Open access • Journal Article • DOI:10.1109/TIP.2006.884916

Cheating Prevention in Visual Cryptography — Source link

Chih-Ming Hu, Wen-Guey Tzeng

Institutions: National Chiao Tung University

Published on: 01 Jan 2007 - IEEE Transactions on Image Processing (Institute of Electrical and Electronics Engineers Inc.)

Topics: Secret sharing, Visual cryptography and Cheating

Cheating Prevention in Visual Cryptography

Chih-Ming Hu and Wen-Guey Tzeng

***Abstract*—Visual cryptography (VC) is a method of encrypting a secret image into shares such that stacking a sufficient number of shares reveals the secret image. Shares are usually presented in transparencies. Each participant holds a transparency. Most of the previous research work on VC focuses on improving two parameters: pixel expansion and contrast. In this paper, we studied the cheating problem in VC and extended VC. We considered the attacks of malicious adversaries who may deviate from the scheme in any way. We presented three cheating methods and applied them on attacking existent VC or extended VC schemes. We improved one cheat-preventing scheme. We proposed a generic method that converts a VCS to another VCS that has the property of cheating prevention. The overhead of the conversion is near optimal in both contrast degression and pixel expansion.**

***Index Terms*—Cheat-preventing, cheating, secret sharing, visual cryptography.**

I. Introduction

EVEN with the remarkable advance of computer technology, using a computer to decrypt secrets is infeasible in some situations. For example, a security guard checks the badge of an employee or a secret agent recovers an urgent secret at some place where no electronic devices are available. In these situations the human visual system is one of the most convenient and reliable tools to do checking and secret recovery. Therefore, Naor and Shamir [19] invented the visual cryptography (VC) in which a secret image (printed text, picture, etc.) is encrypted in a perfectly secure way such that the secret can be decoded directly by the human visual system.

VC is a method of encrypting a *secret image* into *shares* such that stacking a sufficient number of shares reveals the secret image. Shares are usually presented in transparencies. Each participant holds a transparency (share). Unlike conventional cryptographic methods, VC needs no complicated computation for recovering the secret. The act of decryption is to stack shares and view the image that appears on the stacked shares simply. A (k, n)-visual cryptography scheme [denoted as (k, n)-VCS] is a visual secret sharing scheme such that stacking any k or more shares reveals the secret image, but stacking fewer than k shares reveals not any information about the secret image.

VC has been studied intensively since the pioneer work of Noar and Shamir [19]. Most of the previous research work on VC focused on improving two parameters: *pixel expansion* and *contrast* [5], [6], [8], [11], [13], [16], [23]. In these cases, all participants who hold shares are assumed to be semi-honest, that is, they will not present *false* or *fake shares* during the phase of recovering the secret image. Thus, the image shown on the stacking of shares is considered as the *real secret image*. Nevertheless, cryptography is supposed to guarantee security even under the attack of malicious adversaries who may deviate from the scheme in any way. We have seen that it is possible to cheat [12], [14], [18], [26] in VC, though it seems hard to imagine. For cheating, a cheater presents some fake shares such that the stacking of fake and genuine shares together reveals a fake image. With the property of unconditional security, VC is suitable for sending highly classified orders to a secret agent when computing devices may not be available. The secret agent carried some shares, each with a pre-determined order, when departing to the hostile country. When the headquarter decides to execute a specific order, it can simply send another share to the agent so that the agent can recover what the order is. We can see that it would be terrible if the dispatched share cannot be verified due to a cheater's attack.

A VCS would be helpful if the shares are meaningful or identifiable to every participant. A VCS with this extended characteristic is called extended VCS (EVCS) [2], [19]. A (k, n)-EVCS is like a (k, n)-VCS except that each share displays a meaningful image, which will be called *share image* hereafter. Different shares may have different share images. At first glance, it seems very difficult to cheat in EVCS because the cheater does not know the share images that appear on the genuine shares and, thus, has no information about the distributions of black and white pixels of the share images. This information is crucial for cheating in VC. In this paper we show that it is still possible to cheat in EVC.

A. Our Contributions

In this paper, we study the cheating problem in VC and EVC. We present three cheating methods and apply them on existent VC or EVC schemes. Although, the revealed secret is an image, our attacks are not like the attacks against watermarks, such as the Stirmark attack, which makes watermarks undetectable. Our attacks are to reveal fake images to cheat honest participants. Our attacks are more like the man-in-the-middle attack in cryptography. In fact, our attacks are very general for all kinds of VCSs without cheating-prevention mechamism.

We propose a generic method that converts a VCS to another VCS that has the property of cheating prevention (also called *cheat-preventing* VCS). The overhead of the conversion is near optimal. Our contributions are summarized as follows.

1) We propose three cheating methods against VC or EVC schemes. The first two methods are applied to attack

Manuscript received September 29, 2005; revised May 10, 2006. This work was supported in part by the National Science Council of Taiwan under Grant NSC-94-2213-E-009-110. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ercan E. Kuruoglu.

The authors are with the Department of Computer Science, National Chiao-Tung University, Hsinchu, Taiwan 30050, R.O.C. (e-mail: andyhu@cs.nctu.edu.tw; wgtzeng@cs.nctu.edu.tw).

Digital Object Identifier 10.1109/TIP.2006.884916

VC schemes and the third one is applied to attack EVC schemes. These three methods are easy to implement and satisfy the cheating definition for cheating traditional secret sharing schemes.

2) We review some previously proposed cheat-preventing VC or EVC schemes and demonstrate that those schemes are either not robust enough (still cheatable) or improvable.
3) We propose some necessary criteria for a VCS to be secure against cheating robustly. By these criteria, we propose a generic method that converts any VCS to another VCS with the property of cheating prevention. Our conversion is very efficient and incurs little overhead compared with the original VCS. The degression in contrast of the converted VCS is almost optimal. For each pixel of the secret image, we add two additional subpixels to the encoded subpixels only, no matter how many the encoded subpixels are.

The rest of this paper is organized as follows. The models for VCS, EVCS, and cheating behaviors in VC are discussed in Section II. We then describe three cheating methods, each with its distinct cheating behavior in Section III. In Section IV, we survey some previous cheat-preventing schemes and show how to attack and improve them. Finally, we present an efficient and generic transformation from any VCS to a cheat-preventing VCS that is measurably better than all previous schemes in Section V.

B. Previous Work

Naor and Shamir [19] proposed a (k, n)-VCS. Many improvements and extensions follows [1]–[3], [5], [6], [8], [10], [11], [13], [15]–[17], [22]–[25]. For example, Ateniese *et al.* [1] proposed an elegant VCS for general access structures based on the cumulative array method. Tzeng and Hu [22] proposed a new definition for VC, in which the secret image can be either darker or lighter than the background.

Naor and Pinkas [18] showed some methods of authentication and identification for VC. Their scenario focuses on authentication and identification between two participants. Yang and Laih [26] proposed two cheat-preventing methods. Their first method needs an on-line TA (Trusted Authority) to verify the shares of participants. Their second method is a transformation from a VCS (but not a $(2, n)$-VCS) to a cheat-preventing VCS on which the stacking of two shares reveals the verification image. The method needs to add extra $O(n^2)$ subpixels for each pixel in the secret image.

Horng *et al.* [14] proposed a cheating method against some VC schemes. In their cheating method, the cheater needs to know the exact distribution of black and white subpixels of the shares of honest participants. Based on this characteristic, they proposed a cheat-preventing method to prevent the cheater from obtaining the distribution. However, we show that the knowledge of the distribution is not a necessary condition for a successful cheat. They also proposed another cheat-preventing method in which the stacking of the genuine share and verification share reveals the verification image in some small region. We show that it is possible to attack the method.

