**SCIS 2013** The 30th Symposium on
Cryptography and Information Security
kyoto, Japan, Jan. 22 - 25, 2013
The Institute of Electronics,
Information and Communication Engineers

# An Improved Visual Secret Sharing Scheme with Cheat-Prevention

Shenchuan LIU* Masaaki FUJIYOSHI* Hitoshi KIYA*

**Abstract**— This paper proposes a visual secret sharing (VSS) scheme which prevents malicious share holders from deceiving an honest share holder. A VSS scheme encrypts a secret image into shares so that stacking a sufficient number of shares recovers the secret image where each party keeps a share. A cheat prevention VSS scheme gives another piece to each party for verifying the share presented by another party is genuine. The proposed scheme improves the contrast of the recovered image and cheat-prevention functionality by introducing randomness in producing pieces for verification, whereas a conventional scheme serves recovered images with low contrast or cheat prevention is not well functional in another scheme. Experimental results show the effectiveness of the proposed scheme.

**Keywords:** Cheat prevention, Visual cryptography

## 1 Introduction

In recent years, powerful computers offer even ordinary users the encryption of secret information. To encrypt and decrypt secret information, a key (or a key pair) is used and should be securely and safely guarded. This straightforward key protection, however, still has the risk of key loss and leakage. On the other hand, with the development of fast network technologies, many people collaborate on secret projects over the public Internet. Information should be secret even a few member collude to leak the secret information. To overcome such situations, secret sharing (SS) has been proposed [1].

A SS scheme [1] divides a secret into $n$ pieces referred to as shares. $n$ shares are held by $n$ different parties and the secret is recovered if and only if $k$ or more shares are gathered. This scheme is called as a $(k,n)$-threshold SS scheme. Even computer technology is highly developed, it is not always possible to use computer. In order to overcome these situations, visual SS (VSS) in which decryption can be done by human eyes has been proposed for binary images [2]. Later, it has been extended to non-binary images [3, 4]. Other direction reduces the pixel expansion size and improves the contrast of the recovered image [5, 6].

On the other hand, it is assumed in a scenario that malicious parties deceive a honest party, and cheat-prevention VSS schemes have been proposed to fight it [7–9]. This paper focus on cheat-prevention VSS. A literature [9] found that the original cheat-prevention VSS scheme [8] is not well function in some circumstances. The literature [9] also proposes a new scheme, but pixel expansion is sacrificed signifycantly.

This paper proposes a new cheat-prevention VSS scheme which solves the problem in the original cheat-prevention VSS scheme [8] without sacrificing pixel expansion. By introducing randomness into share generation, it overcomes the above mentioned problems.

The rest of this paper is arranged as follows. In Section 2, SS [1], VSS [2], and the concept of cheat-prevention VSS will be reviewed. Conventional cheat-prevention VSS schemes [8, 9] are introduced in Section 3. The improved scheme is proposed in Section 4. Experimental results are shown in Section 5 and conclusions and future works are given in Section 6.

## 2 Preliminaries

This section briefly describes secret sharing (SS), visual SS (VSS), and cheat-prevention VSS.

### 2.1 Secret Sharing

Here, a $(k,n)$-threshold SS [1] in which $k$ of $n$ shares should be gathered to recover the secret information is described with introducing terms and notations.

Let $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ be the set of $n$ parties, and each party $P_i$ holds share $S_i$ where $i = 1, 2, \ldots, n$. Let $2^{\mathbf{P}}$ be the set of all subsets of $\mathbf{P}$.

Let $\Gamma_{\mathrm{Q}}$ and $\Gamma_{\mathrm{F}}$ be the qualified sets and forbidden sets, respectively. Assume $\Gamma_{\mathrm{Q}}$ is monotone increasing and $\Gamma_{\mathrm{F}}$ is monotone decreasing. Denote $\Gamma^*_{\mathrm{Q}}$ and $\Gamma^*_{\mathrm{F}}$ as the minimum qualified sets and the maximum forbidden sets.

