT

This paper proposes a flip visual cryptography (FVC) scheme with perfect security, conditionally optimal contrast, and no expansion of size. The proposed FVC scheme encodes two secret images into two dual-purpose transparencies. Stacking the two transparencies can reveal one secret image. Flipping one of the two transparencies and then stacking with the other transparency can reveal the second secret image. The proposed scheme is proved to have conditionally optimal contrast: its contrast is optimal if the double-secrets non-expanded FVC scheme is required to have perfect security. The perfect security is also proved.

1. Introduction

To avoid sensitive digital contents from being peeped by unauthorized people is significant in the area of information security. Visual cryptography (VC) introduced by Naor and Shamir [2] is an approach to decrypt secret image using human visual system. In VC, the secret image can be revealed by stacking the transparencies generated in the encryption process. Since the decoding process of VC depends on the inspection of stacked images using naked eyes without any computation, it has the potential to be utilized in the critical environment that has no computer resources. We may use the simple example in Fig. 1 to describe VC. Fig. 1(a) shows the binary secret image. After using the encoding process proposed by Naor and Shamir [2], the two generated transparencies are as in (b) and (c) which are extremely noisy. Fig. 1(d) shows the result of stacking together the two transparencies (b) and (c).

Many studies related to VC were proposed. For example, [1,3–5] introduced multi-secret VC; [6–11] proposed non-expanded VC so that the created transparency could be compact; and some other VC schemes [12–15] enabled VC to have more applications. In the above, Wu and Chang [3] proposed a method to generate two circle transparencies for sharing two secret images. When rotating one transparency by a pre-specified angle and then stacking it with another transparency, the second secret image could be revealed. In their method, the size of each transparency was fourfold larger than that of each secret image. Fang and Lin [4] used two rectangular transparencies to share two secret images. In their method, besides revealing one secret image by stacking the two transparencies, shifting one of the transparencies and then stacking them again could also reveal another secret image. The size of each transparency was also fourfold that of each secret image. Shyu et al. [5] extended the multi-secret VC scheme of Wu and Chang [3] from single rotation to several rotations so that they could encode k ($k \geqslant 2$) images in two transparencies. Nevertheless, the transparencies were still $2k$ times the size of each input secret image.

To save the usage of the transparencies, reducing the size of the transparencies is also an approach for study. There are several non-expanded VC schemes. For example, Yang [6] introduced a probability-based method and Shyu [8] presented a random-grid-based method. In both methods, the size of each transparency is the same as that of secret image. Therefore, their methods are particularly suitable for the situation with storage restriction. However, in their methods, only one secret image is hidden when several transparencies are created.

For cryptography, the most important issue is security. Most single-secret VC schemes (for example, [2]) mentioned that no single transparency would leak out the pixel value of the input secret image. Restated, these schemes satisfied the security requirement. However, for multiple secret images, it was rarely discussed in the reported methods about the security issue on the relation of the pixels between the multiple input secret images. In the proposed

[*] Corresponding author.
E-mail address: skchen@mail.ypu.edu.tw (S.-K. Chen).

1047-3203/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2010.08.006

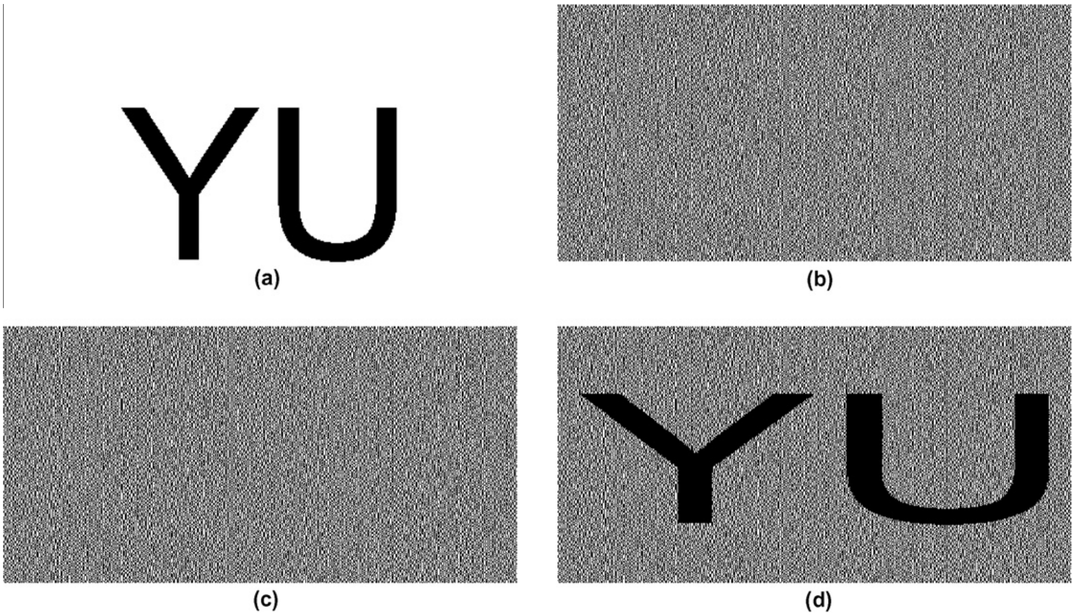

Fig. 1. An example of VC. (a): a secret image; (b and c): the two transparencies generated for (a) by using the VC scheme of Naor and Shamir [2]; (d) the result of stacking (b) and (c).

scheme, a multiple-secrets VC scheme with perfect security is defined as when each single transparency leaks out (a) neither the pixel value, (b) nor the relation of the pixel values between the multiple secret images. There are two possible branches of the design: (1) the stacking result representing black pixel in the secret image is restricted to be 100% opaque; (2) the stacking result representing black pixel in the secret image is not restricted to be 100% opaque. The first branch is called opaque-oriented FVC, and the second is called non-opaque-oriented FVC in the proposed scheme. We will prove later that the contrast in our design here is conditionally optimal, no matter (1) or (2) is used. Throughout this paper, the word "*conditionally optimal*" means that the contrast is optimal if the double-secrets non-expanded FVC scheme is required to have perfect security.

The remainder of this paper is organized as follows: the proposed opaque-oriented FVC scheme is stated in Section 2. Also in Section 2, we prove that the contrast 1/6 is conditionally optimal among the opaque-oriented FVC schemes that use basis matrices design with perfect security and no expansion. The proposed non-opaque-oriented FVC scheme and its proof are stated in Section 3. Experimental results are shown in Section 4. Finally, the conclusions are in Section 5.

2. Opaque-oriented FVC

In this section, we design an opaque-oriented FVC method. This section includes two subsections: (1) the encoding method; (2) the proof of conditionally optimal contrast.

2.1. The encoding method

Two $n \times m$ binary secret images, denoted by S_1 and S_2, are encoded to get two $n \times m$ transparencies T_1 and T_2, respectively. Without the loss of generality, the goal of the proposed FVC scheme is that the secret image S_1 can be decoded by stacking T_1 and T_2 together; whereas the secret image S_2 can be decoded by flipping T_1 over and then stacking with T_2. Fig. 2 illustrates the operation to flip a transparency over. Notably, the transparency in Fig. 2(a) is not a transparency created by our method, because our transparency is completely noise-like. Fig. 2 is just to explain the flip-over operation; and the explanation would have been impossible to understand if Fig. 2(a), and hence Fig. 2(b), had been completely noise-like.

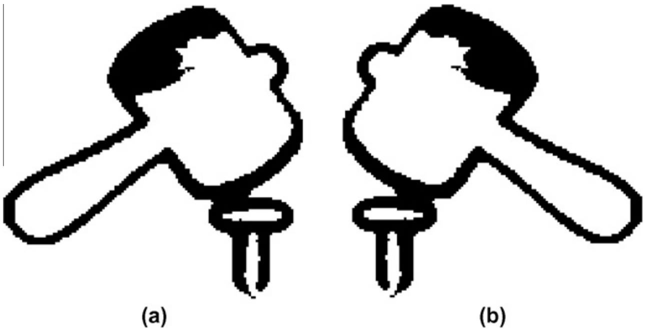

Fig. 2. (a): A transparency; (b): the transparency after flipping.

Let $S_1 = \{s_1(i,j)|0 \leqslant i \leqslant n-1, 0 \leqslant j \leqslant m-1\}$ and $S_2 = \{s_2(i,j)|0 \leqslant i \leqslant n-1, 0 \leqslant j \leqslant m-1\}$ be the two given black-and-white secret images. Each pixel $s_1(i,j)$ and each pixel $s_2(i,j)$ are binary in value W (white) pixel or B (black) pixel). Let $T_1 = \{t_1(i,j)|0 \leqslant i \leqslant n-1, 0 \leqslant j \leqslant m-1\}$ and $T_2 = \{t_2(i,j)|0 \leqslant i \leqslant n-1, 0 \leqslant j \leqslant m-1\}$ be the two transparencies to be generated. In the design of transparencies T_1 and T_2, represent every "*opaque*" pixel of a transparency by 1, and represent every "*transparent*" pixel of a transparency by 0. (To distinguish between secret image and transparency image, the words "*opaque* and *transparent*", rather than "Black and White", are used when the image being talked about is a transparency, rather than an input secret image.) In Definition 1, the stacking operation is symbolized by the symbol "⊗" which is in fact the OR operator. This coincides with the real world experience: in real world, if we stack two transparencies, the places where we can see through are the places where both transparencies are transparent (both are 0s).

Definition 1. Stacking operation $\otimes$The stacking operation for transparencies is symbolized by "⊗", where $0 \otimes 0 = 0$, $0 \otimes 1 = 1$, $1 \otimes 0 = 1$, and $1 \otimes 1 = 1$.

Fig. 3 illustrates the effect of stacking two transparencies T_1 and T_2 and describes what will happen when people flip T_1 over and

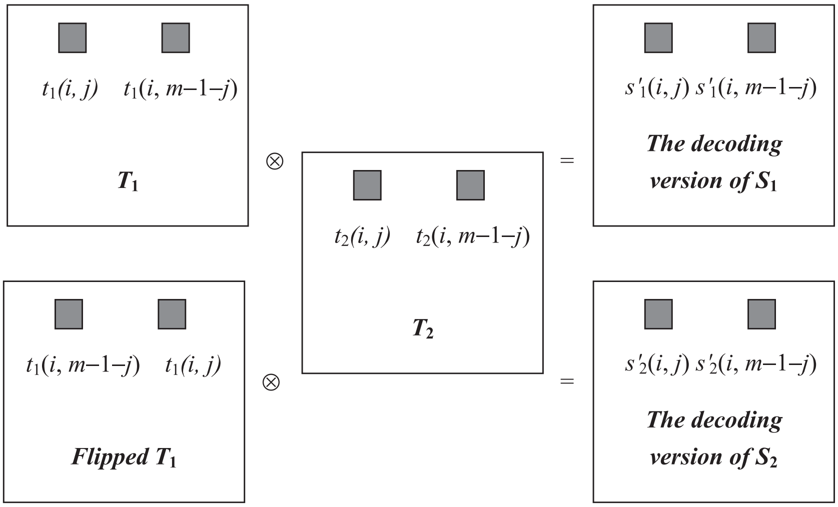
$t_1(i, j)$ $t_1(i, m-1-j)$
T_1
⊗
$t_2(i, j)$ $t_2(i, m-1-j)$
T_2
=
$s'_1(i, j)$ $s'_1(i, m-1-j)$
The decoding version of S_1
$t_1(i, m-1-j)$ $t_1(i, j)$
Flipped T_1
⊗
=
$s'_2(i, j)$ $s'_2(i, m-1-j)$
The decoding version of S_2

Fig. 3. Stacking transparencies T_1 and T_2 to decode secrets S_1 and S_2 of size $n \times m$ each. (Stacking T_1 and T_2 to decode secret S_1; flipping T_1 over and then stacking with T_2 to decode secret S_2.)

then stack it with T_2. The two pixel values $[s_1(i,j),\ s_1(i,m-1-j)]$ are called a symmetric pair, and so are $[s_2(i,j),\ s_2(i,m-1-j)]$. To design a flip visual cryptography (FVC) scheme, possible values of the quadruple $[s_1(i,j),\ s_1(i,m-1-j),\ s_2(i,j),\ s_2(i,m-1-j)]$ for $0 \leqslant i \leqslant n-1$, and $0 \leqslant j \leqslant m/2-1$ should be considered simultaneously. For each quadruple $[s_1(i,j),\ s_1(i,m-1-j),\ s_2(i,j),\ s_2(i,m-1-j)]$ of secret pixels, the quadruple $[t_1(i,j),\ t_1(i,m-1-j),\ t_2(i,j),\ t_2(i,m-1-j)]$ of transparency pixels must meet the following four requirements simultaneously:

(1) $s_1(i,j)$ is decoded by stacking $t_1(i,j)$ and $t_2(i,j)$;
(2) $s_1(i,m-1-j)$ is decoded by stacking $t_1(i,m-1-j)$ and $t_2(i,m-1-j)$;
(3) $s_2(i,j)$ is decoded by stacking $t_1(i,m-1-j)$ and $t_2(i,j)$;
(4) $s_2(i,m-1-j)$ is decoded by stacking $t_1(i,j)$ and $t_2(i,m-1-j)$;

With the use of the symbol ⊗, the four requirements read:

$$\begin{aligned}
&s'_1(i,j) = t_1(i,j) \otimes t_2(i,j);\\
&s'_1(i,m-1-j) = t_1(i,m-1-j) \otimes t_2(i,m-1-j);\\
&s'_2(i,j) = t_1(i,m-1-j) \otimes t_2(i,j);\\
&s'_2(i,m-1-j) = t_1(i,j) \otimes t_2(i,m-1-j).
\end{aligned} \tag{1}$$

Here, $[s'_1(i,j), s'_1(i,m-1-j), s'_2(i,j), s'_2(i,m-1-j)]$ are the *stacking results* to show the quadruple $[s_1(i,j),\ s_1(i,m-1-j),\ s_2(i,j),\ s_2(i,m-1-j)]$. Since we are dealing with visual decoding, the stacking results $[s'_1(i,j), s'_1(i,m-1-j), s'_2(i,j), s'_2(i,m-1-j)]$ do not need to be completely identical to the original secret values $[s_1(i,j),\ s_1(i,m-1-j),\ s_2(i,j),\ s_2(i,m-1-j)]$. Therefore, a prime symbol has been added to s to denote the stacking result.