II. Preliminaries

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be the set of n participants. Each P_i holds a share S_i, $1 \le i \le n$. Let $2^{\mathcal{P}}$ be the set of all subsets of $\mathcal{P}$. A set $X \subseteq \mathcal{P}$ is called a *qualified set* if the stacking of the shares of the participants in X reveals the secret image. A set $Y \subseteq \mathcal{P}$ is called a *forbidden set* if the stacking of their shares reveals no information about the secret image. $\Gamma = (\mathcal{P}, Q, F)$ is an *access structure* if $Q \cap F = \emptyset$ and $Q \cup F = 2^{\mathcal{P}}$. The access structure $\Gamma = (\mathcal{P}, Q, F)$ for (k, n)-secret sharing is that $X \in Q$ if and only if $|X| \ge k$, where $|X|$ is the number of participants in X.

A. Visual Cryptography Scheme

The secret image consists of a collection of *black* and *white pixels*. To construct n shares of an image for n participants, we need to prepare two collections, C_0 and C_1, which consist of $n \times m$ Boolean matrices. A row in a matrix in C_0 and C_1 corresponds to m subpixels of a pixel, where 0 denotes the white subpixel and 1 denotes the black subpixel. For a white (or black) pixel in the image, we randomly choose a matrix M from C_0 (or C_1, respectively) and assign row i of M to the corresponding position of share $S_i, 1 \le i \le n$. Each pixel of the original image will be encoded into n pixels, each of which consists of m subpixels on each share. Since a matrix in C_0 and C_1 constitutes only one pixel for each share. For security, the number of matrices in C_0 and C_1 must be huge. For succinct description and easier realization of the VC construction, we do not construct C_0 and C_1 directly. Instead, we construct two $n \times m$ *basis matrices* S^0 and S^1 and then let C_0 and C_1 be the set of all matrices obtained by permuting columns of S^0 and S^1, respectively.

Let $\mathrm{OR}(B, X)$ be the vector of "bitwise-OR" of rows $i_1, i_2, \ldots, i_q$ of B, where B is an $n \times m$ Boolean matrix and $X = \{P_{i_1}, P_{i_2}, \ldots, P_{i_q}\}$ is a set of participants. Let $w(v)$ be the Hamming weight of row vector v. For brevity, we let $w(B, X) = w(OR(B, X))$. Let $p_b(S) = w(v)/m$, where v is a black pixel in share S and m is the dimension of v. Similarly, $p_w(S) = w(v)/m$, where v is a white pixel in share S. Note that all white (or black) pixels in a share have the same Hamming weight. We use "$S_i + S_j$" to denote "the stacking of shares S_i and S_j." The "stacking" corresponds to the bitwise-OR operation "+" of subpixels in shares S_i and S_j.

The definition of VC [1] for an access structure is as follows.

Definition 2.1: Let $\Gamma = (\mathcal{P}, Q, F)$ be an access structure. Two collections *(multisets)* C_0 and C_1 of $n \times m$ Boolean matrices constitute a (Γ, m)-VCS if there exist a value $\alpha(m) > 0$ and a set $\{(X, t_X)\}_{X \in Q}$ satisfying the following.

1) Any qualified set $X = \{P_{i_1}, P_{i_2}, \ldots, P_{i_q}\} \in Q$ can recover the secret image by stacking their shares. Formally, for any $M \in C_0, w(M, X) \le t_X - \alpha(m) \times m$, whereas, for any $M' \in C_1, w(M', X) \ge t_X$.
2) Any forbidden set $Y = \{P_{i_1}, P_{i_2}, \ldots, P_{i_q}\} \in F$ has no information on the secret image. Formally, the two collections $C_t, t \in \{0, 1\}$, of $q \times m$ matrices obtained by restricting each $n \times m$ matrix in $M \in C_t$ to rows $i_1, i_2, \ldots, i_q$, are indistinguishable in the sense that they contain the same matrices with the same frequencies.

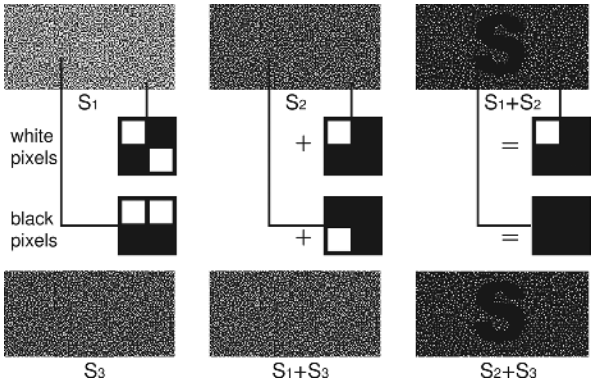

Fig. 1. $(\Gamma, 4)$-VCS and the structures of subpixels.

The value m is called *pixel expansion*, which is the number of subpixels that each pixel of the secret image is encoded into in each share. The value $\alpha(m) \geq 0$ is called *contrast*. The higher the contrast, the more visible by human eyes the secret image. The first property (contrast) ensures that the recovered image shows difference between the white pixels and the black pixels. The second property (security) ensures that nothing about the image can be recovered from the shares of participants in a forbidden set.

The following shows an example of VC.

1) Example 2.1: Let $\mathcal{P} = \{P_1, P_2, P_3\}$, $Q = \{(P_1, P_2), (P_2, P_3), (P_1, P_2, P_3)\}$ and then $F = \{(P_1), (P_2), (P_3), (P_1, P_3), ()\}$. The two basis matrices

$$S^0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} \text{ and } S^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

form a $(\Gamma, 4)$-VCS with contrast $\alpha(m) = 1/4$. The shares S_1, S_2, and S_3, and the stackings of them are given in Fig. 1.

In the above example, each pixel of the secret image is encoded as four subpixels in each share. To encode a white (or black) pixel, we assign row i of S^0 (or S^1, respectively) to share S_i, $1 \leq i \leq n$. In order to ensure security, the order of the subpixels of a pixel is randomly permuted (simultaneously permuted for all shares). This is equivalent to randomly choosing a matrix M from C_0 (or C_1, respectively).

An extended VCS is a VCS such that each share has a meaningful share image.

2) Example 2.2: Fig. 2 shows an EVCS for the access structure Γ of Example 2.1. The share images of S_1, S_2, and S_3 are $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively. Note that $S_1 + S_3$ shows no information about the secret $\boldsymbol{S}$.

B. Cheating in VC

There are two types of cheaters in our scenario. One is a malicious participant ($\mathbf{MP}$) who is also a *legitimate participant*, namely, $\mathbf{MP} \in \mathcal{P}$, and the other is a malicious outsider ($\mathbf{MO}$), where $\mathbf{MO} \notin \mathcal{P}$. In this paper, we show that not only an $\mathbf{MP}$ can cheat, but also an $\mathbf{MO}$ can cheat under some circumstances.

A cheating process against a VCS consists of the following two phases:

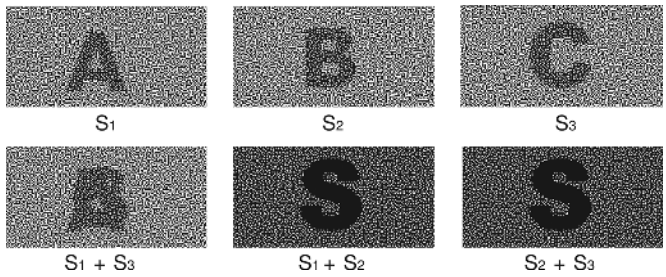

Fig. 2. $(\Gamma, 4)$-EVCS.