Here, $(\mathbf{P}, \Gamma_{\mathrm{Q}}, \Gamma_{\mathrm{F}})$ is an access structure if $\Gamma_{\mathrm{Q}} \cap \Gamma_{\mathrm{F}} = \emptyset$ and $\Gamma_{\mathrm{Q}} \cup \Gamma_{\mathrm{F}} = 2^{\mathbf{P}}$. Access structure $(\mathbf{P}, \Gamma_{\mathrm{Q}}, \Gamma_{\mathrm{F}})$ for a $(k,n)$-threshold SS method is that $X \in Q$ if and only if $|X| \geq k$, where $|X|$ is the number of parties in $X$.

### 2.2 Visual Secret Sharing

VSS is a kind of secret sharing [2], so the access structure of VSS is the same as SS. There are two differences between VSS and SS; 1) images are used as shares in VSS and 2) decryption needs no computations, it is done by human eyes of watching stacked share images.

* Department of Information and Communication Systems, Tokyo Metropolitan University, 6–6 Asahigaoka, Hino-shi, Tokyo 191–0065, Japan. ( liu-shenchuan@sd.tmu.ac.jp, mfujiyoshi@m.ieice.org, kiya@sd.tmu.ac.jp.)

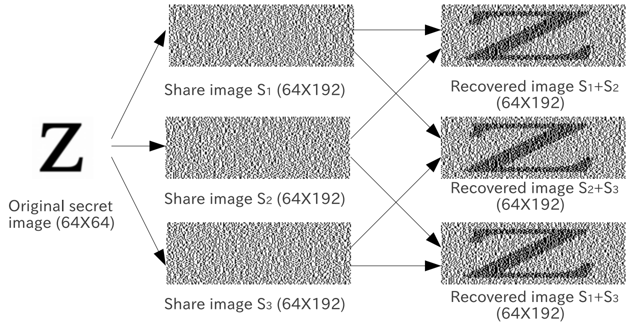

Original secret image (64X64)
Share image S1 (64X192)
Share image S2 (64X192)
Share image S3 (64X192)
Recovered image S1+S2 (64X192)
Recovered image S2+S3 (64X192)
Recovered image S1+S3 (64X192)

Figure 1: An example of the $(2,3)$-threshold visual secret sharing method for binary images [2]. A pixel in the secret image is expanded to three subpixels in a share image, i.e., $m=3$.

Let $\mathbf{S}^0$ and $\mathbf{S}^1$ be the $n\times m$-sized basic matrices for share image generation in a black-and-white VSS method where $\mathbf{S}^0$ and $\mathbf{S}^1$ are for white and black pixels, respectively. For example, in the original $(k,n)$-threshold VSS method [2] in which each party $P_i$ holds share image $S_i$ where $i=1,2,\dots,n$ and a pixel in a secret image is expanded to $m$ subpixels in $S_i$, $\mathbf{S}^0$ and $\mathbf{S}^1$ are given as

$$\mathbf{S}^0=\begin{bmatrix}1&0&0\\1&0&0\\1&0&0\end{bmatrix}\quad\text{and}\quad\mathbf{S}^1=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\qquad(1)$$

under the condition that $n=m=3$ and $k=2$. This $(2,3)$-threshold VSS generates share $S_i$ as

Step 1. For each white pixel in the secret image, put the $i$-th row of $\mathbf{S}^0$ to $S_i$ as $m$-length subpixels.

Step 2. For each black pixel in the secret image, put the $i$-th row of $\mathbf{S}^1$ to $S_i$ as $m$-length subpixels.

Figure 1 shows this $(2,3)$-threshold VSS method [2].

### 2.3 Cheat-Prevention Visual Secret Sharing

In normal VSS methods, two or more parties can collude to generate fake shares. For example, as shown in Eq. (1), subpixels corresponding to a white pixel are $[1\ \ 0\ \ 0]$ regardless of party in the $(2,3)$-threshold VSS with $m=3$. In addition, when two parties collude, $\mathbf{S}^1$ can be easily estimated from subpixels in their shares which subpiexels correspond to black pixels. Now, colluded two parties know $\mathbf{S}^0$ and $\mathbf{S}^1$, they can generate a fake share to deceive the other party. In this scenario, the fake secret image is revealed by stacking the fake shares and share (shares) from honest party (parties). In order to overcome this situation, cheat-prevention VSS have been proposed [8,9].