Definition 2. The 16 basis matrices of a Flip VC (FVC) system are defined according to Fig. 3. In detail, each FVC system is defined according to its $2^4 = 16$ basis matrices $\{C_{WWWW}, C_{WWWB}, C_{WWBW}, \ldots, C_{BBBW}, C_{BBBB}\}$ of 4-by-r each, and r is a constant. All 4-by-r elements of each basis matrix $C_{[s_1(i,j), s_1(i,m-1-j), s_2(i,j),\ s_2(i,m-1-j)]} \in \{C_{WWWW}, C_{WWWB}, C_{WWBW}, \ldots, C_{BBBW}, C_{BBBB}\}$ are 1-bit in value. Notably, $s_1(i,j) \in \{W,B\}$, and so are the values of $s_1(i,m-1-j)$, $s_2(i,j)$, and $s_2(i,m-1-j)$. Hence, there are $2^4 = 16$ basis matrices to cover the 16 possible readings $\{WWWW, WWWB, \ldots, BBBB\}$ of the 4-dimensional input vector $[s_1(i,j), s_1(i,m-1-j), s_2(i,j), s_2(i,m-1-j)]$.

In the definition above, we stated that each FVC system is defined according to its $2^4 = 16$ basis matrices. This is because people can use the 16 basis matrices to encode any two secret images S_1 and S_2 to get two transparencies. In general, to encode four secret pixels $[s_1(i,j),\ s_1(i,m-1-j),\ s_2(i,j),\ s_2(i,m-1-j)]$ grabbed from secret images S_1 and S_2, just choose randomly a column from the corresponding basis matrix $C_{[s_1(i,j), s_1(i,m-1-j), s_2(i,j),\ s_2(i,m-1-j)]}$, then copy the four elements of the chosen column to the four transparency pixels $t_1(i,j)$, $t_1(i,m-1-$ it j), $t_2(i,j)$, $t_2(i,m-1-j)$ of T_1 and T_2, respectively.

To make sure the generated transparencies are secure and useful in unveiling the input secret images, the 16 basis matrices must satisfy the following Security and Contrast constraints. If these two constraints are satisfied, then the FVC defined by these 16 basis matrices is called a *valid* FVC.

I. Security constraint: In each 4-by-r basis matrix, the first and the second rows together consist of $a_0 \times r$ columns of $[0\,0]^T$, $a_1 \times r$ columns of $[0\,1]^T$, $a_2 \times r$ columns of $[1\,0]^T$, $a_3 \times r$ columns of $[1\,1]^T$, where

$$a_0 + a_1 + a_2 + a_3 = 1. \tag{2}$$

The value of a_0 used by any two basis matrices must be identical. (This cross-matrices requirement also holds for a_1, a_2 and a_3, respectively.) Likewise, the 3rd and the 4th rows together consist of $b_0 \times r$ columns of $[0\,0]^T$, $b_1 \times r$ columns of $[0\,1]^T$, $b_2 \times r$ columns of $[1\,0]^T$, and $b_3 \times r$ columns of $[1\,1]^T$, where

$$b_0 + b_1 + b_2 + b_3 = 1. \tag{3}$$

The value of b_0 used by any two basis matrices must be identical. (This cross-matrices requirement also exists for b_1, b_2 and b_3.)

II. Contrast constraint: Get the contrast according to the contrast evaluation process stated below. The Contrast constraint requires that the obtained value α must be positive.

Contrast evaluation: The contrast of a Flip VC is evaluated in the following manner. For each basis matrix, items 1–4 are evaluated below:

1. the *luminance transmission* of $s_1(i,j)$, which is the percentage of 0s in the stacking result when the 1st and 3rd rows are stacked;
2. the *luminance transmission* of $s_1(i,m-1-j)$, which is the percentage of 0s in the stacking result when the 2nd and 4th rows are stacked;
3. the *luminance transmission* of $s_2(i,j)$, which is the percentage of 0s in the stacking result when the 2nd and 3rd rows are stacked;

4. the *luminance transmission* of $s_2(i,m-1-j)$, which is the percentage of 0s in the stacking result when the 1st and 4th rows are stacked.

Then, since each of the four pixels $s_1(i,j)$, $s_1(i,m-1-j)$, $s_2(i,j)$, and $s_2(i,m-1-j)$ only have two possible values $\{W,B\}$, there are $2^4 = 16$ basis matrices (see Table 1, for example). These 16 matrices are distinguished from each other using a quadruple naming system. For example, if $[s_1(i,j),\ s_1(i,m-1-j),\ s_2(i,j),\ s_2(i,m-1-j)]$ is $[B,W,B,B]$, then the corresponding basis matrix is called C_{BWBB}. Now, for each of these 16 matrices, measure its luminance transmission of $s_1(i,j)$. If the first subscript in the matrix name is B, i.e. if $s_1(i,j) = B$, then store its luminance transmission of $s_1(i,j)$ in a pool called Black-pool. Otherwise, store it in a so-called White-pool. (Therefore, 16/2 = 8 of the 16 luminance transmission of $s_1(i,j)$ will be in Black-pool, and the remaining 16–8 = 8 will be in

Table 1
The 16 basis matrices corresponding to the $2^4 = 16$ combinations of $[s_1(i,j),\ s_1(i,m-1-j),\ s_2(i,j),\ s_2(i,m-l-j)]$, respectively. Some basis matrices (C_{WWWB}, C_{WWBW}, C_{WBWW}, and C_{BWWW}) have two forms, but only one form is needed in encoding. The user has freedom to choose the form he wants.

Reading of the input quadruple secret pixels $s_1(i,j)$, $s_1(i,m-l-j)$, $s_2(i,j)$, $s_2(i,m-l-j)$	The basis matrix corresponding to the input quadruple secret pixels
W, *W*, *W*, *W*	$C_{WWWW} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 0&1&1&1&0&1\\ 0&1&1&1&1&0 \end{bmatrix}$
W, *W*, *W*, *B*	$C_{WWWB} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 0&1&1&1&1&0\\ 1&1&0&1&1&0 \end{bmatrix}$, or $\begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&0&0&1&1&0\\ 1&1&0&1&1&0 \end{bmatrix}$
W, *W*, *B*, *W*	$C_{WWBW} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&0&1&1&0&1\\ 0&1&1&1&1&0 \end{bmatrix}$, or $\begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&0&1&1&0&1\\ 1&0&0&1&1&1 \end{bmatrix}$
W, *W*, *B*, *B*	$C_{WWBB} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&0&1&1&0&1\\ 1&1&0&1&1&0 \end{bmatrix}$
W, *B*, *W*, *W*	$C_{WBWW} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 0&1&1&1&1&0\\ 1&0&1&1&1&0 \end{bmatrix}$, or $\begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&0&0&1&0&1\\ 1&0&1&1&1&0 \end{bmatrix}$
W, *B*, *W*, *B*	$C_{WBWB} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 0&1&1&1&0&1\\ 1&1&1&1&0&0 \end{bmatrix}$
W, *B*, *B*, *W*	$C_{WBBW} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&0&1&0&1&1\\ 1&0&1&1&0&1 \end{bmatrix}$
W, *B*, *B*, *B*	$C_{WBBB} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&0&1&1&0&1\\ 1&1&1&0&1&0 \end{bmatrix}$
B, *W*, *W*, *W*	$C_{BWWW} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&1&0&0&1&1\\ 0&1&1&0&1&1 \end{bmatrix}$, or $\begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&1&0&0&1&1\\ 1&0&0&1&1&1 \end{bmatrix}$
B, *W*, *W*, *B*	$C_{BWWB} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&1&0&1&0&1\\ 1&1&0&1&1&0 \end{bmatrix}$
B, *W*, *B*, *W*	$C_{BWBW} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&1&1&1&0&0\\ 0&1&1&1&1&0 \end{bmatrix}$
B, *W*, *B*, *B*	$C_{BWBB} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&1&1&1&0&0\\ 1&1&0&1&1&0 \end{bmatrix}$
B, *B*, *W*, *W*	$C_{BBWW} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&1&0&0&1&1\\ 1&0&1&0&1&1 \end{bmatrix}$
B, *B*, *W*, *B*	$C_{BBWB} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&1&0&0&1&1\\ 1&1&1&0&1&0 \end{bmatrix}$
B, *B*, *B*, *W*	$C_{BBBW} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&1&1&0&0&1\\ 1&0&1&0&1&1 \end{bmatrix}$
B, *B*, *B*, *B*	$C_{BBBB} = \begin{bmatrix} 0&0&1&1&1&1\\ 0&1&0&1&1&1\\ 1&1&1&0&0&1\\ 1&1&1&0&1&0 \end{bmatrix}$

White-pool.) After that, for each of the 16 basis matrices, measure its luminance transmission of $s_1(i, m-1-j)$. If the second subscript in the matrix name is B, then store its luminance transmission of $s_1(i, m-1-j)$ in a pool called Black-pool. Otherwise, store it in a so-called White-pool. Repeat this process analogously for the 16 luminance transmissions of $s_2(i,j)$ according to the 3rd subscript of the matrices' names. Also repeat this process analogously for the 16 luminance transmissions of $s_2(i, m-1-j)$ according to the 4th subscript of the matrices' names. Together, we have $8+8+8+8=32$ numbers in the Black-pool, and $8+8+8+8=32$ numbers in the White-pool. The minimum of the 32 numbers in White-pool is called w (the minimal *luminance transmission* to represent W), and the maximum of the 32 numbers in Black-pool is called b (the maximal *luminance transmission* to represent B).

Define contrast α as

$$\alpha = w - b > 0. \tag{4}$$

Remark. In all VC methods, the stacking result is always with a contrast value smaller than 100–0% = 100% = 1, and this makes the stacking result always looks less clear than the input secret image (for example, compare Fig. 1(a) and (d)). In general, contrast is an important measure specifying the visual quality of the stacking result for a VC method. Roughly speaking, a decoded result with higher contrast is usually clearer.

Theorem 1. *When a FVC defined by 16 basis matrices satisfy the Security and the Contrast constraint addressed in* Definition 2 *, then the generated transparencies are secure and useful in unveiling the input secret images.*

Proof

(i) About the *Security constraint*, its purpose is that: no information about the two secret images can be extracted if someone only gets a transparency. Below we prove the security of the two secret images when someone only obtains transparency T_1. (The proof is likewise if *T*textsubscript1 is replaced by transparency *T*textsubscript2.)

The definition of Security constraint reads that "The value of a_0 used by any two basis matrices must be identical. (This cross-matrices requirement also holds for a_1, a_2, a_3, b_0, b_1, b_2, and b_3, respectively.)" Hence, if a set of basis matrices do not satisfy the Security constraint, then the value of a_0 (or a_1, or a_2, or a_3, orb_0, or b_1, or b_2, or b_3) used by some basis matrices may be different. For example, if the matrix C_{BBBB} in Table 1 is replaced by

$$C'_{BBBB} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix},$$

then the first and the second rows in C'_{BBBB} are with $[0\,0]^T \times 1$, $[0\,1]^T \times 1$, $[1\,0]^T \times 0$, and $[1\,1]^T \times 4$, while the first and second rows in remaining $16-1=15$ matrices of Table 1 are with $[0\,0]^T \times 1$, $[0\,1]^T \times 1$, $[1\,0]^T \times 1$, and $[1\,1]^T \times 3$. Since the first and second rows are used to encode $t_1(i,j)$ and $t_1(i, m-1-j)$ in the same transparency T_1, so if an intruder finds in T_1 a pair of pixels $[t_1(i,j), t_1(i,m-1-j)] = [1,1]$, then his best guess of the four corresponding secret pixels in secret images S_1 and S_2 should be [*BBBB*]. Likewise, if he finds in T_1 a pair of pixels $[t_1(i,j), t_1(i,m-1-j)] = [1,0]$, then he knows that the four corresponding secret pixels cannot be [*BBBB*]. In summary, the transparency T_1 is not a secure transparency, because it has secret-leaking problem.

The paragraph above shows the necessity of the Security constraint (to ensure that no information about the secret images can be extracted). Below we show the sufficiency of the Security constraint. Assume a set of 16 basis matrices satisfies the Security constraint. Therefore, in each 4-by-r basis matrix, the first and the second rows together consist of $a_0 \times r$ columns of $[0\,0]^T$, $a_1 \times r$ columns of $[0\,1]^T$, $a_2 \times r$ columns of $[1\,0]^T$, $a_3 \times r$ columns of $[1\,1]^T$, where the value of a_0 used by any two basis matrices must be identical. (This cross-matrices requirement also holds for a_1, a_2 and a_3, respectively.)

Since the first and second rows are utilized to encode $t_1(i,j)$ and $t_1(i,m-1-j)$ in the same transparency T_1, so if an intruder gets a single transparency T_1 and he finds in T_1 a pair of pixels $[t_1(i,j), t_1(i,m-1-j)] = [0,0]$, then he cannot know whether the four corresponding secret pixels in secret images S_1 and S_2 should be [*WWWW*] or [*WWWB*] or ... or [*BBBB*]. This is because each of the $2^4 = 16$ basis matrices has the same number of columns ($a_0 \times r$ columns) read as $[0\,0]^T$ when the first two rows of the matrix is grabbed. Therefore, there are 1/16 chance that $[0\,0]^T$ was from secret pixels [*WWWW*]. Similarly, there are 1/16 chance that $[0\,0]^T$ was from secret pixels [*WWWB*]. Similarly, there are 1/16 chance that $[0\,0]^T$ was from secret pixels [*WWBW*]. In fact, the same 1/16 chance holds for each of the 16 basis matrices.