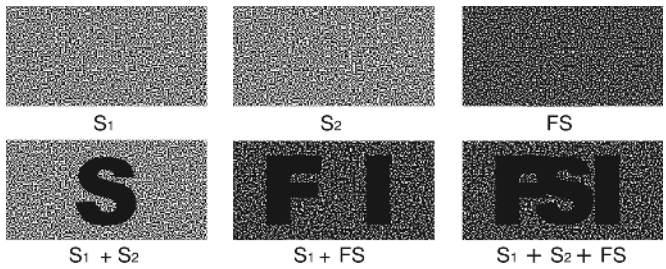

Fig. 3. Example of cheating a (2,2)-VCS.

1) fake share construction phase: the cheater generates the fake shares;
2) image reconstruction phase: the fake image appears on the stacking of genuine shares and fake shares.

In order to cheat successfully, honest participants who present their shares for recovering the secret image should not be able to distinguish fake shares from genuine shares. A reconstructed image is *perfect black* if the subpixels associated to a black pixel of the secret image are all black. Most proposed VC schemes have the property of perfect blackness. For example, the reconstructed secret images $\mathbf{S}$ in Example 2.1 are all perfectly black.

We only consider to cheat the participants who together do not constitute a qualified set. Since all participants together in a qualified set can recover the real secret image in perfect blackness already, it is not possible to cheat them.

1) Example 2.3: Fig. 3 shows how to cheat participants in a (2,2)-VCS. Since $S_1 + FS$ reveals the fake image $\mathbf{FI}$, P_1 is cheated to believe that the secret image is $\mathbf{FI}$. Although $S_1 + S_2 + FS$ successfully reveals the fake image, the real secret image $\mathbf{S}$ also appears on $S_1 + S_2 + FS$ due to the property of perfect blackness for secret images. The participants of a qualified set (P_1, P_2), in this example, cannot be cheated.

A successful cheat against a VCS is defined as follows. By the general practice for security analysis, the cheater is required to succeed with a significant probability only.

Definition 2.2: For a (Γ, m)-VCS with basis matrices S^0 and S^1, an $\mathbf{MP}$ or an $\mathbf{MO}$ cheats successfully if it finds a fake image and generates fake shares satisfying the following.

1) For $Y = \{P_{i_1}, P_{i_2}, \ldots, P_{i_q}\} \notin Q$, the stacking of their shares and the fake shares reveals the fake image. If the cheater is an $\mathbf{MP}$, some P_{i_j} is the cheater, $1 \leq j \leq q$.
2) The fake shares cannot be distinguished from the genuine shares. Formally, for each fake share (FS), there is a share

Input: share S_1. (Wlog, we assume that the cheater is P_1)
Fake share construction phase:
Assume that each pixel of S_1 has x black and y white subpixels. Then, P_1 chooses a fake image and prepares $r = \lceil \frac{m}{x} \rceil - 1$ fake shares $FS_1, FS_2, \ldots, FS_r$ as follows:

1) For each white pixel of the fake image, copy the corresponding subpixels of the pixel in S_1 to each fake share.
2) For each black pixel of the fake image, randomly assign x black and y white subpixels to each fake share such that the pixel in the stacking of these fake shares and S_1 is perfect black.

Image reconstruction phase (the fake image):
Let $Y = \{P_1, P_{i_1}, P_{i_2}, \ldots, P_{i_q}\}$ be a set of participants. If $Y \notin Q$, the stacking of genuine shares $S_1, S_{i_1}, S_{i_2}, \ldots, S_{i_q}$ and fake shares $FS_1, FS_2, \ldots, FS_r$ shall reveal the fake image.

Fig. 4. Cheating method CA-1, initiated by an $\mathbf{MP}$.

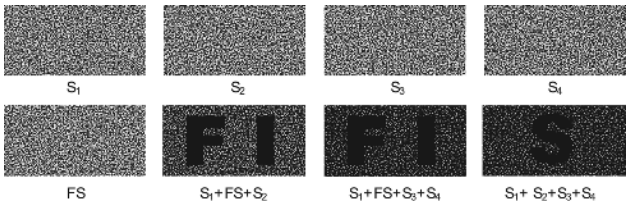

Fig. 5. Example of cheating a (4,4)-VCS by an MP.

S_i such that the subpixels of FS are identically distributed as those of S_i.

III. Three Cheating Methods

Our first cheating method is initiated by an $\mathbf{MP}$, while the second cheating method is initiated by an $\mathbf{MO}$. Both of them apply to attack VC. Our third cheating method is initiated by an $\mathbf{MP}$ and applies to attack EVC.

A. Cheating a VCS by an $\mathbf{MP}$

The cheating method **CA-1**, depicted in Fig. 4, applies to attack any VCS. Without loss of generality, we assume that P_1 is the cheater. Since the cheater is an $\mathbf{MP}$, he uses his genuine share as a template to construct a set of fake shares which are indistinguishable from its genuine share. The stacking of these fake shares and S_1 reveals the fake image of perfect blackness. We see that, for $Y = \{P_1, P_{i_1}, P_{i_2}, \ldots, P_{i_q}\} \notin Q$, the stacking of their shares reveals no images. Thus, the stacking of their shares and the fake shares reveals the fake image due to the perfect blackness of the fake image.

1) Example 3.1: Fig. 5 shows how to cheat the participants in a (4,4)-VCS. There are four shares S_1, S_2, S_3, and S_4 in the (4,4)-VCS. P_1 is assumed to be the $\mathbf{MP}$. By ***CA-1***, one fake share FS_1 is generated. Since $Y = (P_1, P_3, P_4)$ (or (P_1, P_2)) $\notin Q$, we see that $S_1 + FS_1 + S_3 + S_4$ (or $S_1 + FS_1 + S_2$) reveals the fake image $\mathbf{FI}$. Thus, P_3 and P_4 (or P_2) are cheated to believe that $\mathbf{FI}$ is the secret image.

Input: none.
Fake share construction phase:
The $\mathbf{MO}$ chooses a fake image and does the following:

1) Encode the fake image into two fake shares FS_1 and FS_2 with the optimal $(2, 2)$-VCS.
2) Generate enough pairs of fake shares $FS_{1,i}$ and $FS_{2,i}$ with various sizes and subpixel distributions, $1 \le i \le r$ for some r.

Image reconstruction phase (the fake image):
Let $Y = \{P_{i_1}, P_{i_2}, \ldots, P_{i_q}\} \notin Q$. The stacking of $S_{i_1}, S_{i_2}, \ldots, S_{i_q}$ and two fake shares $FS_{1,c}$ and $FS_{2,c}$ shows the fake image for some c, $1 \le c \le r$.

Fig. 6. Cheating method CA-2, initiated by an MO.

For some prominent (n, n)- and (k, n)-VCSs [5], [6], [19], the numbers of black and white subpixels in a pixel are almost equal. The cheater needs only $r = \lceil m/x \rceil - 1 = 1$ fake share to cheat successfully.

Theorem 3.1: The $\mathbf{MP}$ in **CA-1** successfully cheats any VCS.

Proof: Contrast. Let S^0 and S^1 be the basis matrices of a VCS and the pixel expansion is m. For, $Y = \{P_1, P_{i_1}, P_{i_2}, \ldots, P_{i_q}\} \notin Q$, $w(S^0, Y) = w(S^1, Y)$. By the construction of **CA-1**, for a white pixel of the fake image, the weight of the OR-vector of $OR(S^0, Y)$ and the fake shares is equal to $w(S^0, Y) = t_Y - \alpha(m) \times m < m$. For a black pixel of the fake image, the weight of the OR-vector of $OR(S^1, Y)$ and the fake shares is equal to m. Thus, the contrast property is satisfied and the fake image appears.