A successful cheat-prevention VSS prevents any party from deceiving an honest party, and it is summarized [8] that an efficient and robust cheat-prevention scheme should has the following properties:

Req. 1. Each party verifies the shares presented by other participants.

Req. 2. The verification image of each party is different and confidential.

Req. 3. A cheat-prevention scheme should be applicable to any VSS method.

Req. 4. The increase in pixel expansion should be as small as possible.

Req. 5. The contrast of the secret image in the stacking of shares is not significantly reduced.

Req. 6. It does not rely on the help of an on-line trusted authority.

## 3 Conventional Cheat-Prevention VSS Schemes

This section reviews two conventional schemes [8,9] for cheat-prevention VSS. In a cheat-prevention VSS, each of all $n$ parties $\mathbf{P}$ holds two images for satisfying Req. 1; one (share image $S_i$) is for secret recovery and the other (verification image share $V_i$) is for verification.

Before revealing the secret image, party $P_i$ can check others' share image $S_j$ by stacking $V_i$ and $S_j$, i.e., $V_i+S_j$ where $j=1,2,\dots,n$ and $j\neq i$. If this stacked image $(V_i+S_j)$ reveals verification image $V^i$ of $P_i$, which is set in advance, share image $S_j$ is judged to be authentic, if not, $S_j$ is forged. It is noted that verification image $V^i$ of $P_i$ should be know to $P_i$ only.

### 3.1 Conventional Scheme 1

Based on given $\mathbf{S}^0$ and $\mathbf{S}^1$ which are the $n\times m$-sized basic matrices for share image generation in a black-and-white VSS method, this scheme [8] firstly creates four $n\times(m+2)$-sized basic matrices $\mathbf{T}^0$, $\mathbf{T}^1$, $\mathbf{R}^0$, and $\mathbf{R}^1$ as

$$\mathbf{T}^0=\left[\begin{array}{c|c}10&\\\vdots&\mathbf{S}^0\\10&\end{array}\right],\mathbf{T}^1=\left[\begin{array}{c|c}10&\\\vdots&\mathbf{S}^1\\10&\end{array}\right],\qquad(2)$$

$$\mathbf{R}^0=\left[\begin{array}{c|c}10&\\\vdots&0\\10&\end{array}\right],\mathbf{R}^1=\left[\begin{array}{c|c}01&\\\vdots&0\\01&\end{array}\right],\qquad(3)$$

where $\mathbf{T}^0$ and $\mathbf{T}^1$ are used for generating share $S_i$ and $\mathbf{R}^0$ and $\mathbf{R}^1$ are used for generating verification image share $V_i$, respectively.

This scheme generates share image $S_i$ as follows:

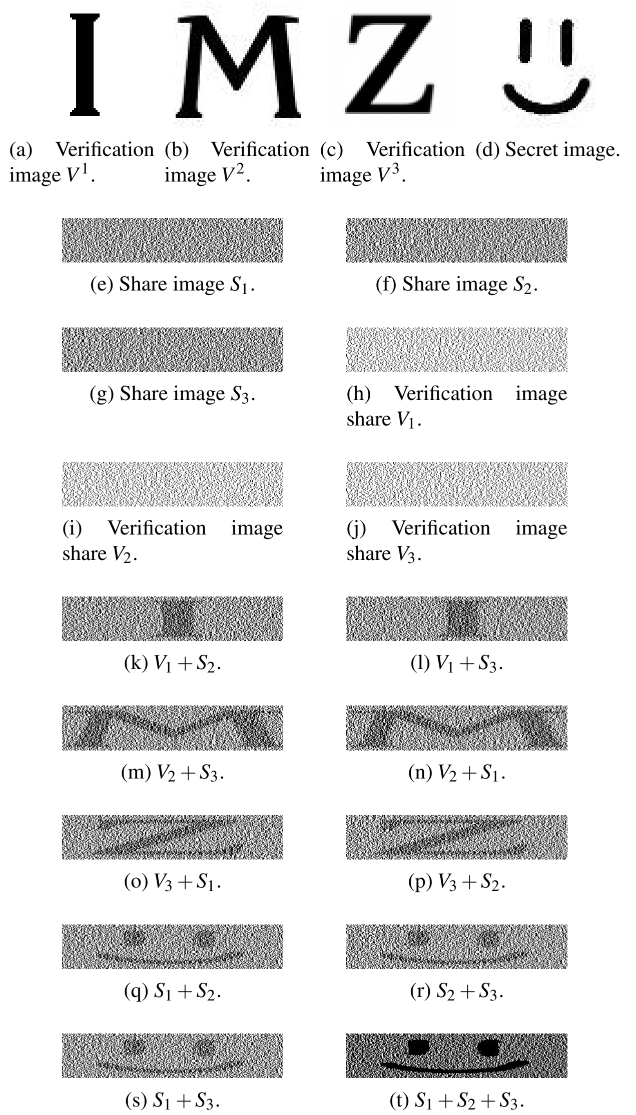
Step 1. For each white pixel in the secret image, put the $i$-th row of $\mathbf{T}^0$ to $S_i$ as $(m+2)$-length subpixels.