Therefore, the intruder cannot know whether the four corresponding secret pixels in secret images S_1 and S_2 should be [*WWWW*] or [*WWWB*] or ... or [*BBBB*]. The above analysis still holds if $[0\,0]^T$ is replaced by $[0\,1]^T$ or $[1\,0]^T$ or $[1\,1]^T$. Therefore, no matter what the contents of two secret images S_1 and S_2 are, the transparency T_1 is always of perfect security: no secret-leaking will occur. The perfect security of transparency T_2 can be proved likewise using the 3rd and 4th rows of the 16 basis matrices, as defined in the second half of the Security constraint.

(ii) About the *Contrast constraint*, the definition is in Eq. (4) which reads $\alpha = w - b > 0$. If the value of α is not positive, then there are two possible cases:

Case 1. ($\alpha = 0$). In this case, we cannot see the information in the stacking result, because the luminance transmission of representing W and B are identical.

Case 2. ($\alpha < 0$). In this case, the luminance transmission to represent W is smaller than the luminance transmission to represent B. Then we will see that W is darker than B, and the stacking result will look like the negative film of a photo, an inappropriate view. □

Property 1. *The set of basis matrices shown in* Table 1 *is a valid FVC and it satisfies the security and the contrast of stacking result is 1/6.*

Proof. Table 1 shows a set of 16 basis matrices mentioned below Definition 2. In the 1st and 2nd rows of *each* basis matrix shown in Table 1, there are $(1/6) \times 6 = 1$ column of $[0\,0]^T$, $(1/6) \times 6 = 1$ column of $[0\,1]^T$, $(1/6) \times 6 = 1$ column of $[1\,0]^T$, and $(3/6) \times 6 = 3$ columns of $[1\,1]^T$. Hence, the cross-matrices constant-ratio (a_0: a_1: a_2: a_3) requirement mentioned below Eq. (2) holds. In the 3rd and 4th rows of each basis matrix, the cross-matrices constant-ratio (b_0: b_1: b_2: b_3) requirement mentioned below Eq. (3) also holds. The cross-matrices property required by the Security constraint is thus satisfied. Moreover, after the computation stated below, it can be shown that $w = 1/6$ and $b = 0$, so the contrast α is 1/6–0 = 1/6 which is a positive number, and hence the Contrast constraint is also satisfied by the Flip VC defined using Table 1.

The detail computation of w and b for Table 1 is as follows. First, statements 1–4 below are true for each basis matrix in Table 1. Therefore, every element of the Black-pool is 0, and each element of the White-pool is 1/6. Because the maximum element of the Black-pool (i.e. b) is 0 and the minimum element of the White-pool (i.e. w) is 1/6, contrast α is thus 1/6–0 = 1/6.

1. When the 1st and 3rd rows are stacked, if the first subscript in the matrix name is B, then the ratio of 0s in the stacking result is 0%; otherwise, the ratio is 1/6 = 16.7%.
2. When the 2nd and 4th rows are stacked, if the second subscript in the matrix name is B, then the ratio of 0s in the stacking result is 0%; otherwise, the ratio is 1/6 = 16.7%.
3. When the 2nd and 3rd rows are stacked, if the 3rd subscript in the matrix name is B, then the ratio of 0s in the stacking result is 0%; otherwise, the ratio is 1/6 = 16.7%.
4. When the 1st and 4th rows are stacked, if the forth subscript in the matrix name is B, then the ratio of 0s in the stacking result is 0%; otherwise, the ratio is 1/6 = 16.7%. □

We explain below in more detail what the two ratios 0% and 16.7% stand for. According to Fig. 3, each of the four secret pixels in [$s_1(i,j)$, $s_1(i,m-1-j)$, $s_2(i,j)$, $s_2(i,m-1-j)$] is recovered by tracing its two arrows in Fig. 3 back to two of the four transparency pixels in [$t_1(i,j)$, $t_1(i,m-1-j)$, $t_2(i,j)$, $t_2(i,m-1-j)$]. For example, the recovered version of secret pixel $s_1(i,j)$ is obtained by $s_1'(i,j) = t_1(i,j) \otimes t_2(i,j)$; whereas the recovered version of secret pixel $s_2(i,j)$ is obtained by $s_2'(i,j) = t_1(i,m-1-j) \otimes t_2(i,j)$. As for the encoding to generate the two transparencies t_1 and t_2, note that each 4-by-6 basis matrix in Table 1 has 6 columns; so, in the encoding process, each time an input quadruple [$s_1(i,j)$, $s_1(i,m-1-j)$, $s_2(i,j)$, $s_2(i,m-1-j)$] is given, there are 6 possible ways to encode this quadruple. For example, if the input secret quadruple [$s_1(i,j)$, $s_1(i,m-1-j)$, $s_2(i,j)$, $s_2(i,m-1-j)$] is [W,W,B,B], then [$t_1(i,j)$, $t_1(i,m-1-j)$, $t_2(i,j)$, $t_2(i,m-1-j)$] is encoded as [1,0,1,0] if the 3rd column of the basis matrix C_{WWWB} in Table 1 is selected. Likewise, [$t_1(i,j)$, $t_1(i,m-1-j)$, $t_2(i,j)$, $t_2(i,m-1-j)$] is encoded as [1,1,0,0] if the 6th column of matrix C_{WWWB} is selected in the random-selection process. Notably, the index $WWBB$ means that the input quadruple secret pixels are $s_1(i,j) = 0$, $s_1(i,m-1-j) = 0$, $s_2(i,j) = 1$, $s_2(i,m-1-j) = 1$. Now, no matter which of the 6 columns of matrix C_{WWBB} is selected, the value $s_2'(i,m-1-j) = t_1(i,j) \otimes t_2(i,m-1-j)$ obtained by stacking is always 1, because (1st row) ⊗ (4th row) = [1,1,1,1,1,1] for matrix $CWWWB$ of Table 1, and so is $s_2'(i,j)$. However, the value $s_1'(i,j) = t_1(i,j) \otimes t_2(i,j)$ obtained by stacking is not always 0, because (1st row) ⊗ (3rd row) = [1,0,1,1,1,1] for that matrix C_{WWBB}. In other words, depending on which of the 6 columns is selected, the chance that $t_1(i,j) \otimes t_2(i,j) = 0$ is only 1/6 = 16.7%. Similar argument also shows that the chance that $t_1(i,m-1-j) \otimes t_2(i,m-1-j) = 0$ is only 1/6 = 16.7%, too. Moreover, for each of the 16 basis matrices in Table 1, the probability that the stacking result can recover a black secret pixel (i.e. a secret pixel with value 1) is always 100%; but the probability that the stacking result can recover a white secret pixel (i.e. a secret pixel with value 0) is always 1/6 = 16.7%, rather than 100%. As a result, the black area of the input secret images is still black after stacking the two transparencies; however, since the 6 columns of each basis matrix in Table 1 is randomly selected, the white area of the input secret images looks gray (rather than plain white). This is because in each white area, the area is formed of many pixels, and after stacking the two transparencies, 16.7% of these pixels are white while 83.3% of these pixels are black. From the view of human vision (recalling that the decoder is human eyes rather than computers), since 83.3% of the pixels in a white area is black (opaque) and 16.7% of the pixels in the same white area is white (transparent), the whole white area looks like dark-gray in brightness, rather than plain white. Therefore, the white area of the original input image looks brighter than the corresponding area of the stacked output. Notably, darker output in white area is a very common phenomenon for any VC approach. For example, in Fig. 1, which shows the stacking result of the VC method proposed by Naor and Shamir [2], the input image's white area also becomes darker after VC's encoding-then-stacking.

2.2. The proof of conditionally optimal contrast in opaque-oriented FVC

In this subsection, the contrast in opaque-oriented FVC, which is no more than 1/6, is proven. To satisfy the Security constraint, the constant-ratios (a_0: a_1: a_2: a_3) and (b_0: b_1: b_2: b_3) in basis matrices C[$s_1(i,j)$, $s_1(i,m-1-j)$, $s_2(i,j)$, $s_2(i,m-1-j)$] must meet Eqs. (2) and (3). Moreover, in the first and second rows of each basis matrix, the occurrence of $[0\,0]^T$, $[0\,1]^T$, $[1\,0]^T$, $[1\,1]^T$ must keep the constant-ratio (a_0: a_1: a_2: a_3); and in the 3rd and the 4th rows of each basis matrix, the occurrence of $[0\,0]^T$, $[0\,1]^T$, $[1\,0]^T$, $[1\,1]^T$ must keep the constant-ratio (b_0: b_1: b_2: b_3). For each basis matrix, $c_{u,v} \geqslant 0$, where u = 0,1,2,3 and v = 0,1,2,3, is defined as the percentage of column $[\lfloor u/2\rfloor \quad u \bmod 2 \quad \lfloor v/2\rfloor \quad v \bmod 2]^T$ which appears in columns of the basis matrix.

By the Security constraint, we knows

$$a_u = c_{u,0} + c_{u,1} + c_{u,2} + c_{u,3} \quad \text{for } u = 0,1,2,3; \tag{5}$$

$$b_v = c_{0,v} + c_{1,v} + c_{2,v} + c_{3,v} \quad \text{for } v = 0,1,2,3. \tag{6}$$

By the definition of *luminance transmission*, the four stacking results $s_1'(i,j), s_1'(i,m-1-j), s_2'(i,j), s_2'(i,m-1-j)$ are represented by stacking two specific rows in basis matrix, which consists of 16 possible columns $[\lfloor u/2\rfloor \quad u \bmod 2 \quad \lfloor v/2\rfloor \quad v \bmod 2]^T$, where u, $v \in$ {0,1, 2,3}. Therefore, the *luminance transmission*of each stacking result $s_1'(i,j), s_1'(i,m-1-j), s_2'(i,j), s_2'(i,m-1-j)$ can be represented by the sum of a subset {$c_{u,v}$} which satisfies the result of stacking two specific rows defined in Definition 2.

1. The *luminance transmission* of stacking result $s_1(i,j)$ is

$$\sum_{u=0}^{3}\sum_{v=0}^{3} c_{u,v} \times \overline{(\lfloor u/2\rfloor \otimes \lfloor v/2\rfloor)} = c_{0,0} + c_{0,1} + c_{1,0} + c_{1,1}; \tag{7}$$

2. The *luminance transmission* of stacking result $s_1(i,m-1-j)$ is

$$\sum_{u=0}^{3}\sum_{v=0}^{3} c_{u,v} \times \overline{(\lfloor u \bmod 2\rfloor \otimes \lfloor v \bmod 2\rfloor)} = c_{0,0} + c_{0,2} + c_{2,0} + c_{2,2}; \tag{8}$$

3. The *luminance transmission* of stacking result $s_2(i,j)$ is

$$\sum_{u=0}^{3}\sum_{v=0}^{3} c_{u,v} \times \overline{(\lfloor u \bmod 2\rfloor \otimes \lfloor v/2\rfloor)} = c_{0,0} + c_{0,1} + c_{2,0} + c_{2,1}; \tag{9}$$

4. The *luminance transmission* of stacking result $s_2(i,m-1-j)$ is

$$\sum_{u=0}^{3}\sum_{v=0}^{3} c_{u,v} \times \overline{(\lfloor u/2\rfloor \otimes \lfloor v \bmod 2\rfloor)} = c_{0,0} + c_{0,2} + c_{1,0} + c_{1,2}; \tag{10}$$

where $\bar{\ }$ is the complement operator. The contrast α satisfies the Eq. (4). Notably, the *luminance transmission* of representing B is 0 and representing W is the contrast α by the definition of opaque-oriented FVC. Therefore, the complement operator is used in Eqs. (7)–(10) for opposite definition between B(1)/W(0) pixels and *luminance transmission*. Some basis matrices are considered below to gain the upper bound of contrast α.

I. Consider the basis matrix C_{BBBB}. By Eqs. (7)–(10), the *luminance transmission* of the four secret pixels $s_1(i,j)$, $s_1(i, m-1-j)$, $s_2(i,j)$, $s_2(i,m-1-j)$ are

$$\begin{aligned}&c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1}=0,\\&c_{0,0}+c_{0,2}+c_{2,0}+c_{2,2}=0,\\&c_{0,0}+c_{0,1}+c_{2,0}+c_{2,1}=0,\quad \text{and}\\&c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2}=0.\end{aligned}$$

Due to $c_{u,v} \geqslant 0$, for all u, v. Therefore, $c_{0,0} = c_{0,1} = c_{0,2} = c_{1,0} = c_{1,1} = c_{1,2} = c_{2,0} = c_{2,1} = c_{2,2} = 0$. By Eq. (5),

$$\begin{aligned}&a_0=c_{0,0}+c_{0,1}+c_{0,2}+c_{0,3}=c_{0,3},\\&a_1=c_{1,0}+c_{1,1}+c_{1,2}+c_{1,3}=c_{1,3},\quad \text{and}\\&a_2=c_{2,0}+c_{2,1}+c_{2,2}+c_{2,3}=c_{2,3}.\end{aligned}$$

By Eq. (6),

$$\begin{aligned}&b_0=c_{0,0}+c_{1,0}+c_{2,0}+c_{3,0}=c_{3,0},\\&b_1=c_{0,1}+c_{1,1}+c_{2,1}+c_{3,1}=c_{3,1},\quad \text{and}\\&b_2=c_{0,2}+c_{1,2}+c_{2,2}+c_{3,2}=c_{3,2}.\end{aligned}$$

By Eq. (5), $a_3 = c_{3,0} + c_{3,1} + c_{3,2} + c_{3,3} \geqslant c_{3,0} + c_{3,1} + c_{3,2} = b_0 + b_1 + b_2$. Therefore,

$$\begin{aligned}&a_3\geqslant b_0+b_1+b_2 \Rightarrow 1-a_0-a_1-a_2\geqslant b_0+b_1+b_2\\&\quad\Rightarrow (a_0+a_1+a_2)+(b_0+b_1+b_2)\leqslant 1\\&\quad\Rightarrow (1-a_3)+(1-b_3)\leqslant 1 \Rightarrow a_3+b_3\geqslant 1 \qquad (11)\end{aligned}$$