Indistinguishability. The fake shares are generated according to S_1. Each pixel in the fake shares has the same number of white and black subpixels as those in S_1. Also, those subpixels are randomly distributed for each fake share. Thus, the fake shares are indistinguishable from S_1. □

B. Cheating a VCS by an $\mathbf{MO}$

Our second cheating method **CA-2**, depicted in Fig. 6, demonstrates that an $\mathbf{MO}$ can cheat even without any genuine share at hand. The idea is as follows. We use the optimal (2,2)-VCS to construct the fake shares for the fake image. Then, we tune the size of fake shares so that they can be stacked with genuine shares.

Now, the only problem is to have the right share size for the fake shares. Our solution is to try all possible share sizes. In the case that the $\mathbf{MO}$ gets one genuine share, there will be no such problem. It may seem difficult to have fake shares of the same size as that of the genuine shares. We give a reason to show the possibility. The shares of a VCS are usually printed in transparencies. We assume that this is done by a standard printer or copier which accepts only a few standard sizes, such as A4, A3, etc. Therefore, the size of genuine shares is a fraction, such as 1/4, of a standard size. We can simply have the fake shares of these sizes. Furthermore, it was suggested to have a solid frame to align shares [19] in order to solve the alignment problem during the image reconstruction phase. The $\mathbf{MO}$ can simply choose the size of the solid frame for the fake shares.

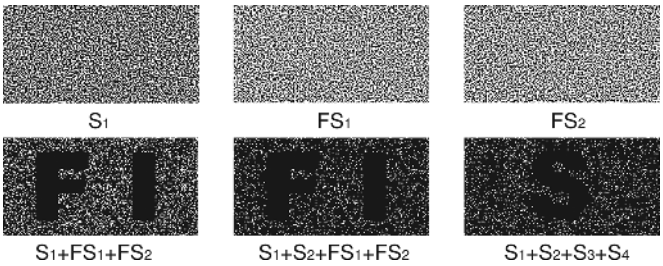

Fig. 7. Example of cheating a (4,4)-VCS by an **MO**.

Therefore, it is possible for the **MO** to have the right size for the fake shares.

1) Example 3.2: Fig. 7 shows that an ***MO*** cheats a (4,4)-VCS. The four genuine shares S_1, S_2, S_3, and S_4 are those in Fig. 5 and the two fake shares are FS_1 and FS_2. For clarity, we put S_1 here to demonstrate that the fake shares are indistinguishable from the genuine shares. We see that the stacking of fewer than four genuine shares and two fake shares shows the fake image **FI**.

Theorem 3.2: The **MO** in ***CA-2*** successfully cheats a VCS if the right share size is obtained.

Proof: Contrast. For $Y = \{P_{i_1}, P_{i_2}, \ldots, P_{i_q}\} \notin Q$, let $Z_Y = S_{i_1} + S_{i_2} + \cdots + S_{i_q}$. Since FS_1 and FS_2 are two shares of the optimal $(2,2)$-VCS, $p_b(FS_1 + FS_2) = 1$ and $p_w(FS_1 + FS_2) = 1/2$. By **CA-2**, the distribution of subpixels of the genuine shares are random and independent of that of the fake shares. For the white pixel in $Z_Y + FS_1 + FS_2$, we have, with high probability

$$p_w(Z_Y + FS_1 + FS_2) = 1 - (1 - p_w(Z_Y))\left(1 - \frac{1}{2}\right) = \frac{1}{2} + \frac{p_w(Z_Y)}{2} < 1.$$

Also, due to the perfect black property in recovering the fake image, we have $p_b(Z_Y + FS_1 + FS_2) = 1$. Thus, the contrast property in $Z_Y + FS_1 + FS_2$ is satisfied and the fake image appears.

Indistinguishability. We assume that the size of $FS_{1,c}$ and $FS_{2,c}$ is correct. By the construction of **CA-2**, the fake shares are indistinguishable from the genuine ones. □

C. Cheating an EVCS by an MP

In the definition of VC, it only requires the contrast be nonzero. Nevertheless, we observe that if the contrast is too small, it is hard to "see" the image. Based upon this observation, we demonstrate the third cheating method **CA-3**, depicted in Fig. 8, against an EVCS. The idea of **CA-3** is to use the fake shares to reduce the contrast between the share images and the background. Simultaneously, the fake image in the stacking of fake shares has enough contrast against the background since the fake image is recovered in perfect blackness.

Let ϵ be the threshold for contrast that human eyes distinguish the image from the background. The value ϵ varies for different

Input: share S_1. (Wlog, we assume that the cheater is P_1.)
Fake share construction phase:
P_1 chooses a fake image and does the following:

1) Create S_1', which is S_1, but without the share image. The share image of S_1 is removed by changing d black subpixels into white subpixels in each black pixel, where d is the difference between the numbers of black subpixels of a black and a white pixel.
2) Create $r = \lceil \frac{m}{x} \rceil - 1$ temporary fake shares FS_i', $1 \le i \le r$, by using S_1' according to **CA-1**.
3) Randomly change d white subpixels into black subpixels of each pixel of the share image in FS_i', $1 \le i \le r$.
4) Construct FS_i by randomly adding ϵm black subpixels (changing from white subpixels) to each pixel in FS_i', $1 \le i \le r$. The threshold value ϵm, like those in Table I, is obtained by experiments.

Image reconstruction phase (the fake image):
Same as in **CA-1**.

Fig. 8. Cheating method CA-3 against an EVCS.

sizes, contrasts and types of share images. We conduct some experiments to obtain ϵ empirically. We consider four types of pictures (in Fig. 9) with four different sizes (Z_1: 200 × 100 pixels, Z_2: 200 × 200 pixels, Z_3: 400 × 200 pixels, and Z_4: 400 × 400 pixels) and four different contrasts (1/4, 1/9, 1/16, and 1/25). The values (ϵm) in Table I represent the number of black subpixels which we should add for each pixel of the fake shares in order to reduce the contrast between the background and the share images to be less than ϵ. The larger the size and contrast of the image are, the more black subpixels we need to add to the fake shares. Most EVCSs do not have a large contrast, we can easily cheat them by adding a small number of black subpixels to the pixels of the share images in the fake shares.

1) Example 3.3: Fig. 10 shows the results of cheating a (Γ, m)-EVCS, where $\mathcal{P} = \{P_1, P_2, P_3\}$, and $Q = \{(P_1, P_2), (P_2, P_3), (P_1, P_2, P_3)\}$. In this example, P_1 is the cheater who constructs a fake share FS_2 with share image **B** in substitute for P_2 to cheat P_3. $S_1 + FS_2 + S_3$ reveals the fake image **FI**.

Theorem 3.3: The **MP** in **CA-3** successfully cheats an EVCS by producing fake shares with meaningful share images if the ϵ is correct.

Proof: By Step 3 in **CA-3**, the share image appears on the fake share.

Contrast. Since the fake shares are constructed by the same way of **CA-1**, the recovered fake image in perfect blackness appears on the stacking of shares. Furthermore, the share images of the fake shares are invisible since we have added an enough number of black subpixels to blur them.

Indistinguishability. The proof is the same as that of Theorem 3.1 except that we have to show that honest participants cannot identify fake shares. Since share images are used for identification, honest participants will not know the exact shapes of share images. They care only about the content of share images. Therefore, the cheater who is a legitimate participant can create

(a) (b) (c) (d)

Fig. 9. Four different types of pictures.