Step 2. For each black pixel in the secret image, put the $i$-th row of $\mathbf{T}^1$ to $S_i$ as $(m+2)$-length subpixels.

According to verification image $V^i$ of party $P_i$, this scheme generates verification image share $V_i$ as follows:

Step 1. For each white pixel in $V^i$, put the $i$-th row of $\mathbf{R}^0$ to $V_i$ as $(m+2)$-length subpixels.

Step 2. For each black pixel in $V^i$, put the $i$-th row of $\mathbf{R}^1$ to $V_i$ as $(m+2)$-length subpixles.



(a) Verification image $V^1$. (b) Verification image $V^2$. (c) Verification image $V^3$. (d) Secret image.

(e) Share image $S_1$. (f) Share image $S_2$.

(g) Share image $S_3$. (h) Verification image share $V_1$.

(i) Verification image share $V_2$. (j) Verification image share $V_3$.

(k) $V_1 + S_2$. (l) $V_1 + S_3$.

(m) $V_2 + S_3$. (n) $V_2 + S_1$.

(o) $V_3 + S_1$. (p) $V_3 + S_2$.

(q) $S_1 + S_2$. (r) $S_2 + S_3$.

(s) $S_1 + S_3$. (t) $S_1 + S_2 + S_3$.

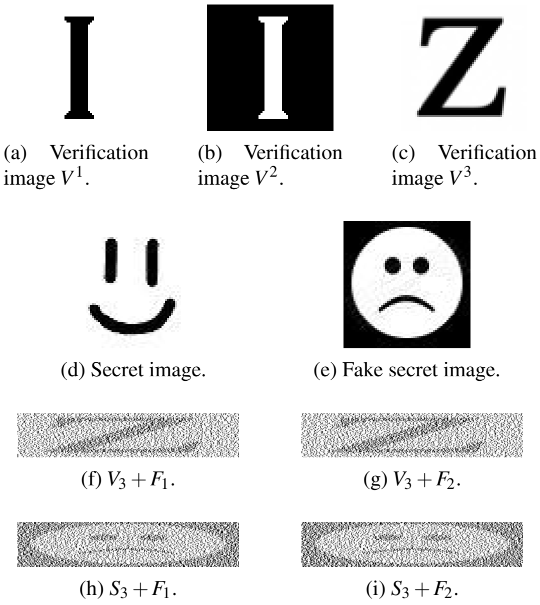
Figure 2: An example of conventional cheat-prevention VSS scheme 1 [8] on $(2,3)$-threshold VSS [2]. Image shares and verification image shares are five times larger in width to verification and secret images, i.e., $m+2=5$.

It is noted that columns in $\mathbf{T}^0$, $\mathbf{T}^1$, $\mathbf{R}^0$, and $\mathbf{R}^1$ are permuted before generating share images to be secure.

Figure 2 shows an example of this cheat-prevention VSS scheme [8] when it is applied to $(2,3)$-threshold VSS [2]. It is shown that party $P_i$ can see his/her own verification image $V^i$ by stacking $V_i$ and $S_j$ where $S_j$ is the image share from party $P_j$. It is also shown that stacking shares shows the secret image.