II. Consider the basis matrix C_{BWBB}. By Eqs. (7)–(10), the *luminance transmission* of the four secret pixels $s_1(i,j)$, $s_1(i,m-1-j)$, $s_2(i,j)$, $s_2(i,m-1-j)$ are

$$\begin{aligned}&c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1}=0,\\&c_{0,0}+c_{0,2}+c_{2,0}+c_{2,2}=\alpha,\\&c_{0,0}+c_{0,1}+c_{2,0}+c_{2,1}=0,\quad \text{and}\\&c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2}=0.\end{aligned}$$

Therefore, $c_{0,0} = c_{0,1} = c_{0,2} = c_{1,0} = c_{1,1} = c_{1,2} = c_{2,0} = c_{2,1} = 0$, and $c_{2,2} = \alpha$. By Eqs. (5) and (6), $a_2 = c_{2,0} + c_{2,1} + c_{2,2} + c_{2,3} \geqslant c_{2,2}$, and $b_2 = c_{0,2} + c_{1,2} + c_{2,2} + c_{3,2} \geqslant c_{2,2}$, so

$$\alpha \leqslant a_2 \quad \text{and} \quad \alpha \leqslant b_2. \qquad (12)$$

III. Consider the basis matrix C_{WBBW} and C_{WBWB}. When the basis matrices is C_{WBBW}, by Eqs. (7)–(10), the *luminance transmission* of the four secret pixels $s_1(i,j)$, $s_1(i,m-1-j)$, $s_2(i,j)$, $s_2(i,m-1-j)$ are

$$\begin{aligned}&c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1}=\alpha,\\&c_{0,0}+c_{0,2}+c_{2,0}+c_{2,2}=0,\\&c_{0,0}+c_{0,1}+c_{2,0}+c_{2,1}=0,\quad \text{and}\\&c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2}=\alpha.\end{aligned}$$

Therefore, $c_{0,0} = c_{0,1} = c_{0,2} = c_{2,0} = c_{2,1} = c_{2,2} = 0$, and

$$\begin{aligned}&\alpha=c_{1,0}+c_{1,1}=c_{1,0}+c_{1,2}\Rightarrow \alpha=(c_{1,0}+c_{1,1}+c_{1,0}+c_{1,2})/2\\&\quad=[(c_{1,0}+c_{1,1}+c_{1,2})+c_{1,0}]/2.\end{aligned}$$

Because, by Eq. (5),

$$\begin{aligned}&a_1=c_{1,0}+c_{1,1}+c_{1,2}+c_{1,3}\geqslant c_{1,0}+c_{1,1}+c_{1,2},\quad \text{and by Eq.(6)},\\&b_0=c_{0,0}+c_{1,0}+c_{2,0}+c_{3,0}\geqslant c_{1,0},\quad \text{so the contrast}\\&\alpha=[(c_{1,0}+c_{1,1}+c_{1,2})+c_{1,0}]/2\leqslant (a_1+b_0)/2. \qquad (13)\end{aligned}$$

When stacking result is C_{WBWB}, by Eq. (6), the *average luminance transmission* of the four secret pixels $s_1(i,j)$, $s_1(i,m-1-j)$, $s_2(i,j)$, $s_2(i,m-1-j)$ are

$$\begin{aligned}&c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1}=\alpha,\\&c_{0,0}+c_{0,2}+c_{2,0}+c_{2,2}=0,\\&c_{0,0}+c_{0,1}+c_{2,0}+c_{2,1}=\alpha,\\&c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2}=0.\end{aligned}$$

Therefore, $c_{0,0} = c_{0,2} = c_{1,0} = c_{1,2} = c_{2,0} = c_{2,2} = 0$, and

$$\begin{aligned}&\alpha=c_{0,1}+c_{1,1}=c_{0,1}+c_{2,1}\Rightarrow \alpha=(c_{0,1}+c_{1,1}+c_{0,1}+c_{2,1})/2\\&\quad=[(c_{0,1}+c_{1,1}+c_{2,1})+c_{0,1}]/2.\end{aligned}$$

Because, by Eq. (5),

$$a_0=c_{0,0}+c_{0,1}+c_{0,2}+c_{0,3}\geqslant c_{0,1}\quad \text{and by Eq.(6)},$$

$$b_1=c_{0,1}+c_{1,1}+c_{2,1}+c_{3,1}\geqslant c_{0,1},\quad \text{so the contrast}$$

$$\alpha=[(c_{0,1}+c_{1,1}+c_{2,1})+c_{0,1}]/2\leqslant (b_1+a_0)/2. \qquad (14)$$

By Eqs. (13) and (14), $\alpha \leqslant (a_1 + b_0)/2$,and $\alpha \leqslant (b_1 + a_0)/2$, we have $2\alpha \leqslant (a_1 + b_0)/2 + (b_1 + a_0)/2$, so

$$\begin{aligned}&\alpha\leqslant (b_1+a_0+a_1+b_0)/4. \qquad (15)\\&\alpha\leqslant (b_1+a_0+a_1+b_0)/4\quad (\text{By Eq.}(15))\\&=[1-(a_2+a_3+b_2+b_3)]/4\quad (\text{By Eq.}(2)\quad \text{and Eq.}(3))\\&\leqslant [1-(a_2+b_2)]/4\quad (a_3\geqslant 0, b_3\geqslant 0)\\&\leqslant (1-2\alpha)/4\quad (\text{By Eq.}(12))\\&\Rightarrow \alpha\leqslant 1/6.\end{aligned}$$

Therefore, the contrast of opaque-oriented FVC is no more than 1/6 if perfect security is required. The result also means that the encoding matrices shown in Table 1 are the optimal solution.

3. Non-opaque-oriented FVC

In the description and proof above for opaque-oriented FVC in Section 2, the conditionally optimal contrast is 1/6. In this section, we design a non-opaque-oriented FVC method; and the conditionally optimal contrast is 1/4. This section includes two subsections: (1) the encoding method; (2) the proof of conditionally optimal contrast.

3.1. The encoding method

In the encoding method of this section, we use 16 basis matrices of 8 columns each (rather than 6 columns) to encode the quadruple secret pixels $[s_1(i,j), s_1(i, m-1-j), s_2(i,j), s_2(i,m-1-j)]$. Each generated transparency will be of perfect security by using the 16 basis matrices to encode. The security and contrast are addressed below.

Property 2. *The set of basis matrices shown in Table 2 is a valid FVC and it satisfies the security and the contrast of stacking result is 1/4.*

Proof. Table 2 shows a set of 16 basis matrices mentioned in Definition 2. In rows 1 and 2 of *each* basis matrix shown in Table 2, there are $(2/8) \times 8 = 2$ column of $[0\ 0]^T$, $(2/8) \times 8 = 2$ column of $[0\ 1]^T$, $(2/8) \times 8 = 2$ column of $[1\ 0]^T$, and $(2/8) \times 8 = 2$ columns of $[1\ 1]^T$. Hence, the cross-matrices constant-ratio (a_0: a_1: a_2: a_3) requirement mentioned below Eq. (2) holds. In rows 3 and 4 of each basis matrix, the cross-matrices constant-ratio (b_0: b_1: b_2: b_3) requirement mentioned below Eq. (3) also holds. The cross-matrices property required by the Security constraint is thus satisfied. Moreover, after the computation stated below, it can be shown that $w = 3/8$ and $b = 1/8$, so the contrast α is $3/8-1/8 = 1/4$ which is a positive number, and hence the Contrast constraint is also satisfied by the FVC defined using Table 2.

Table 2
Encoding matrices of all combinations of $[s_1(i,j), s_1(i,m-1-j), s_2(i,j), s_2(i,m-1-j)]$.

Reading of the input quadruple secret pixels$[s_1(i,j), s_1(i,m-1-j), s_2(i,j), s_2(i,m-1-j)]$	The basis matrix corresponding to the input quadruple secret pixels
(W, W, W, W)	$C_{WWWW}\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&0&1&1&0&1&1\\0&0&0&1&0&1&1&1\\0&0&1&0&1&0&1&1\end{bmatrix}$
(W, W, W, B)	$C_{WWWB}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&0&0&1&0&1&1&1\\0&1&1&1&0&0&0&1\end{bmatrix}$
(W, W, B, W)	$C_{WWBW}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&1&0&0&1&1&0&1\\0&0&1&0&1&0&1&1\end{bmatrix}$
(W, W, B, B)	$C_{WWBW}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&0&0&1&0&1&1&1\\0&0&1&0&1&0&1&1\end{bmatrix}$
(W, B, W, W)	$C_{WWBB}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&0&0&1&0&1&1&1\\0&1&0&0&1&1&0&1\end{bmatrix}$
(W, B, W, B)	$C_{WBWB}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&0&0&1&0&1&1&1\\1&1&0&1&0&1&0&0\end{bmatrix}$
(W, B, B, W)	$C_{WBBW}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&1&0&0&1&1&0&1\\1&0&0&0&1&1&1&0\end{bmatrix}$
(W, B, B, B)	$C_{WBBB}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&1&0&0&1&1&0&1\\1&1&1&0&1&0&0&0\end{bmatrix}$
(B, W, W, W)	$C_{BWWW}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&1&1&1&0&0&0&1\\0&0&1&0&1&0&1&1\end{bmatrix}$
(B, W, W, B)	$C_{BWWB}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&1&1&1&0&0&0&1\\1&0&1&1&0&0&1&0\end{bmatrix}$
(B, W, B, W)	$C_{BWBW}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\1&1&0&1&0&1&0&0\\0&0&0&1&0&1&1&1\end{bmatrix}$
(B, W, B, B)	$C_{BWBB}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&1&1&1&0&0&0&1\\1&1&1&0&1&0&0&0\end{bmatrix}$
(B, B, W, W)	$C_{BBWW}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\0&1&1&1&0&0&0&1\\0&1&0&0&1&1&0&1\end{bmatrix}$
(B, B, W, B)	$C_{BBWB}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\1&1&0&1&0&1&0&0\\0&1&0&0&1&1&0&1\end{bmatrix}$
(B, B, B, W)	$C_{BBBW}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\1&1&0&1&0&1&0&0\\0&1&1&1&0&0&0&1\end{bmatrix}$
(B, B, B, B)	$C_{BBBB}=\begin{bmatrix}0&0&0&0&1&1&1&1\\0&0&1&1&0&0&1&1\\1&1&0&1&0&1&0&0\\1&1&1&0&1&0&0&0\end{bmatrix}$

The detail computation of w and b for Table 2 is as follows. First, statements 1–4 below are true for each basis matrix in Table 2. Therefore, every element of the Black-pool is 1/8, and each element of the White-pool is 3/8. So the maximum element of the Black-pool (i.e. b) is 1/8, and the minimum element of the White-pool (i.e. w) is 3/8, and contrast α is thus 3/8–1/8 = 1/4.

1. When the 1st and 3rd rows are stacked, if the first subscript in the matrix name is B, then the ratio of 0s in the stacking result is 1/8 = 12.5%; otherwise, the ratio is 3/8 = 37.5%.
2. When the 2nd and 4th rows are stacked, if the second subscript in the matrix name is B, then the ratio of 0s in the stacking result is 1/8 = 12.5%; otherwise, the ratio is 3/8 = 37.5%.
3. When the 2nd and 3rd rows are stacked, if the 3rd subscript in the matrix name is B, then the ratio of 0s in the stacking result is 1/8 = 12.5%; otherwise, the ratio is 3/8 = 37.5%.
4. When the 1st and 4th rows are stacked, if the 4th subscript in the matrix name is B, then the ratio of 0s in the stacking result is 1/8 = 12.5%; otherwise, the ratio is 3/8 = 37.5%. □

3.2. The proof of conditionally optimal contrast in non-opaque-oriented FVC

Non-opaque-oriented FVC also satisfies Eqs. (2)–(10) of Section 2. In the following, w is the *luminance transmission* of stacking result to represent white pixel W, b is the *luminance transmission* of stacking result to represent black pixel B, and $\alpha = w - b$ is the contrast. Some basis matrices are considered below to gain the upper bound of contrast α.

I. Consider the basis matrix C_{WWBB}. By Eqs. (7)–(10), the *luminance transmission* of the four secret pixels $s_1(i,j)$, $s_1(i,m-1-j)$, $s_2(i,j)$, $s_2(i,m-1-j)$ are

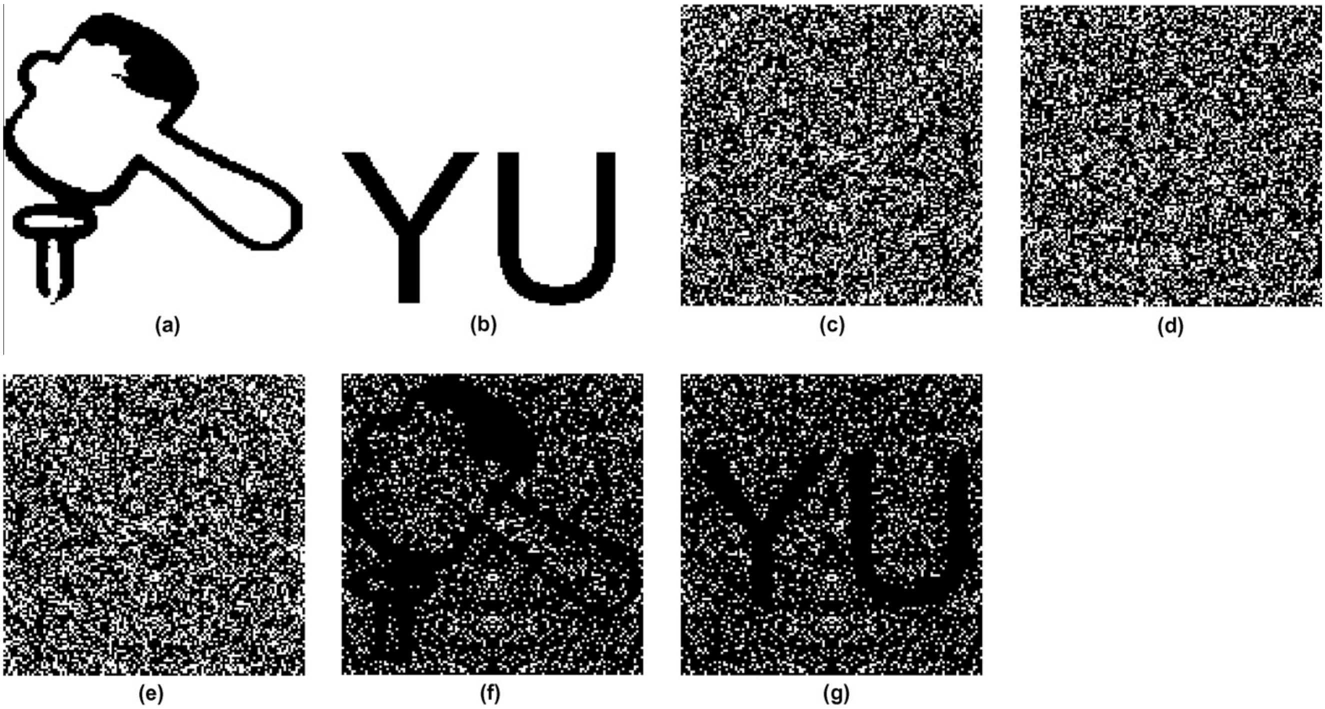

Fig. 4. The experimental result of the opaque-oriented FVC: (a and b): the secret images; (c and d): the two generated transparencies; (e): flipping (c) over; (f): the result of stacking (c) and (d) together; (g): the result of stacking (d) and (e) together.