TABLE I
NUMBER OF ADDED BLACK SUBPIXELS FOR THE PICTURES IN FIG. 9 WITH DIFFERENT SIZES AND CONTRASTS

	(a)				(b)				(c)				(d)			
	SZ1	SZ2	SZ3	SZ4	SZ1	SZ2	SZ3	SZ4	SZ1	SZ2	SZ3	SZ4	SZ1	SZ2	SZ3	SZ4
1/4	20	22	23	24	10	13	16	19	15	20	22	24	15	17	18	21
1/9	17	21	21	23	7	8	14	17	14	20	21	22	13	16	17	19
1/16	5	7	5	7	2	2	2	3	2	4	6	9	4	4	3	6
1/25	4	6	4	6	1	1	1	2	1	3	5	8	2	2	1	4

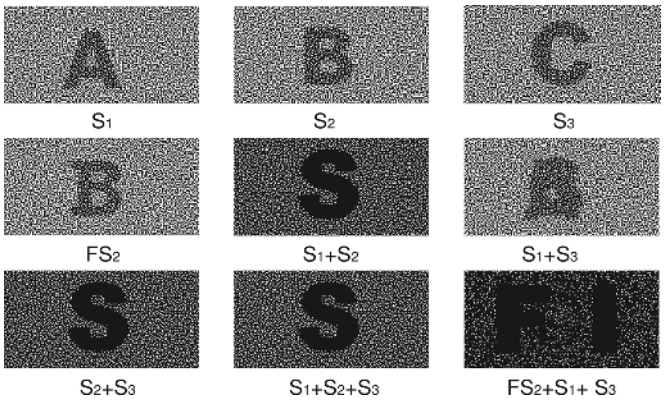

Fig. 10. Example of cheating a (Γ, m)-EVCS.

reasonable share images on fake shares according to his own share to cheat other participants. □

IV. ATTACKS AND IMPROVEMENT ON PREVIOUS CHEAT-PREVENTING METHODS

There are two types of cheat-preventing methods [26]. The first type is to have a trusted authority (TA) to verify the shares of participants. The second type is to have each participant to verify the shares of other participants. In this section, we present attacks and improvement on four existent cheat-preventing methods.

A. Attack on Yang and Laih's First Cheat-Preventing Method

The first cheat-preventing method of Yang and Laih [26] needs a TA to hold the special verification share for detecting fake shares. It generates $n+1$ shares $VS, S_1, S_2, \ldots, S_n$, where VS is the verification share. If $VS + S_i$ shows the verification image that is known to all participants, the share S_i is genuine. Let S^0 and S^1 be the basis matrices of a (Γ, m)-VCS. They assign pixels to shares by four sets $C_{0,0}$, $C_{0,1}$, $C_{1,0}$, and $C_{1,1}$ which are the sets of all $(n+1) \times (m+2)$-matrices obtained by permuting the columns of

$$S^{00} = \left[\begin{array}{c|ccc} 10 & 0 & \ldots & 0 \\ \hline 10 & & & \\ \vdots & & S^0 & \\ 10 & & & \end{array}\right]$$

$$S^{01} = \left[\begin{array}{c|ccc} 10 & 0 & \ldots & 0 \\ \hline 10 & & & \\ \vdots & & S^1 & \\ 10 & & & \end{array}\right]$$

$$S^{10} = \left[\begin{array}{c|ccc} 10 & 0 & \ldots & 0 \\ \hline 01 & & & \\ \vdots & & S^0 & \\ 01 & & & \end{array}\right]$$

$$S^{11} = \left[\begin{array}{c|ccc} 10 & 0 & \ldots & 0 \\ \hline 01 & & & \\ \vdots & & S^1 & \\ 01 & & & \end{array}\right]$$

respectively. Pixels are assigned to shares by a random matrix in C_{b_1,b_2}, where b_1 indicates the pixel in the verification image and b_2 indicates the pixel in the secret image. We see that the verification image shall appear on $VS + S_i$ if the share S_i is genuine since the first two subpixels reveals the verification image.

Our attack, depicted in Fig. 11, involves two malicious participants. Without loss of generality, we assume that they are P_1 and P_2. P_1 and P_2 together constructs a fake share FS such that $FS + VS$ reveals the verification image and FS cheats other participants.

We see how the attack works.

1) $FS+VS$ reveals the verification image. The reason is that the first two subpixels (before permutation) of FS and S_1 are the same. The first two subpixels of $FS + VS$ are the same as those of $S_1 + VS$. Thus, the verification image appears on $FS + VS$. The details are as follows.
   For the white pixel of the verification image, the first two pairs of subpixels in S_1 and S_2 are (1,1) and (0,0) by S^{00} and S^{01}, the corresponding subpixels in FS are the same as

Input: shares S_1 and S_2. (Wlog, we assume that P_1 and P_2 are cheaters.)
Fake share construction phase: P_1 and P_2 choose a fake image that has *no overlapping* with the verification image and then create the fake share FS as follows:

1) For a white pixel in the fake image, assign the corresponding pixel of S_1 to FS.
2) For a black pixel in the fake image, we assign its $m+2$ subpixels in FS as follows. Let (r, s) be the pair of the corresponding subpixels in S_1 and S_2, respectively. We consider two such pairs (r_1, s_1) and (r_2, s_2). If (r_1, s_1)=(1,0) and (r_2, s_2)=(0,0), we assign 0 and 1 to the corresponding subpixels in FS. The above step is repeated till no more assignments to FS are possible.
3) For the rest of unassigned subpixels in FS, copy those from S_1.

Share verification phase: P_1 and P_2 submit S_1 and FS to TA. TA checks the validity of S_1 and FS.
Image reconstruction phase (the fake image): For $Y = \{P_1, P_2, P_{i_1}, P_{i_2}, \ldots, P_{i_q}) \notin Q$, $S_1 + FS + S_{i_1} + S_{i_2} + \cdots + S_{i_q}$ reveals the fake image.

Fig. 11. Cheat against Yang and Laih's cheat-preventing Method.

those in S_1 by Step 2 in the fake share construction phase. Thus, the pixel of $FS+VS$ is white since S_1+VS shows whiteness in the pixel. For the black pixel of the verification image, the first two pairs of subpixels in S_1 and S_2 are (0,0) and (1,1) by S^{10} and S^{11}, the corresponding subpixels in FS are the same as those in S_1. Thus, the pixel of $FS+VS$ is black since $S_1 + VS$ shows blackness in the pixel.

2) For $Y = \{P_1, P_2, P_{i_1}, P_{i_2}, \ldots, P_{i_q}) \notin Q$, $S_1+FS+S_{i_1}+S_{i_2}+\cdots+S_{i_q}$ reveals the fake image. For the white pixel of the fake image, the pixel in FS is the same as that in S_1 by Step 1. Thus, the pixel in $S_1 + FS$ is white. For the black pixel of the fake image, the subpixels 1 and 0 of S_1 is changed to 0 and 1 in FS (see Step 2). Thus, the white pixel, containing subpixels

$$[\ldots 1 \ldots 0 \ldots] + [\ldots 0 \ldots 0 \ldots] = [\ldots 1 \ldots 0 \ldots]$$

of S_1+S_2 is changed to a black pixel, containing subpixels

$$[\ldots 1 \ldots 0 \ldots] + [\ldots 0 \ldots 1 \ldots] = [\ldots 1 \ldots 1 \ldots]$$

in $S_1 + FS$. Thus, the fake image appears on $S_1 + FS + S_{i_1} + \cdots + S_{i_q}$.

3) FS are indistinguishable by other participants. For each pixel, the numbers of black and white subpixels in the pixels of FS and S_1 are the same since the only change is to swap subpixels b and w in S_1 to w and b in FS. Thus, FS and S_1 look the same and other participants cannot distinguish them.