### 3.2 Conventional Scheme 2

It was found that cheat-prevention in conventional scheme 1 [8] is breakable when adversaries use complementary verification images [9]. As shown in Eq. (3), verification image $V_i$ has subpixels in which the first column is '1' for white pixels and subpixels in which the second column is '1' for black pixels. All other $(m+1)$ columns are zeros. From this fact, if two complementary verification images are used by two malicious parties, they can fool conventional scheme 1. The attack by $(n-1)$ parties in the conventional cheat-prevention VSS scheme 1 on $(k,n)$-threshold VSS method goes as follows:



(a) Verification image $V^1$. (b) Verification image $V^2$. (c) Verification image $V^3$.

(d) Secret image. (e) Fake secret image.

(f) $V_3 + F_1$. (g) $V_3 + F_2$.

(h) $S_3 + F_1$. (i) $S_3 + F_2$.

Figure 3: Attack [9] to conventional scheme 1 [8].

Step 1. Parties $P_i$ and $P_j$ choose complimentary verification images $V^i$ and $V^j$, respectively.

Step 2. Each party receives the image share and verification image share.

Step 3. Determine the positions of the added columns in $\mathbf{R}^0$ and $\mathbf{R}^1$ based on the positions of '1's in $V_i + V_j$.

Step 4. Determine basic matrices $\mathbf{T}^0$ and $\mathbf{T}^1$ based on the reconstructed secret image.

Step 5. Determine the structure of the victim's share image.

Step 6. Create forged share images and verification images.

A tangible example is given here with three parties $P_1$, $P_2$, and $P_3$ in the conventional scheme 1 on $(2,3)$-threshold VSS with Eq. (1). It is assumed that $P_1$ and $P_2$ are collusive cheaters and $P_3$ is the victim. Column permutation of $\mathbf{T}^0$, $\mathbf{T}^1$, $\mathbf{R}^0$, and $\mathbf{R}^1$ are omitted here for simplicity, and the permutation does not prevent $P_1$ and $P_2$ from deceiving $P_3$. The attack is illustrated as follow:

Step 1. $P_1$ and $P_2$ choose complimentary verification images $V^1$ and $V^2$ as shown in Fig. 3 (a) and (b), respectively.

Step 2. Each party receives the share and verification image shares.

Step 3. $P_1$ and $P_2$ stack their verification image shares $V_1$ and $V_2$ to determine the positions of the added columns in $\mathbf{R}^0$ and $\mathbf{R}^1$ by focusing the position of '1.' It is easily determined that the first and second columns are added to zero matrices to form $\mathbf{R}^0$ and $\mathbf{R}^1$.

Step 4. The basic matrices $\mathbf{T}^0$ and $\mathbf{T}^1$ can be uniquely determined because the first and second rows of $\mathbf{T}^0$ and $\mathbf{T}^1$ which for $S_1$ and $S_2$ are known and the positions of the added columns in $\mathbf{T}^0$ and $\mathbf{T}^1$ which are the same as those in $\mathbf{R}^0$ and $\mathbf{R}^1$ are known.

Step 5. Subpixels in $S_3$ are now determined from the third row of $\mathbf{T}^0$ and $\mathbf{T}^1$

Step 6. According to the third row of $\mathbf{T}^0$ and $\mathbf{T}^1$ and the positions of added columns, fake share images $F_1$ and $F_2$ which for $P_1$ and $P_2$, respectively, can be forged.

As shown in Figs. 3 (f) and (g), $P_3$ confirms own verification image $V^3$ (shown in Fig. 3 (c)) from $V_3 + F_1$ and $V_3 + F_2$. However, $S_3 + F_1$ and $S_3 + F_2$ reveal the fake secret image (shown in Fig. 3 (e)) instead of the secret image (shown in Fig. 3 (d)) as shown in Figs. 3 (f) and (g).

The literature [9] has proposed a scheme to solve this problem which the scheme is referred to as conventional scheme 2 in this paper. In order to foil up $u$ collusive cheaters, $u+1$ zero columns and one 1 column, i.e., $u+2$ columns are added to the basic matrices with $m$-columns. This remedy is achieved at the cost of higher pixel expansion which is against Reqs. 4 and 5 as shown in Fig. 4, and the literature also points out this problem.