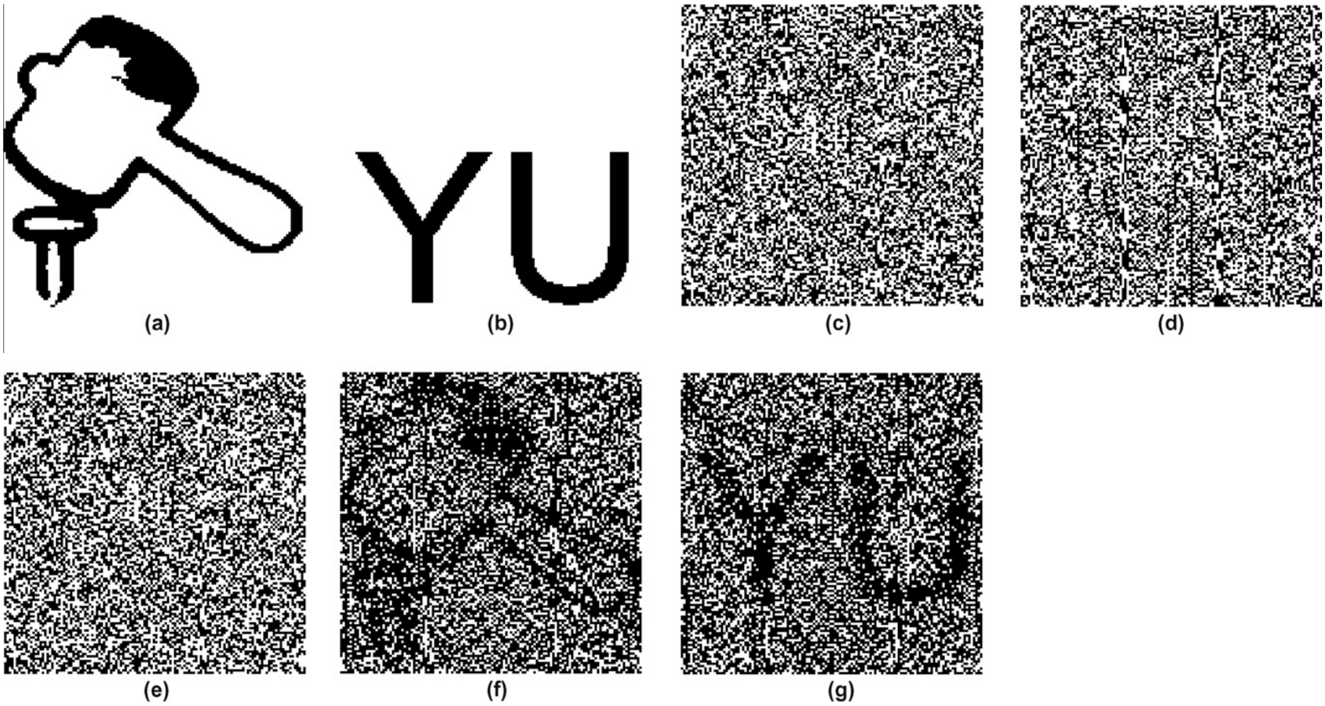

Fig. 5. The experimental result of the non-opaque-oriented FVC: (a and b): the secret images; (c and d): the two generated transparencies; (e): flipping (c) over; (f): the result of stacking (c) and (d) together; (g): the result of stacking (d) and (e) together.

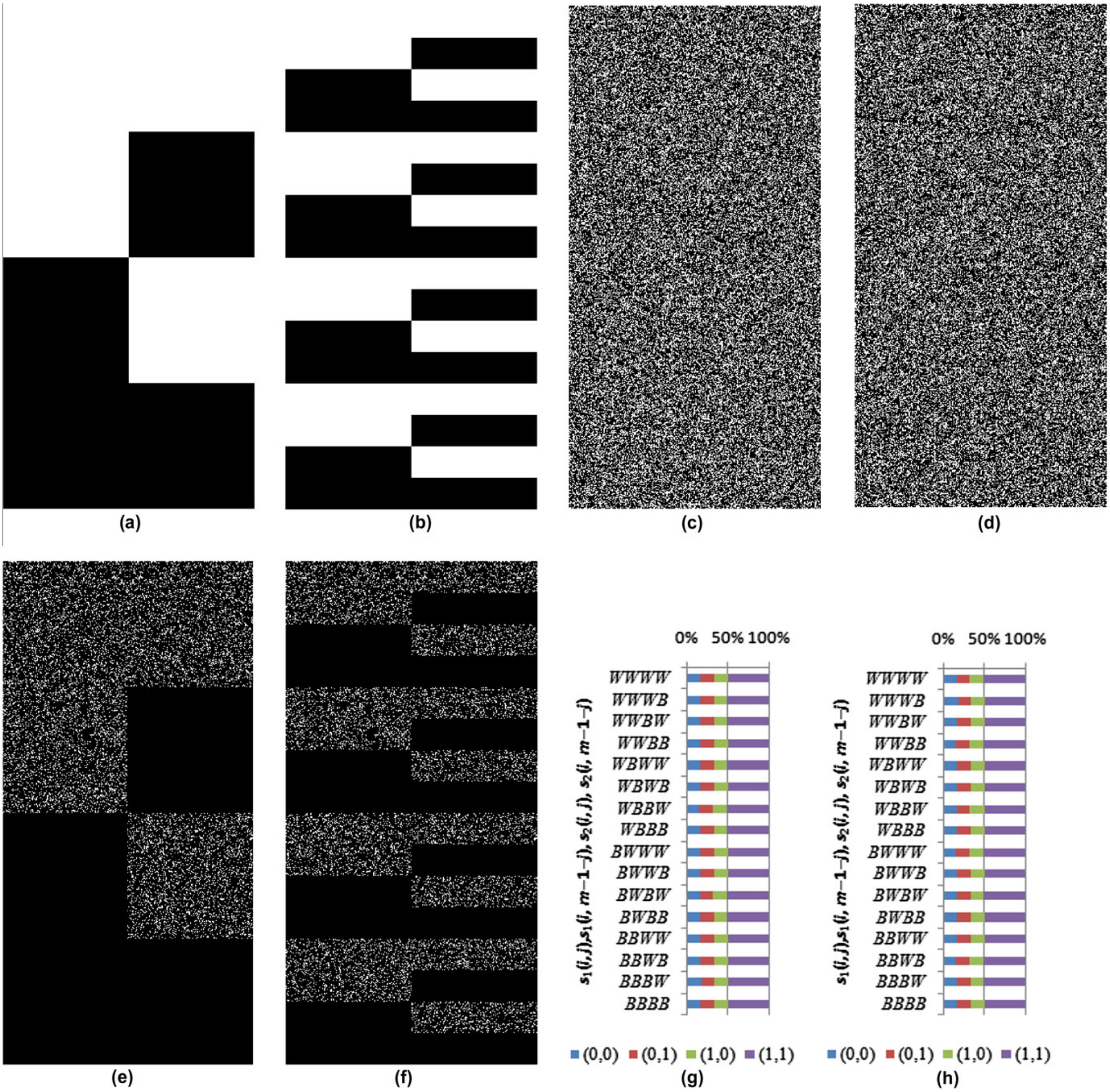

Fig. 6. Security test of Scheme 1. (a and b): The two secret images; (c and d): the two generated transparencies; (e and f): the two stacking results; (g): statistical result of (c); (h): statistical result of (d).

$$c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1}=w,$$
$$c_{0,0}+c_{0,2}+c_{2,0}+c_{2,2}=w,$$
$$c_{0,0}+c_{0,1}+c_{2,0}+c_{2,1}=b,\quad \text{and}$$
$$c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2}=b.$$

By Eqs. (5) and (6), $b_1 = c_{0,1} + c_{1,1} + c_{2,1} + c_{3,1} \geqslant c_{1,1}$, and $b_2 = c_{0,2} + c_{1,2} + c_{2,2} + c_{3,2} \geqslant c_{2,2}$. Therefore,

$$\begin{aligned}\alpha = w - b = [w + w - b - b]/2 = [(c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1})\\ +(c_{0,0}+c_{0,2}+c_{2,0}+c_{2,2})-(c_{0,0}+c_{0,1}+c_{2,0}+c_{2,1})\\ -(c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2})]/2=[(c_{1,1}+c_{2,2})\\ -(c_{1,2}+c_{2,1})]/2 \leqslant (c_{1,1}+c_{2,2})/2 \leqslant (b_1+b_2)/2. \qquad (16)\end{aligned}$$

II. Consider the basis matrix C_{WWWW} and C_{BBBB}. For the basis matrix $C[W,W,W,W]$, by Eqs. (7)–(10),

$$c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1}=w,$$
$$c_{0,0}+c_{0,2}+c_{2,0}+c_{2,2}=w,$$
$$c_{0,0}+c_{0,1}+c_{2,0}+c_{2,1}=w,\quad \text{and}$$
$$c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2}=w.$$

By Eq. (5), $a_0 = c_{0,0} + c_{0,1} + c_{0,2} + c_{0,3} \geqslant c_{0,0} + c_{0,1} + c_{0,2}$, and $a_1 = c_{1,0} + c_{1,1} + c_{1,2} + c_{1,3} \geqslant c_{1,0} + c_{1,1} + c_{1,2}$.

By Eq. (6), $b_0 = c_{0,0} + c_{1,0} + c_{2,0} + c_{3,0} \geqslant c_{0,0} + c_{1,0}$. Therefore,

$$\begin{aligned}w &= [(c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1})+(c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2})]/2\\ &=[(c_{0,0}+c_{0,1}+c_{0,2})+(c_{1,0}+c_{1,1}+c_{1,2})+(c_{0,0}+c_{1,0})]/2\\ &\leqslant (a_0+a_1+b_0)/2. \qquad (17)\end{aligned}$$

For the basis matrix C_{BBBB}, by Eqs. (7)–(10),

Table 3
Characterization of VC methods.

Methods	Pixel-expansion factor	Number of hidden secrets
Naor and Shamir [2]	4	Single
Yang's [6]	1	
Shyu [8]	1	
Wu and Chang [3]	4	Double
Shyu et al. [5]	4	
The proposed method	1	Double

$$\begin{aligned}
&c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1}=b,\\
&c_{0,0}+c_{0,2}+c_{2,0}+c_{2,2}=b,\\
&c_{0,0}+c_{0,1}+c_{2,0}+c_{2,1}=b, \quad \text{and}\\
&c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2}=b.
\end{aligned}$$

By Eq. (12), $b_3 = c_{0,3} + c_{1,3} + c_{2,3} + c_{3,3} \geqslant c_{0,3} + c_{1,3}$. Therefore,

$$\begin{aligned}
b &= [(c_{0,0}+c_{0,1}+c_{1,0}+c_{1,1})+(c_{0,0}+c_{0,2}+c_{1,0}+c_{1,2})]/2\\
&\geqslant [(c_{0,0}+c_{0,1}+c_{0,2})+(c_{1,0}+c_{1,1}+c_{1,2})]/2\\
&= [(c_{0,0}+c_{0,1}+c_{0,2}+c_{0,3})+(c_{1,0}+c_{1,1}+c_{1,2}+c_{1,3})-(c_{0,3}\\
&\quad +c_{1,3})]/2\\
&\geqslant (a_0+a_1-b_3)/2. \qquad (18)
\end{aligned}$$

By Eqs. (4), (17) and (18),

$$\begin{aligned}
\alpha = w-b &\leqslant (a_0+a_1+b_0)/2-(a_0+a_1-b_3)/2\\
&= (b_0+b_3)/2. \qquad (19)
\end{aligned}$$

By Eqs. (16) and (19),

$$\begin{aligned}
\alpha &\leqslant [(b_1+b_2)/2+(b_0+b_3)/2]/2 = (b_0+b_1+b_2+b_3)/4\\
&= 1/4 \quad (\text{By Eq.}(6)).
\end{aligned}$$