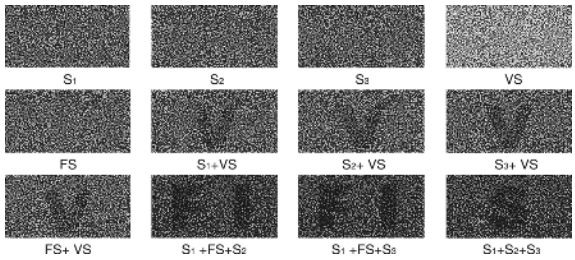

Fig. 12. Example of cheating the cheat-preventing (3,3)-VCS of Yang and Laih.

1) Example 4.1: Fig. 12 shows the results of cheating a (3,3)-VCS of Yang and Laih. We see that all shares including the fake share FS pass verification by revealing the correct verification image $\boldsymbol{V}$. Since $S_1 + FS + S_3$ reveals a fake image $\boldsymbol{FI}$, P_3 is cheated.

B. Attacks on Horng et al.'s Cheat-Preventing Methods

In the first cheat-preventing method of Horng *et al.* [14], each participant P_i has a verification share V_i. The share's S_is are generated as usual. Each V_i is divided into $n-1$ regions $R_{i,j}$, $1 \le j \le n$, $j \ne i$. Each region $R_{i,j}$ of V_i is designated for verifying share S_j. The region $R_{i,j}$ of $V_i + S_j$ shall reveal the verification image for P_i verifying the share S_j of P_j. The verification image in $R_{i,j}$ is constructed by a (2,2)-VCS. Although the method requires that the verification image be confidential, we show that it is still possible to cheat.

Assume that P_1 knows the regions of the verification share V_i. P_1 generates a fake share FS_1 to cheat P_i as follows. The pixels of FS_1 in the region $R_{i,1}$ are the same as those in S_1. The rest pixels of FS_1 (outside the region $R_{i,1}$) are constructed by **CA-1**. As a result, the correct verification image appears on the region $R_{i,1}$ of $FS_1 + V_i$ and P_i believes that FS_1 is a genuine share. By **CA-1**, the stacking of FS_1 and other genuine shares reveals a reasonable fake image. Moreover, even the cheater does not know the verification region assigned to a participant, the attack is still possible. Since the verification share is divided into $n-1$ regions, each verification region is small for a fairly large n. We choose a simple fake image. The probability that no overlapping between the fake image and the region $R_{i,1}$ occurs is high. By setting the background pixels in FS_1 from S_1, $FS_1 + V_i$ shows the verification image in the verification region $R_{i,1}$ of V_i.

By our proposed attacks, we conclude the following principle on using verification images.

1) Essential Principle: The verification images should be confidential and spread over the whole region of a share.

Horng *et al.*'s second cheat-preventing method uses the approach of redundancy [14]. It uses a $(2, n+l)$-VCS to implement a $(2, n)$-VCS cheat-preventing scheme. The scheme needs no on-line TA for verifying shares. The scheme generates $n+l$ shares by the $(2, n+l)$-VCS for some integer $l > 0$, but distributes only n shares to the participants. The rest of shares are destroyed. They reason that since the cheater does not know the exact basis matrices even with all shares, the cheater cannot succeed. However, our three cheating methods do not need to use

the basis matrices. Any of our cheating methods can cheat this cheat-preventing approach.

C. Improvement on Yang and Laih's Second Cheat-Preventing Method

The second cheat-preventing method of Yang and Laih [26] is a transformation of a (Γ, m)-VCS (but not a $(2, n)$-VCS) to another cheat-preventing $(\Gamma, m+n(n-1))$-VCS. The stacking of any two shares reveals the verification image. This is how share verification is done.

Let S^0 and S^1 be the basis matrices of a (Γ, m)-VCS. Their method constructs four sets $C_{0,0}$, $C_{0,1}$, $C_{1,0}$, $C_{1,1}$ of $n \times (m + n(n-1))$-matrices obtained by permuting the columns of the following four matrices, respectively:

$$S^{00} = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 0 & \dots & 1 & 1 & \\ 1 & 0 & 1 & 1 & \dots & 1 & 1 & \\ 1 & 1 & 1 & 0 & \dots & 1 & 1 & S^0 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & \\ \vdots & & \vdots & & \dots & 1 & 0 & \\ 1 & 1 & 1 & 1 & \dots & 1 & 0 & \end{array}\right]$$

$$S^{01} = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 0 & \dots & 1 & 1 & \\ 1 & 0 & 1 & 1 & \dots & 1 & 1 & \\ 1 & 1 & 1 & 0 & \dots & 1 & 1 & S^1 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & \\ \vdots & & \vdots & & \dots & 1 & 0 & \\ 1 & 1 & 1 & 1 & \dots & 1 & 0 & \end{array}\right]$$

$$S^{10} = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 0 & \dots & 1 & 1 & \\ 0 & 1 & 1 & 1 & \dots & 1 & 1 & \\ 1 & 1 & 0 & 1 & \dots & 1 & 1 & S^0 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & \\ \vdots & & \vdots & & \dots & 1 & 0 & \\ 1 & 1 & 1 & 1 & \dots & 0 & 1 & \end{array}\right]$$

$$S^{11} = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 0 & \dots & 1 & 1 & \\ 0 & 1 & 1 & 1 & \dots & 1 & 1 & \\ 1 & 1 & 0 & 1 & \dots & 1 & 1 & S^1 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & \\ \vdots & & \vdots & & \dots & 1 & 0 & \\ 1 & 1 & 1 & 1 & \dots & 0 & 1 & \end{array}\right].$$

The pixel expansion of this construction is $m' = m+n(n-1)$ and contrast is $\alpha(m') = (1 + (\alpha(m) \times m))/m'$, where $\alpha(m)$ is the contrast of the original VCS without cheating prevention.

By our observation [22], what the human eyes care about is contrast, no matter whether the image is darker or lighter than the background. Therefore, we gave a new definition VCS_2 for VC based on this observation and made improvements on some types of access structures [22]. Our improvements are applicable to Yang and Laih's cheat-preventing method. It reduces the pixel expansion to $m + n(n-1)/2$. Moreover, since the verification image can be made public to all participants, we can let the verification image appear on the shares. By this, we can further reduce the pixel expansion to $m + n(n-1)/4$.

Our improvement is based on the following three theorems, which are proven in [22].

Theorem 4.1: [22] *(Composition property)* Let $\Gamma_1 = (\mathcal{P}, Q_1, F_1)$ and $\Gamma_2 = (\mathcal{P}, Q_2, F_2)$ be two access structures. Assume that $Q_1 \cap Q_2 = \emptyset$. If there exist a (Γ_1, m_1)-VCS_2 and a (Γ_2, m_2)-VCS_2, there exist a $(\Gamma, m_1 + m_2)$-VCS_2, where $\Gamma = (\mathcal{P}, Q_1 \cup Q_2, F_1 \cap F_2)$. VCS_2 is a visual cryptography scheme based on the new definition proposed in [22].

Theorem 4.2: [22] *(Deletion property)* Let $\Gamma = (\mathcal{P}, Q, F)$ be an access structure. If S^0 and S^1 are basis matrices for a (Γ, m)-VCS_2, $S^{0'}$ and $S^{1'}$ are basis matrices for a $(\Gamma, m - k)$-VCS_2, where $S^{0'}$ and $S^{1'}$ are obtained from S^0 and S^1 by deleting the same k columns.

Theorem 4.3: [22] *(Inverse property)* Let $\Gamma = (\mathcal{P}, Q, F)$ be an access structure. If S^0 and S^1 are basis matrices for a (Γ, m)-VCS_2, $S^{0'}$ and $S^{1'}$ are basis matrices for a (Γ, m)-VCS_2, where $S^{0'} = S^1$ and $S^{1'} = S^0$.