The next section proposes a new cheat-prevention VSS scheme which overcomes the problem of conventional schemes 1 and 2.

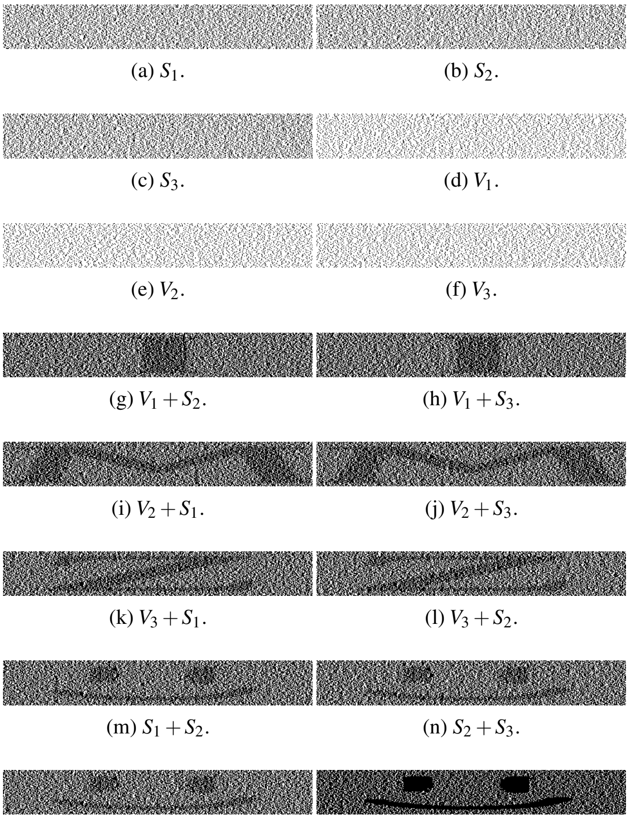

(a) $S_1$. (b) $S_2$. (c) $S_3$. (d) $V_1$. (e) $V_2$. (f) $V_3$. (g) $V_1 + S_2$. (h) $V_1 + S_3$. (i) $V_2 + S_1$. (j) $V_2 + S_3$. (k) $V_3 + S_1$. (l) $V_3 + S_2$. (m) $S_1 + S_2$. (n) $S_2 + S_3$. (o) $S_3 + S_1$. (p) $S_1 + S_2 + S_3$.

Figure 4: An example of conventional scheme 2 [9] on $(2,3)$-threshold VSS method [2] with Eq. (1). Pixels in the secret image are expanded to seven subpixles under the condition that foiling up two collusive parties, i.e., $m+u+2 = 3+2+2 = 7$. The contrast of the recovered secret images are low.

## 4 Proposed Scheme

This section gives details of the proposed scheme. Let $\mathbf{S}^0$ and $\mathbf{S}^1$ be the $n \times m$-sized basic matrices for share generation in a black-and-white VSS method in which each party $P_i$ holds share image $S_i$ where $i = 1, 2, \ldots, n$ and a pixel in a secret image is expanded to $m$ subpixels in a share image.

### 4.1 Algorithm

Firstly, the proposed scheme creates four $n \times (m+2)$-sized basic matrices $\mathbf{T}^0$, $\mathbf{T}^1$, $\mathbf{R}^0$, and $\mathbf{R}^1$ as the same as conventional scheme 1 [8], i.e., as Eqs. (2) and (3). In addition, party-dependent $(m+2)$-length row vector $\mathbf{r}_i^0$ is obtained. Let $\mathbf{t}_i^0$ be the $i$-th row of $\mathbf{T}^0$ where $i = \{1, 2, \ldots, n\}$;

$$\mathbf{t}_i^0 = [1 \;\; 0|\mathbf{s}_i^0], \tag{4}$$

where $\mathbf{s}_i^0$ is the $i$-th row of $\mathbf{S}^0$. With the assumption that the number of 1's in $\mathbf{s}_i^0$ is $k$ where $0 < k < m$, the number of 1's in $\mathbf{t}_i^0$ is $(k+1)$. One 1 is randomly chosen from $(k+1)$ of 1's, and $k$ of 1's are set to zero to obtain new $(m+2)$-length row vector $\mathbf{r}_i^0$ which contains exact one 1.