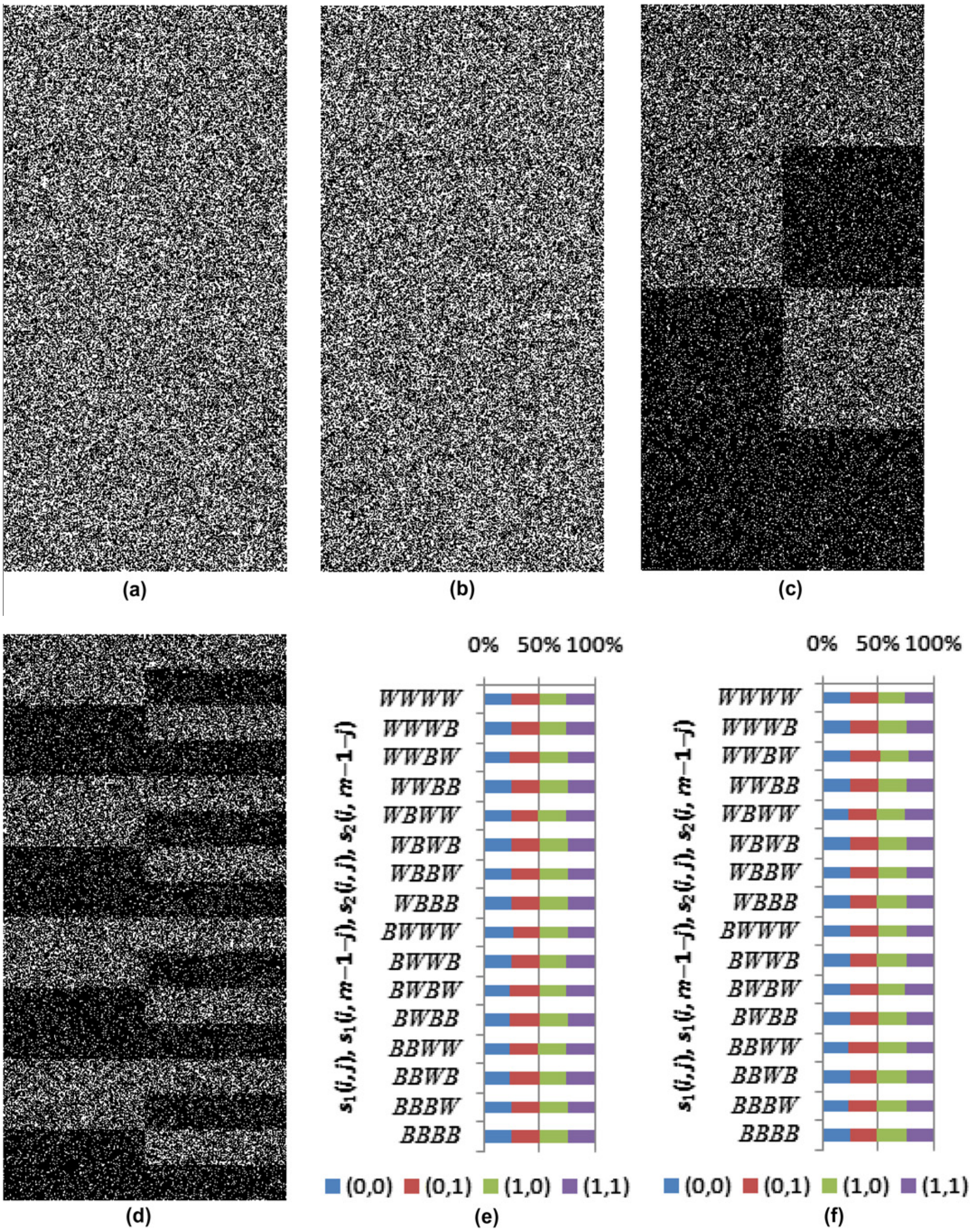

Fig. 7. Security test of Scheme 2. (a and b): the two generated transparencies; (c and d): the two stacking results; (e): statistical result of (a); (f): statistical result of (b).

Therefore, if non-opaque-oriented FVC is used, then the conditionally-optimal contrast is 1/4. The result is better than the conditionally-optimal contrast value 1/6 of the opaque-oriented FVC in Section 2. The encoding matrices shown in Table 2 are conditionally optimal, because (a) Property 2 shows that, for these matrices, the contrast of stacking result is 1/4; and (b) the proof given above indicates that: under the perfect Security constraint, no basis matrices can yield a contrast larger than 1/4.

4. Experimental results

Experiments and comparisons are presented in this section. Section 4.1 presents the results of the proposed method. Section 4.2 gives the security testing of the transparencies. Section 4.3 shows the comparisons with other studies. Section 4.4 shows the expanded version of our method.

4.1. Experiments of proposed method

This subsection presents experimental results for the proposed scheme which can generate non-expanded transparencies with perfect security and can decode one more secret image by flipping one of the transparencies. The opaque-oriented FVC experiment is shown in Fig. 4. The two secret images are displayed in Fig. 4(a) and (b); and the two generated non-expanded transparencies are shown in Fig. 4(c) and (d). Fig. 4(e) shows the result of flipping Fig. 4(c) over. Fig. 4(f) shows the result of stacking 4(c) and 4(d) together; Fig. 4(g) shows the result of stacking 4(c) and (e) together. The non-opaque-oriented FVC experiment is shown in Fig. 5. The two secret images are displayed in Fig. 5(a) and (b); and the two generated non-expanded transparencies are shown in Fig. 5(c) and (d). Fig. 5(e) shows the result of flipping Fig. 5(c) over. Fig. 5(f) shows the result of stacking 5(c) and (d) together; Fig. 5(g) shows the result of stacking Fig. 5(c) and (e) together.

4.2. Security test of proposed method

In this subsection, we conduct two experiments for security testing. The first one is for Scheme 1 and the second one is for Scheme 2. Fig. 6 shows the first experiment. Fig. 6(a) and (b) illustrates two secret images S_1 and S_2, which consist of 16 sub-regions from top to down, and in each sub-region, $s_1(i,j)$ and $s_1(i,m-1-j)$ are at the left-hand and right-hand sides of S_1, and $s_2(i,j)$ and $s_2(i,m-1-j)$ are at the left-hand and right-hand sides of S_2. Then the four sections (the left-hand and right-hand sides of S_1 and the left-hand and right-hand sides of S_2) in each sub-region are painted using all possible colors $\{WWWW, WWWB, \ldots, BBBB\}$. In other words, each sub-region is encoded with a basis matrix being referred to.

The generated transparencies T_1 and T_2 are shown in Fig. 6(c) and (d). The result of stacking T_1 and T_2 together is shown in Fig. 6(e). When T_1 is flipped and then stacked with T_2, the stacking result is shown in Fig. 6(f). To test security of T_1, in each sub-region of T_1, we count the probability distribution of symmetric pairs $[t_1(i,j), t_1(i,m-1-j)] \in \{[0,0],[0,1],[1,0],[1,1]\}$. Fig. 6(g) shows the statistical result. The probability distributions are about $[1/6, 1/6, 1/6, 3/6]$, no matter which basis matrix is used (the small variance is caused by randomly choosing a column in the basis matrix; so it is unrelated to the secret pixels, i.e. the intruder cannot judge the secret values by the small variance). Therefore, if an intruder only has T_1 (e.g. Fig. 6(c)), then no information about the secret image is unveiled. Fig. 6(h) shows statistical analysis of the second transparency. The result is similar to Fig. 6(g), so it is also secure.

Fig. 7 shows the second experiment. The two secret images are the images in Fig. 6(a) and (b). The generated transparencies T_1 and

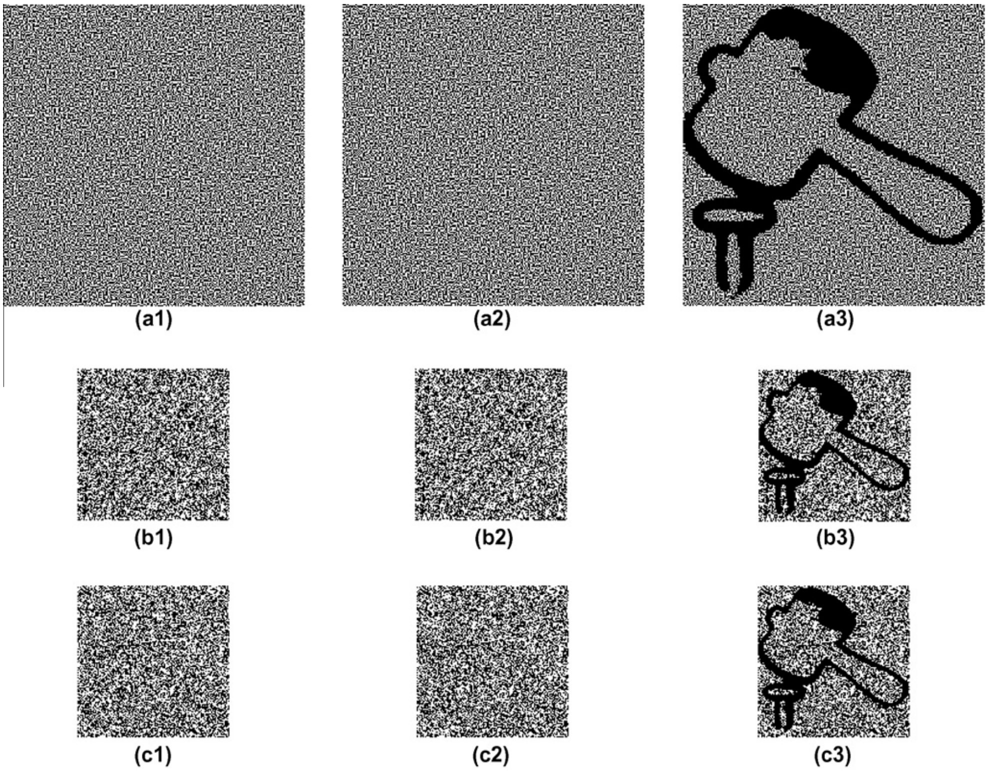

Fig. 8. Three “single-secret” (2,2) VC methods vs. our “double-secrets” FVC method. (a): Naor and Shamir’s method: (a1 and a2) are the two generated transparencies, and (a3) is the stacking result. (b): Yang’s method: (b1 and b2) are the two generated transparencies, and (b3) is the stacking result. (c): Shyu’s method: (c1 and c2) are the two generated transparencies, and (c3) is the stacking result. (d): Ours.

T_2 are shown in Fig. 7(a) and (b). The result of stacking T_1 and T_2 together is shown in Fig. 7(c). When T_1 is flipped and is stacked with T_2, we have the stacking result shown in Fig. 7(d). Fig. 7(e) shows the statistical result of T_1 where the probability distribution of symmetric pair $[t_1(i,j), t_1(i,m-1-j)] = \{[0,0], [0,1], [1,0], [1,1]\}$ are about [1/4, 1/4, 1/4, 1/4]. Fig. 7(f) shows the statistical result of T_2. Therefore, the two transparencies T_1 and T_2 are both secure.

4.3. Comparison with other studies

Table 3 lists the comparisons with previously reported VC methods [2,3,5,6,8]. Many reported methods had pixel expansion problem; and non-expanded methods often encoded only a single secret image. The proposed method encodes double secret images, and does not cause any pixel expansion.

Previously reported VC methods [2,3,5,6,8] are implemented. First, single-secret VCs [2,6,8] are demonstrated in Fig. 8, and let the number of transparencies is two for each method. Fig. 8(a)–(c) shows Naor and Shamir's method [2]. The expansion rate is 4 (ours is 1), and the contrast is 1/2 (ours is 1/6 or 1/4). Fig. 8(d)–(f) shows Yang's method [6]. The contrast is 1/2, and the stacking result (f) is also tumultuous and hence not as good as Naor and Shamir's; but this is because there is no expansion (just like ours). Fig. 8(g)–(i) shows Shyu's method [8], and the result is similar to Yang's. Notably, the three methods [2,6,8] only encode a single secret image in two transparencies, but the proposed method encode two secret images; so [2,6,8] has better visual quality than ours.

Next, in Figs. 9 and 10, we demonstrate two circular VC methods [3,5]. Both methods encode multiple secret images in two circular transparencies, and each secret image is revealed by stacking

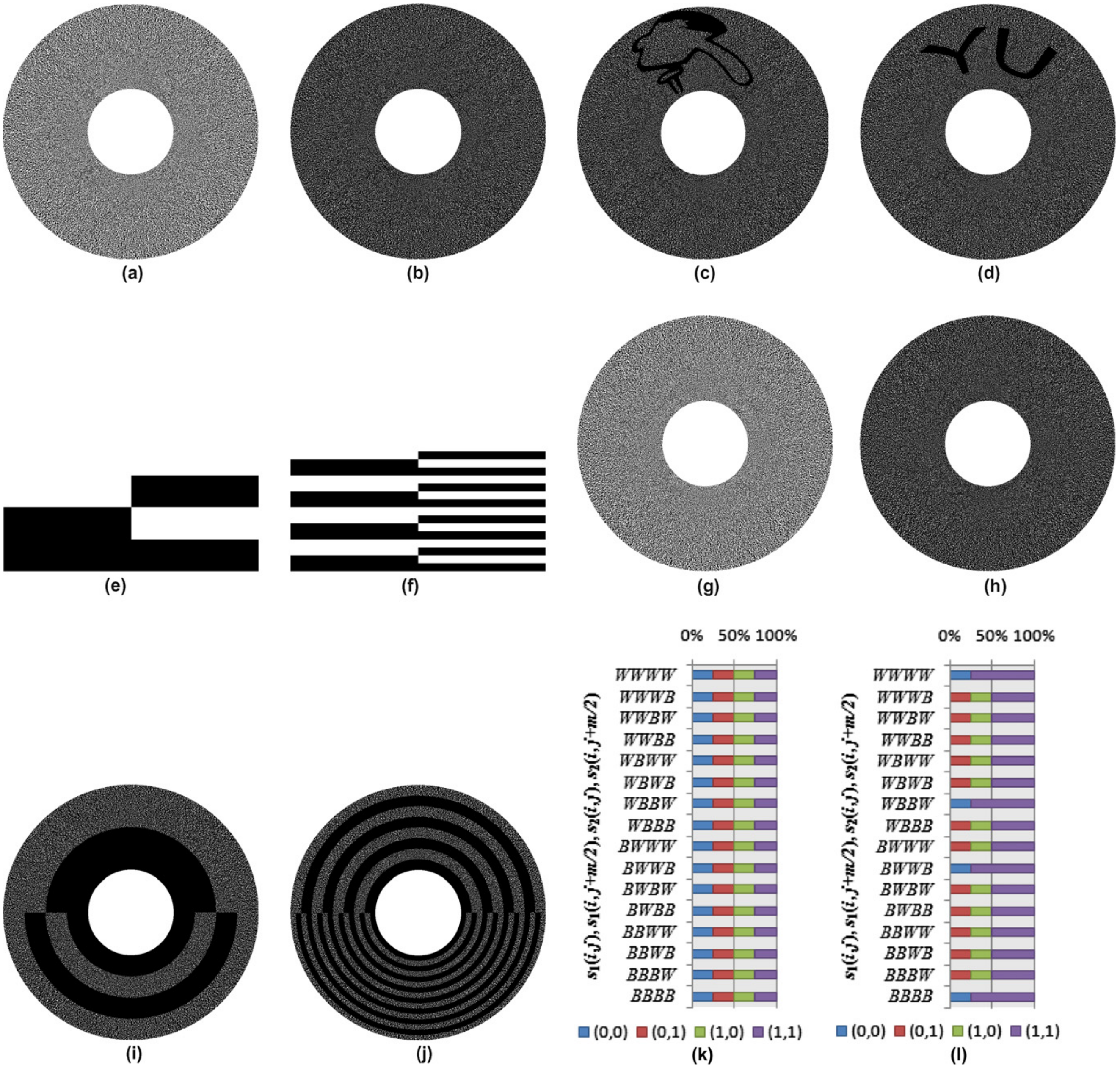

Fig. 9. About the method of Wu and Chang [3]. The expansion rate is 4. (a and b): Two generated circular transparencies T_1 and T_2. (c): The result of stacking T_1 and T_2. (d): The result of stacking rotational T_1 with T_2. (e and f): Two new secret images S_1 and S_2. (g and h): Two new transparencies T_1 and T_2 generated from S_1 and S_2. (i): The result of stacking T_1 with T_2. (j): The result of stacking rotational T_1 with T_2. (k): The probability distribution of symmetric pairs for all 16 sub-regions in T_1. (l): The probability distribution of symmetric pairs for all 16 sub-regions in T_2.

the two transparencies with a rotation of the first transparency using a pre-defined degree. To facilitate the comparison, let the number of secret images be two, and the rotational degrees be 0° and 180°.