We denote the left appended matrices in $S^{b_1 b_2}$ as $n(n-1)/2$ sub-matrices $S_k^{b_1 b_2}$, where $1 \le k \le n(n-1)/2, b_1, b_2 \in \{0, 1\}$. Each sub-matrix $S_k^{b_1 b_2}$ consists of two columns counting from left to right. Based on Theorems 4.1–4.3, we can exchange the roles of S_k^{00} and S_k^{10}, and also S_k^{01} and S_k^{11}, and delete $n(n-1)/2$ common columns. Furthermore, we delete all columns having one "0" only for the case that the verification image may not appear on the shares. By these steps, the pixel expansion of the appended matrices is reduced to $n(n-1)/4$.

Let $\mathcal{P} = \{P_1, P_2, P_3, P_4\}$. The basis matrices for a cheat-preventing (Γ, m)-VCS using Yang and Laih's cheat-preventing method are as follows:

$$S^{00} = \left[\begin{array}{cccccccccccc|c} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & S^0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & \end{array}\right]$$

$$S^{01} = \left[\begin{array}{cccccccccccc|c} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & S^1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & \end{array}\right]$$

$$S^{10} = \left[\begin{array}{cccccccccccc|c} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & S^0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & \end{array}\right]$$

$$S^{11} = \left[\begin{array}{cccccccccccc|c} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & S^1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & \end{array}\right].$$

We reduce the pixel expansion of the left appended matrices from 12 to 3, as follows:

$$S^{00} = \left[\begin{array}{ccc|c} 0 & 0 & 1 & \\ 0 & 1 & 0 & \\ 1 & 1 & 1 & S^0 \\ 1 & 0 & 0 & \end{array}\right], \quad S^{01} = \left[\begin{array}{ccc|c} 0 & 0 & 1 & \\ 0 & 1 & 0 & \\ 1 & 1 & 1 & S^1 \\ 1 & 0 & 0 & \end{array}\right]$$

$$S^{10} = \left[\begin{array}{ccc|c} 0 & 1 & 1 & \\ 1 & 0 & 1 & \\ 0 & 0 & 0 & S^0 \\ 1 & 1 & 0 & \end{array}\right], \quad S^{11} = \left[\begin{array}{ccc|c} 0 & 1 & 1 & \\ 1 & 0 & 1 & \\ 0 & 0 & 0 & S^1 \\ 1 & 1 & 0 & \end{array}\right].$$

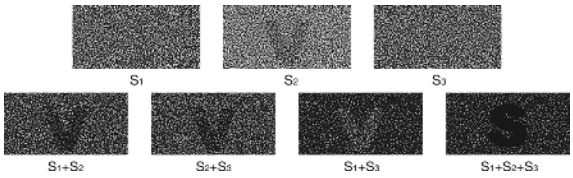

Fig. 13. Improved (3,3)-VCS$_2$ for Yang and Laih's cheat-preventing method.

1) Example 4.2: Fig. 13 shows the results of the improved cheat-preventing (3,3)-VCS$_2$. We see that the stacking of any two shares reveals the verification image $\mathbf{V}$. $S_1 + S_3$ reveals the reversed verification image and S_2 shows the verification image.

V. Generic Transformation for Cheating Prevention

By the attacks and improvement in previous sections, we propose that an efficient and robust cheat-preventing method should have the following properties.

1) It does not rely on the help of an on-line TA. Since VC emphasizes on easy decryption with human eyes only, we should not have a TA to verify validity of shares.
2) The increase to pixel expansion should be as small as possible.
3) Each participant verifies the shares of other participants. This is somewhat necessary because each participant is a potential cheater.
4) The verification image of each participant is different and confidential. It spreads over the whole region of the share. We have shown that this is necessary for avoiding the described attacks.
5) The contrast of the secret image in the stacking of shares is not reduced significantly in order to keep the quality of VC.
6) A cheat-preventing method should be applicable to any VCS.

We now present a generic transformation from a VCS to another cheat-preventing VCS. The resultant cheat-preventing VCS meets all the above requirements. The idea is similar to the first cheat-preventing method of Yang and Laih [26]. But, we let each participant hold a verification share. Our cheat-preventing scheme needs no help from an on-line TA. The verification image for each participant is different and known to the participant only.

Our transformation is quite efficient and almost optimal as it adds only two subpixels for each pixel of the original image. That is, if the pixel expansion of the VCS is m, the pixel expansion of the transformed VCS is $m + 2$. The contrast is slightly reduced from $\alpha(m)$ to $\alpha(m') = (\alpha(m) \times m + 1)/(m + 2)$. Our transformation is depicted in Fig. 14. It generates two shares for each participant. One is the secret share and the other is the verification share. Let S^0 and S^1 be the $n \times m$ basis matrices of a (Γ, m)-VCS. At first, we create two $n \times (m + 2)$-dimensional basis matrices T^0 and T^1. The transformed $(\Gamma, m + 2)$-VCS uses T^0 and T^1 as the basis matrices to generate shares for the participants as usual. Then, for each participant P_i, it generates a verification share V_i for a chosen verification image. For each white pixel in the verification image, it puts the pixel of $(m + 2)$-dimensional $[1 \quad 0 \quad 0 \quad \ldots \quad 0]$ to V_i (after corresponding permutation as for the share S_i). For each black pixel in the verification image, it puts the pixel of $(m + 2)$-dimensional $[0 \quad 1 \quad 0 \quad 0 \quad \ldots \quad 0]$ to V_i (after corresponding permutation as for the share S_i). We see that the verification image is encoded into the first two subpixels. If participant P_i wants to verify the share S_j of participant P_j, he checks whether $V_i + S_j$ shows his verification image.

Input: S^0 and S^1 of a (Γ, m)-VCS.
Shares construction phase:

1) Let

$$T^0 = \left[\begin{array}{c|c} 10 & \\ \vdots & S^0 \\ 10 & \end{array}\right] \quad and \quad T^1 = \left[\begin{array}{c|c} 10 & \\ \vdots & S^1 \\ 10 & \end{array}\right].$$

2) Use T^0 and T^1 as the basis matrices for generating shares S_i, $1 \le i \le n$, of $(\Gamma, m + 2)$-VCS.
3) For each participant P_i, $1 \le i \le n$, choose a verification image and generate a verification share V_i as follows:
   a) For each white pixel in the verification image, put the pixel of $(m + 2)$-dimensional $[1 \quad 0 \quad 0 \quad \cdots \quad 0]$ (subpixels) to V_i (after corresponding permutation as for the share S_i).
   b) For each black pixel in the verification image, put the pixel of $(m + 2)$-dimensional $[0 \quad 1 \quad 0 \quad 0 \quad \cdots \quad 0]$ (subpixels) to V_i (after corresponding permutation as for the share S_i).

Share verification phase:
Before stacking their shares, each participant P_i checks whether $V_i + S_j$ shows his verification image, where P_j is another participant.

Fig. 14. Our generic transformation for VCS with cheating prevention.

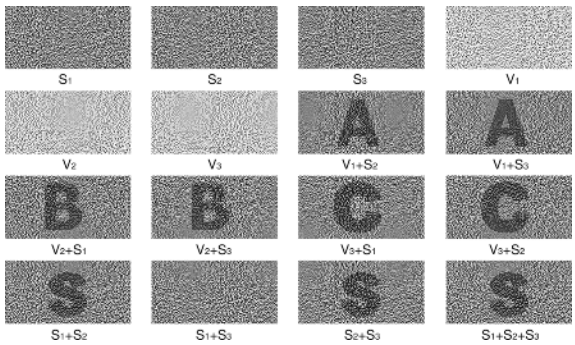

Fig. 15. Example of a transformed VCS with cheating prevention.