Then, $\mathbf{T}^0$ and $\mathbf{T}^1$ are used for generating share images $S_i$ as in the conventional schemes [8, 9]. In contrast, according to the pixel value of secret and verification images, verification image share generation can be divided into 4 cases as,

Case 1. Focal pixel in the secret and verification images are black.

Case 2. Focal pixel in the secret and verification images are black and white, respectively.

Case 3. Focal pixel in the secret and verification images are white and black, respectively.

Case 4. Focal pixel in the secret and verification images are white.

Each $(m+2)$-length subpixels in verification image share $V_i$ are generated as follows:

**Cases 1, 2, or 3** Use $\mathbf{R}^0$ and $\mathbf{R}^1$ as in the conventional scheme 1 [8]. That is, put the $i$-th row of $\mathbf{R}^0$ and $\mathbf{R}^1$ to $V_i$ as $(m+2)$-length subpixels, for white and black pixels in verification image $V^i$, respectively.

**Case 4** Put party-dependent row vector $\mathbf{r}_i^0$ to $V_i$ as $(m+2)$-length subpixels.

### 4.2 Example

A tangible example of the proposed scheme is given by using $(2,3)$-threshold VSS method with Eq. (1). Then, $\mathbf{T}^0$ is given as

$$\mathbf{T}^0 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5}$$

Thus, party-dependent row vector $\mathbf{r}_i^0$ is randomly defined as

$$\mathbf{r}_i^0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{or} \quad \mathbf{r}_i^0 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{6}$$

So, it is impossible to identify the position of added columns by stacking verification image shares of collusive cheaters.

As the key point of the attack to conventional scheme 1 [8] is to determine the position of added columns [9], the proposed scheme introduces randomness in generating verification image shares which makes it impossible to do so.

### 4.3 Features

The features of the proposed scheme are summarized here.

#### 4.3.1 Cheat-Prevention Functionality Improvement

The problem of conventional scheme 1 [8] is due to all rows in $\mathbf{R}^0$ are the same and simultaneously all rows in $\mathbf{R}^1$ are the same. That is, all parties receives verification image share $V_i$'s in which subpixels corresponding to black pixels in verification image $V^i$ are the same regardless of party and simultaneously subpixels corresponding to white pixels in $V^i$ are the same regardless of party. This fact allows malicious parties to collude for deceiving a honest party by using complementary verification images. On the other hand, the proposed scheme introduces party-dependent subpixels to $V_i$'s. This simple strategy prevents malicious parties from estimating $\mathbf{R}^0$ and $\mathbf{R}^1$, i.e., from deceiving a honest party.

#### 4.3.2 Improvement in Pixel Expansion and Contrast of Recovered Images

The problem of conventional scheme 2 [9] is due to increasing zero columns to prevent malicious parties from deceiving a honest party, c.f., Figs. 2 and 4. On the other hand, the proposed scheme simply introduces randomness to the share generation process. The proposed scheme, thus, keeps the subpixel size and it results in keep the contrast of the recovered secret image.

## 5 Experimental Results

The proposed scheme is implemented on $(2,3)$-threshold VSS method in this experiment in which parties $P_1$ and $P_2$ try to deceive party $P_3$. Verification images for $P_1$, $P_2$, and $P_3$ are the images shown in Figs. 3 (a), (b), and (c), respectively. The secret and fake secret images are those shown in Figs. 3 (d) and (e), respectively.

Figures 5 (a), (b), and (c) show secret image shares $S_1$, $S_2$, and $S_3$, respectively, and Figs. 5 (d), (e), and (f) are verification image shares $V_1$, $V_2$, and $V_3$, respectively. Figures 5 (g) and (h) are forged secret shares for $P_1$ and $P_2$.

As shown in Figs. 5 (i) and (j), as the proposed scheme only introduces randomness in one case of four cases, so the shape of 'Z' is slightly remained. But a clear verification image 'Z' is not revealed, so the proposed scheme is efficient. Furthermore, Figs. 5 (k) and (l) show that stacking fake and real shares reveals a unclear fake image, whereas Figs. 5 (m), (n), (o), and (p) show that stacking real shares recovers clear secret images.