Fig. 9(a) and (b) is the two circular transparencies T_1 and T_2 generated by Wu and Chang [3] in which the expansion rate is 4 (each secret pixel is represented as a 2 × 2 block in two transparencies), and the contrast is 1/4. Fig. 9(c) is the results of stacking T_1 and T_2; and Fig. 9(d) is the results in which T_1 is rotated 180° and stacked with T_2. Let m denote the width of secret image, two pixels $t_1(i,j)$ and $t_1(i,j+m/2)$ are at two opposite positions in T_1, and so are $t_2(i,j)$ and $t_2(i,j+m/2)$ in T_2. In the stacking, the secret S_1 is revealed by stacking $t_1(i,j)$ with $t_2(i,j)$ to decode $s_1(i,j)$, and stacking $t_1(i,j+m/2)$ with $t_2(i,j+m/2)$ to decode $s_1(i,j+m/2)$; the second secret is revealed by stacking $t_1(i,j+m/2)$ with $t_2(i,j)$ to decode $s_2(i,j)$, and stacking $t_1(i,j)$ with $t_2(i,j+m/2)$ to decode $s_2(i,j+m/2)$. Therefore, the two pixel values $(t_1(i,j), t_1(i,j+m/2))$ form a symmetric pair, and so do $(t_2(i,j), t_2(i,j+m/2))$. Since the four secret pixels $[s_1(i,j), s_1(i,j+m/2), s_2(i,j), s_2(i,j+m/2)]$ have $2^4 = 16$ possible colors $\{WWWW, WWWB, \ldots, BBBB\}$, to test the security of 16 types of colors, an experiment is shown in Fig. 9(e)–(l). Fig. 9(e) and (f) illustrates two secret images S_1 and S_2, which consist of 16 equal sub-regions from top to bottom, and in each sub-region, $s_1(i,j)$ and $s_1(i,j+m/2)$ are at the left and right sides of S_1, and $s_2(i,j)$ and $s_2(i,j+m/2)$ are at the left and right sides of S_2. Then those four sections (the left and right sides of S_1 and the left and right sides of S_2) in each sub-region are painted using all possible colors $\{WWWW, WWWB, \ldots, BBBB\}$, respectively. The generated transpar-

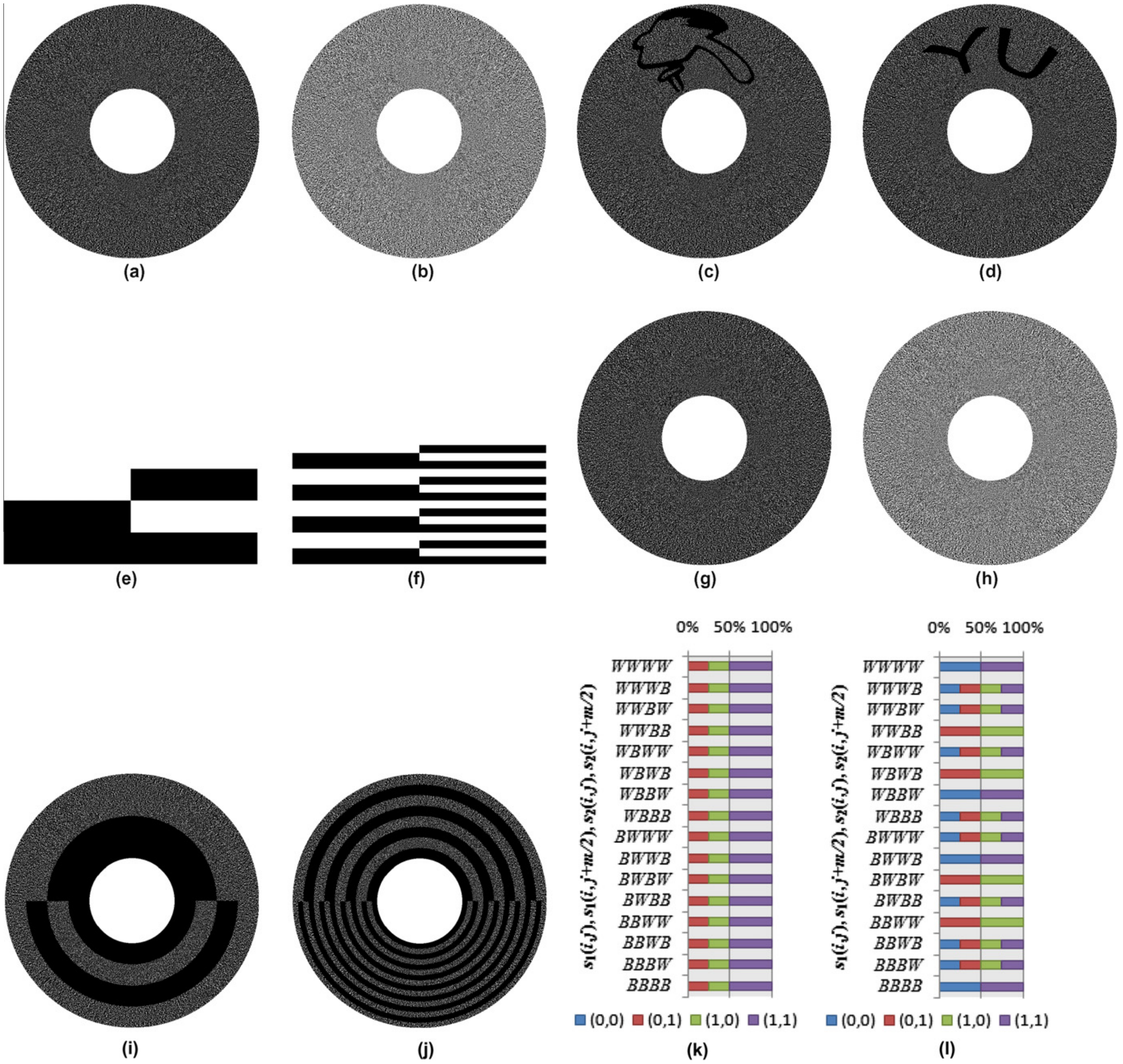

Fig. 10. About the method of Shyu et al. [5]. The expansion rate is 4. (a and b): Two generated circular transparencies T_1 and T_2. (c): The result of stacking T_1 and T_2. (d): The result of stacking rotational T_1 with T_2. (e and f): Two new secret images S_1 and S_2. (g and h): Two new transparencies T_1 and T_2 generated from S_1 and S_2. (i): The result of stacking T_1 with T_2. (j): The result of stacking rotational T_1 with T_2. (k): The probability distribution of symmetric pairs for all 16 sub-regions in T_1. (l): The probability distribution of symmetric pairs for all 16 sub-regions in T_2.

encies are shown in Fig. 9(g) and (h). Fig. 9(i) shows the result of stacking T_1 and T_2, and Fig. 9(j) is the result of stacking the rotated T_1 with T_2.

To inspect the security issue of transparency T_1, Fig. 9(k) displays the probability distribution of symmetric pairs $[t_1(i,j), t_1(i,j+m/2)] \in \{[0,0], [0,1], [1,0], [1,1]\}$ in each sub-region, where the probabilities are [1/4, 1/4, 1/4, 1/4] for all types of colors. Therefore, the first transparency is secure, because the intruder cannot judge the values of secret pixels $[s_1(i,j), s_1(i,j+m/2), s_2(i,j), s_2(i,j+m/2)]$ by observing the probability distribution of symmetric pairs. However, as shown in Fig. 9(l). The transparency T_2 leaks some information; because in S_1 and S_2, when the four secret pixels $[s_1(i,j), s_1(i,j+m/2), s_2(i,j), s_2(i,j+m/2)] \in \{WWWW, WBBW, BWWB, BBBB\}$, then the probability distribution of symmetric pairs $[t_2(i,j), t_2(i,j+m/2)] \in \{[0,0], [0,1], [1,0], [1,1]\}$ is [1/4,0,0,3/4]; when $[s_1(i,j), s_1(i,j+m/2), s_2(i,j), s_2(i,j+m/2)] \notin \{WWWW, WBBW, BWWB, BBBB\}$, then the probabilities are [0,1/4,1/4,2/4], respectively. Hence, the intruder can judge whether the four secret pixels $[s_1(i,j), s_1(i,j+m/2), s_2(i,j), s_2(i,j+m/2)]$ are $\{WWWW, WBBW, BWWB, BBBB\}$ or not. In other words, if [0,0] pair or [1,1] pair appear in second transparency T_2, then we can claim that the corresponding position of secret images (S_1 and S_2) must be either [*WWWW*] or [*WBBW*] or [*BWWB*] or [*BBBB*]. Likewise, if [0,1] pair or [1,0] pair appear in second transparency T_2, then we can claim that the corresponding position of input images (S_1 and S_2) cannot be [*WWWW*] or [*WBBW*] or [*BWWB*] or [*BBBB*]. In summary, secret-leaking occurs in transparency 2.

Fig. 10 is a demonstration about the method of Shyu et al. [5]. Fig. 10(a) and (b) is the two generated circular transparencies in which the expansion rate is 4, and the contrast is 1/4. Fig. 10(c) is the results of stacking (a) with (b), and Fig. 10(d) is the results of stacking rotated (a) with (b). The security test is shown in Fig. 10(e) –(l). Fig. 10(e) and (f) is the two new secret images which are the same as Fig. 9(e) and (f). Fig. 10(g) and (h) is the two generated transparencies, and the stacking results are Fig. 10(i) and (j). The security of T_1 is shown in Fig. 10(k), where the probability distribution of symmetric pairs $[t_1(i,j), t_1(i,j+m/2)] \in \{(0,0), (0,1), (1,0), (1,1)\}$ is [0,1/4,1/4,2/4] in all types of colors, so T_1 is secure. On the other hand, T_2 may leaks some information. The security of T_2 is shown in Fig. 10(l). When the four secret pixels $[s_1(i,j), s_1(i,j+m/2), s_2(i,j), s_2(i,j+m/2)] \in \{WWWW, WBBW, BWWB, BBBB\}$, the probability distribution of symmetric pairs is [1/2,0,0,1/2]; when the four secret pixels are $\{WWBB, WBWB, BWBW, BBWW\}$, the probability distribution is [0,1/2,1/2,0]. When the four secret pixels are in $\{BWWW, WBWW, WWBW, WWWB, WBBB, BWBB, BBWB, BBBW\}$, the probability distribution is [1/4,1/4,1/4,1/4]. Therefore, the intruder can judge and divide the four secret pixels $[s_1(i,j), s_1(i,j+m/2), s_2(i,j), s_2(i,j+m/2)]$ to 3 sets by observing the probability distribution of symmetric pairs in transparency T_2.

Figs. 9 and 10 show two well-known circular VCs [3,5]. Stacking results of the two methods [3,5] are 100% opaque both, and their contrast $\alpha = 1/4$ is better than our 1/6. But, as shown in Figs. 9(l) and 10(l), Methods [3,5] are not of perfect security: the second transparency generated by [3,5] have secret-leaking problem. In summary, under the constraint of avoiding secret-leaking (the fundamental requirement of VC), the best contrast value can be achieved is 1/6 (or 1/4, if the block pixels in stacking results are not restricted to 100% opaque), and ours already achieve this optimal contrast value 1/6 for Scheme 1 (and 1/4 for Scheme 2). So we may say that ours are with conditionally optimal contrast under perfect-security requirement. As for others (for example [3,5]), they might have contrast values better than ours, but it is because their methods did not meet perfect-security requirement.