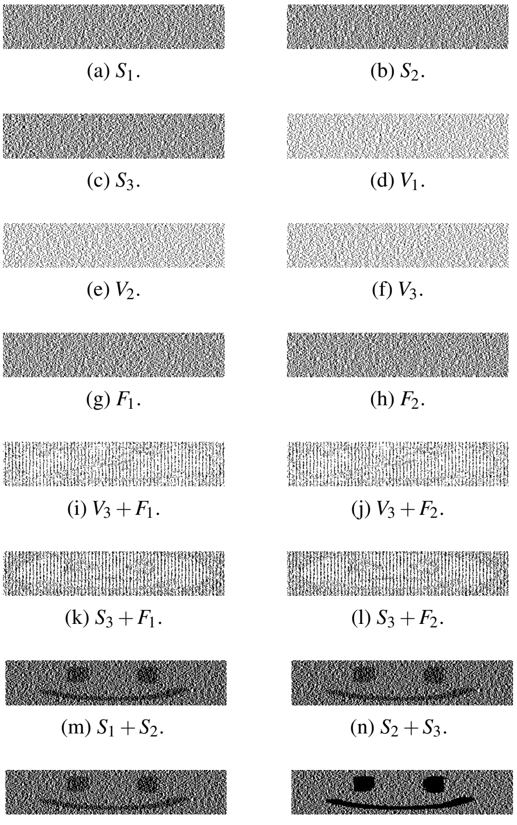

(a) $S_1$. (b) $S_2$. (c) $S_3$. (d) $V_1$. (e) $V_2$. (f) $V_3$. (g) $F_1$. (h) $F_2$. (i) $V_3+F_1$. (j) $V_3+F_2$. (k) $S_3+F_1$. (l) $S_3+F_2$. (m) $S_1+S_2$. (n) $S_2+S_3$. (o) $S_1+S_3$. (p) $S_1+S_2+S_3$.

Figure 5: An example of the proposed scheme on $(2,3)$-threshold VSS method [2].

## 6 Conclusions

This paper has improved the visual secret sharing scheme with cheat-prevention. The proposed scheme has better performance than conventional scheme 1 [8] in cheat-prevention functionality and less pixel expansion than conventional scheme 2 [9]. Experimental results show the effectiveness of the proposed scheme.

Future works include introducing randomness in all four cases and minimizing the pixel expansion.

## References

[1] A. Shamir, "How to share a secret," *Commun. ACM*, vol.22, pp.612–613, Nov. 1979.

[2] M. Naor and A. Shamir, "Visual cryptography," in *Proc. IACR EUROCRYPT*, LNCS, vol.950, pp.1–12, 1994.

[3] I. Biehl and S. Wetzel, "Traceable visual cryptography," in *Proc. Int. Conf. Information Communication Security*, LNCS, vol.1334, pp.61–71, 1997.

[4] T. Hofmeister, M. Krause, and H.-U. Simon, "Contrast-optimal *k* out of *n* secret sharing schemes in visual cryptography," *Theoret. Comput. Sci.*, vol.240, no.2, pp. 471–485, 2000.

[5] S. Cimato, A. De Santis, A.L. Ferrara, and B. Masucci, "Ideal contrast visual cryptography schemes with re-

versing," *Inf. Process. Lett.*, vol.93, no.4, pp.199–206, 2005.

[6] P.A. Eisen and D.R. Stinson, "Threshold visual cryptography with specified whiteness levels of reconstructed pixels," *Designs, Codes, Cryptog.*, vol.25, no.1, pp.15–61, 2002.

[7] D.S. Tsai, T.H. Chen, and G. Horng, "A cheating prevention scheme for binary visual cryptography with homogeneous secret images," *Pattern Recog.*, vol.40, no.8, pp. 2356–2366, Aug. 2007.

[8] C.M. Hu and W.G. Tzeng, "Cheating prevention in visual cryptography," *IEEE Trans. Image Process.*, vol.16, no.1, pp. 36–45, Jan. 2007.

[9] Y.C. Chen, G. Horng, and D.S Tsai, "Comment on 'Cheating Prevention in Visual Cryptography " *IEEE Trans. Image Process.*, vol.21, no.7, pp.–, July 2012.