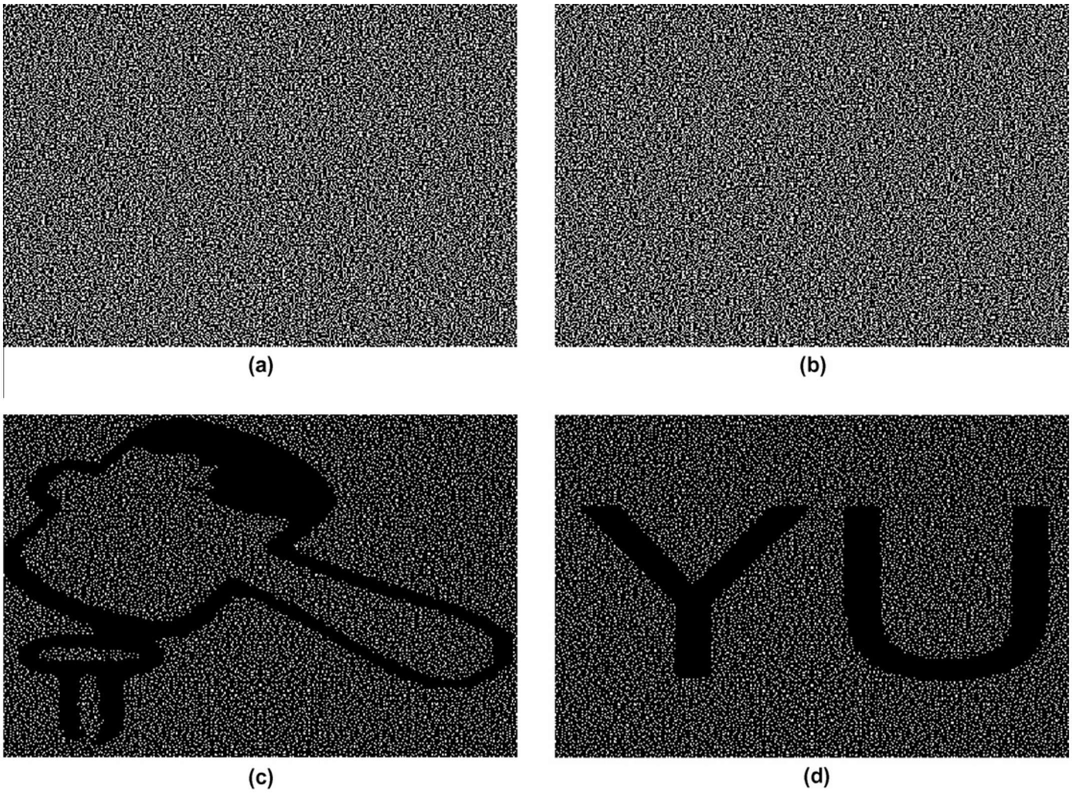

Fig. 11. The expanded version (block-based rather than pixel-based) of our Scheme 1. (a and b): The two generated transparencies where the two secret images are Fig. 4(a) and (b). (c): The result of stacking (a) and (b). (d): The result of stacking (b) with the flipped version of (a).

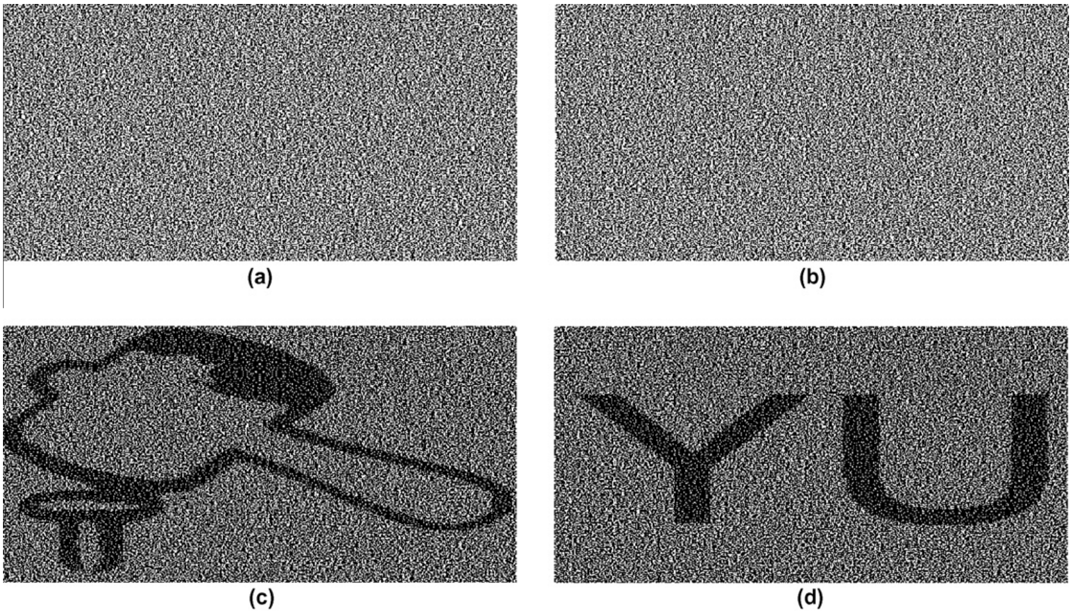

Fig. 12. The expanded version (block-based rather than pixel-based) of our Scheme 2. (a and b): The two generated transparencies where the two secret images are Fig. 4(a) and (b). (c): The result of stacking (a) and (b). (d): The result of stacking (b) with the flipped version of (a).

4.4. The expanded version of our method

In order to yield no expansion, we use probability model to encode the shares. The price is that it may cause non-harmonic disarray of stacking result. If we are not constrained by the no-expansion rule, then all columns of basis matrix are used to encode the secret pixels $[s_1(i,j), s_1(i,j+m/2), s_2(i,j), s_2(i,j+m/2)]$, therefore, the expansion rate is the value of r. The results are as shown in Figs. 11 and 12. Fig. 11 shows the expanded version of flip. Scheme 1, with the expansion rate being $6 = 3 \times 2$. (Notably, $r = 6$ is the minimal r we can have for Scheme 1. On the other hand, r will also be the expansion rate for our expanded version. So, in the expanded version, our minimal expansion rate will be 6 for Scheme 1 [8 for Scheme 2 because minimal r is 8 for Scheme 2].) Fig. 11 (a) and (b) shows the two generated transparencies, and (c and d) show the stacking results. Fig. 12 shows the expanded version of flip Scheme 2, with the expansion rate being $8 = 4 \times 2$. Fig. 12(a) and (b) shows the two generated transparencies and (c and d) show the stacking results. We can see that the visual quality is competitive again. In summary, the disarray of stacking result is due to the requirement of no-expansion, along with the perfect security for double secret; but the major weakness of pixel-expansion VC is that the size of transparencies will expand several times and waste space for carrying or storage.

5. Discussion

In this section, some related topics are discussed in this section. Section 5.1 addresses the method of finding the basis matrices of FVC, and Section 5.2 shows the contrast values of the proposed method by other definition of contrast.

5.1. How to find the basis matrices of FVC

Basically, we may say that people can create these basis matrices by exhaustive search, as long as they meet the specified requirements. However, in reality, to save searching time, some basis matrices can be generated from others by exchanging rows. For instance, suppose the matrix C_{BWWW} of Scheme 1 is set to

$$C_{BWWW} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix},$$

where the B is represented by stacking the 1st and 3rd rows to obtain six 1s, and the three W are represented by stacking the 2nd and 4th rows, the 2nd and 3rd rows, and the 1st and 4th rows, respectively, to obtain one 0 and five 1s. Then the 1st and 2nd rows can be exchanged to get

$$C_{WWBW} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix},$$

where the B is by stacking the 2nd and 3rd rows, and the three W are by stacking the 1st and 3rd rows, the 2nd and 4th rows, and the 1st and 4th rows. Using this method, we can generate four basis matrices $C_{BWWW}, C_{WBWW}, C_{WWBW}$ and C_{WWWB}, as long as one of the four matrices is found.

Actually, all the 16 basis matrices can be divided into 6 sets, namely, $\{C_{WWWW}\}, \{C_{BWWW}, C_{WBWW}, C_{WWBW}, C_{WWWB}\}, \{C_{WBBW}, C_{BWWB}, C_{BWBW}, C_{WBWB}\}, \{C_{WWBB}, C_{BBWW}\}, \{C_{BBBW}, C_{BBWB}, C_{BWBB}, C_{WBBB}\}$, and $\{C_{BBBB}\}$. In each set, only one matrix needs to be found, and the remaining is generated by exchanging the rows. Therefore, only 6 basis matrices are actually searched.

Next, to search the basis matrices, we need decide the value of r. The factor to determine the value r is the contrast of the constructed basis matrices. For basis matrices whose width is r, the possible contrast is $1/r, 2/r, 3/r, \ldots, r/r$. In Section 2.2, we have proved that the upper bound of contrast of Scheme 1 is 1/6. In symbols, the contrast is

$$\{i/r|r, i \in Z, 0 < i/r \leqslant 1/6\}.$$

To reach 1/6 (the upper bound of contrast for Scheme 1), the value r must be a multiple of 6. If r is not a multiple of 6, then the possible contrast i/r cannot equal to 1/6, so the contrast will be less than 1/6.

Analogously, in Scheme 2, the width (i.e. value r) of basis matrices must be a multiple of 4, because in Section 3.2 we already proved that 1/4 is the upper bound of the contrast for Scheme 2. Unfortunately, when $r = 4$, we could not find the basis matrices even after exhaustive search. So we tried $r = 8$ and obtained the basis matrices shown in Table 2 whose contrast reached the upper bound 1/4.

5.2. Discussion about contrast values

In our method, the two definitions follow the basis matrices definitions which are given by Naor and Shamir [2], but some details are modified to conform to the structure of FVC. The (t,n) visual cryptography, which is defined by Naor and Shamir [2], needs two basis matrices to encode a secret pixel which has only two values $\{W,B\}$ in a secret image; however, our method needs consider four secret pixels simultaneously (two pixels in S_1 and two pixels in S_2), so it needs $2^4 = 16$ basis matrices to encode four pixels. However, the contrast evaluation in [2] did not consider the fact that, in darker image, human eyes have higher sensitiveness about (real-life-sense) contrast. (This fact was mentioned in Section 3 of Ref. [1] by Liu et al.) To overcome the drawback, we bring up two schemes in the proposed method, where Scheme 1 set the black color of stacking result is 100% opaque, and Scheme 2 do not set the constraint. We let the readers choose the one they like.

Liu et al. [16] give a new definition of contrast for expanded VC (i.e. each secret pixel is encoded into many pixels in transparencies), but our design is a non-expanded VC (i.e. each secret pixel is encoded into a pixel in transparencies). For readers benefit, we also give the contrast value defined by Liu et al. [16], when the expanded VC version shown in Figs. 11 and 12 are used. The expanded Scheme 1 has contrast

$$\alpha^{\text{Liu}} = \frac{(h-l)m}{h(m-h)+l(m-l)+m^2} \xlongequal{h=6,l=5,m=6} \frac{6}{41} \approx 0.146,$$

and the expanded Scheme 2 has contrast

$$\alpha^{\text{Liu}} = \frac{(h-l)m}{h(m-h)+l(m-l)+m^2} \xlongequal{h=7,l=5,m=8} \frac{8}{43} \approx 0.186.$$

6. Conclusions

Opaque-oriented and non-opaque-oriented FVC schemes are both introduced in this paper. We have proved that both schemes satisfy perfect security and they are conditionally optimal in contrast. The generated transparencies do not lead to any expansion of size. The experimental results show the revealing of double-secrets via flipping and stacking the transparencies together.

Just like other VC methods, the whole decoding process uses no computer or any computation; so the decoding is very fast, and can be used in environment where computer is not stable or available. Due to the double-secrets feature of the proposed method, one of the applications is the double checking of ownership for personality identification. Since the size is non-expanded, the space needed to carry a transparency to a meeting is economic (size is the same as the space needed to carry an original image).

Acknowledgments

The work is supported by National Science Council of Republic of China, under Grant NSC97-2221-E-009-120-MY3. The authors thank the reviewers for valuable suggestions to improve the paper.

References

[1] S.K. Chen, S.J. Lin, Non-expansible flip-flop visual cryptography with perfect security, in: The Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 2009.

[2] M. Noar, A. Shamir, Visual cryptography, Advances in Cryptology — EUROCRYPT'94 (1995) 1–12.

[3] H.C. Wu, C.C. Chang, Sharing visual multi-secrets using circle shares, Computer Standards & Interfaces 28 (2005) 123–135.

[4] W.P. Fang, J.C. Lin, Visual cryptography with extra ability of hiding confidential data, Journal of Electronic Imaging 15 (2006) 023020.

[5] S.J. Shyu, S.Y. Huang, Y.K. Lee, R.Z. Wang, K. Chen, Sharing multiple secrets in visual cryptography, Pattern Recognition 40 (2007) 3633–3651.

[6] C.N. Yang, New visual secret sharing schemes using probabilistic method, Pattern Recognition Letters 25 (2004) 481–494.

[7] A.D. Bonis, A.D. Santis, Randomness in secret sharing and visual cryptography schemes, Theoretical Computer Science 314 (2004) 351–374.

[8] S.J. Shyu, Image encryption by random grids, Pattern Recognition 40 (2007) 1014–1031.

[9] T.H. Chen, K.H. Tsao, Visual secret sharing by random grids revisited, Pattern Recognition 42 (2009) 2203–2217.

[10] Y.F. Chen, Y.K. Chan, C.C. Huang, M.H. Tsai, Y.P. Chu, A multiple-level visual secret sharing scheme without image size expansion, Information Sciences 177 (2007) 4696–4710.

[11] R. Ito, H. Kuwakado, H. Tanaka, Image size invariant visual cryptography, IEICE Transactions on Fundamentals for Electronics, Communications and Computer Science E82-A (1999) 2172–2177.

[12] R.Z. Wang, S.J. Shyu, Scalable secret image sharing, Signal Processing: Image Communication 22 (2007) 363–373.

[13] W.P. Fang, Friendly progressive visual secret sharing, Pattern Recognition 41 (2008) 1410–1414.

[14] S.J. Lin, J.C. Lin, VCPSS: A two-in-one two-decoding-options image sharing method combining visual cryptography (VC) and polynomial-style sharing (PSS) approaches, Pattern Recognition 40 (2007) 3652–3666.

[15] C.N. Yang, T.S. Chen, Aspect ratio invariant visual secret sharing schemes with minimum pixel expansion, Pattern Recognition Letters 26 (2005) 193–206.

[16] F. Liu, C.K. Wu, X.J. Lin, A new definition of the contrast of visual cryptography scheme, Information Processing Letters 110 (2010) 241–246.