1) Example 5.1: Fig. 15 shows a transformed $(\Gamma, m + 2)$-VCS with cheating prevention, where $\mathcal{P} = \{P_1, P_2, P_3\}$ and $Q = \{(P_1, P_2), (P_2, P_3), (P_1, P_2, P_3)\}$. The verification images for participants P_1, P_2, and P_3 are $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively. Note that the simple verification images are for demonstration only. By our proposed principle in Section IV-B, we should use more complicated verification images.

Theorem 5.1: The algorithm in Fig. 14 transforms any (Γ, m)-VCS to another (Γ, m')-VCS with cheating prevention, where $m' = m + 2$ and $\alpha(m') = (\alpha(m) \times m + 1)/m'$.

Proof: Since the first two subpixels are all the same for all pixels in all shares of (Γ, m')-VCS, the secret image is not affected except that the contrast is slightly reduced to $\alpha(m') = (\alpha(m) \times m+1)/m'$. Thus, the transformation produces another $(\Gamma, m+2)$-VCS.

For P_i verifying the share S_j of P_j, we see how the verification image appears on $V_i + S_j$. For each black pixel of the verification image, the first two subpixels of $V_i + S_j$ is $[0 \quad 1] + [1 \quad 0] = [1 \quad 1]$. For each white pixel of the verification image, the first two subpixels of $V_i + S_j$ is $[1 \quad 0] + [1 \quad 0] = [1 \quad 0]$. Thus, the black and white pixels of the verification image have a positive contrast and P_i can see the verification image in $V_i + S_j$.

Each participant has his own private verification image, which is not known to other participants. Since the first two subpixels $[1 \quad 0]$ (before permutation) of all shares are the same, a participant P_i even with all shares cannot know the positions of black pixels of the verification image of participant P_j, $j \neq i$. Therefore, P_i cannot produce a fake share FS_i such that $FS_i + V_j$ shows the verification image of P_j. Participant P_i cannot cheat participant P_j for $i \neq j$. Furthermore, we see that collaboration of some participants cannot succeed to cheat, either. □

VI. Conclusion

We have proposed three cheating methods against VCS and EVCS. We examined previous cheat-preventing schemes and found that they are either not robust enough or still improvable. We presented an improvement on one of these cheat-preventing schemes. By our attacks, we pointed out an essential principle for a robust cheat-preventing VCS. We finally proposed an efficient transformation of VCS for cheating prevention. Our transformation incurs minimum overhead on contrast and pixel expansion. It only added two subpixels for each pixel in the image and the contrast is reduced only slightly.

References

[1] G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson, "Visual cryptography for general access structures," *Inf. Comput.*, vol. 129, no. 2, pp. 86–106, 1996.

[2] ——, "Extended capabilities for visual cryptography," *Theoret. Comput. Sci.*, vol. 250, no. 1–2, pp. 143–161, 2001.

[3] I. Biehl and S. Wetzel, "Traceable visual cryptography," in *Proc. 1st Int. Conf. Information Communication Security*, 1997, vol. 1334, LNCS, pp. 61–71.

[4] C. Blundo and A. De Santis, "Visual cryptography schemes with perfect reconstructions of black pixels," *Comput. Graph.*, vol. 22, no. 4, pp. 449–455, 1998.

[5] C. Blundo, A. De Santis, and D. R. Stinson, "On the contrast in visual cryptography schemes," *J. Cryptol.*, vol. 12, no. 4, pp. 261–289, 1999.

[6] C. Blundo, P. D'Arco, A. De Santis, and D. R. Stinson, "Contrast optimal threshold visual cryptography schemes," *SIAM J. Discrete Math.*, vol. 16, no. 2, pp. 224–261, 2003.

[7] E. F. Brickell and D. R. Stinson, "The detection of cheaters in threshold schemes," *SIAM J. Discrete Math.*, vol. 4, no. 4, pp. 502–510, 1991.

[8] S. Cimato, A. De Santis, A. L. Ferrara, and B. Masucci, "Ideal contrast visual cryptography schemes with reversing," *Inf. Process. Lett.*, vol. 93, no. 4, pp. 199–206, 2005.

[9] A. De Bonis and A. De Santis, "Randomness in secret sharing and visual cryptography schemes," *Theoret. Comput. Sci.*, vol. 314, no. 3, pp. 351–374, 2004.

[10] S. Droste, "New results on visual cryptography," in *Proc. Advances in Cryptology*, 1996, vol. 1109, LNCS, pp. 401–415.

[11] 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.

[12] H. Yan, Z. Gan, and K. Chen, "A cheater detectable visual cryptography scheme," (in Chinese) *J. Shanghai Jiaotong Univ.*, vol. 38, no. 1, 2004.

[13] 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.

[14] G.-B. Horng, T.-G. Chen, and D.-S. Tsai, "Cheating in visual cryptography," *Designs, Codes, Cryptog.*, vol. 38, no. 2, pp. 219–236, 2006.

[15] C.-M. Hu and W.-G. Tzeng, "Compatible ideal contrast visual cryptography with reversing," in *Proc. 8th Information Security Conf.*, 2005, vol. 3650, LNCS, pp. 300–313.

[16] M. Krause and H.-U. Simon, "Determining the optimal contrast for secret sharing schemes in visual cryptography," *Combin., Probab., Comput.*, vol. 12, no. 3, pp. 285–299, 2003.

[17] R. Ito, H. Kuwakado, and H. Tanaka, "Image size invariant visual cryptography," *IEICE Trans. Fundam. A*, vol. E 82, no. 10, pp. 2172–2176, 1999.

[18] M. Naor and B. Pinkas, "Visual authentication and identification," in *Proc. Advances in Cryptology*, 1997, vol. 1294, LNCS, pp. 322–336.

[19] M. Naor and A. Shamir, "Visual cryptography," in *Proc. Advances in Cryptology*, 1994, vol. 950, LNCS, pp. 1–12.

[20] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[21] M. Tompa and H. Woll, "How to share a secret with cheaters," *J. Cryptol.*, vol. 1, no. 2, pp. 133–138, 1988.

[22] W.-G. Tzeng and C.-M. Hu, "A new approach for visual cryptography," *Designs, Codes, Cryptog.*, vol. 27, no. 3, pp. 207–227, 2002.

[23] E. R. Verheul and H. C. A. Van Tilborg, "Constructions and properties of k out of n visual secret sharing schemes," *Designs, Codes, Cryptog,*, vol. 11, no. 2, pp. 179–196, 1997.

[24] D. Q. Viet and K. Kurosawa, "Almost ideal contrast visual cryptography with reversing," in *Proc, Topics in Cryptology*, 2004, vol. 2964, LNCS, pp. 353–365.

[25] C.-N. Yang, "New visual secret sharing schemes using probabilistic method," *Pattern Recognit. Lett.*, vol. 25, no. 4, pp. 481–494, 2004.

[26] C.-N. Yang and C.-S. Laih, "Some new types of visual secret sharing schemes," in *Proc. Nat. Computer Symp.*, 1999, vol. 3, pp. 260–268.

Chih-Ming Hu received the B.S. degree in computer and information science from the National Army Military Academy, Taiwan, R.O.C., in 1987, and the M.S. degree in computer and information science from the National Chiao-Tung University, Hsinchu, Taiwan, in 1998, where he is currently pursuing the Ph.D. degree in the Department of Computer Science.

His current research interest is in cryptology.

Wen-Guey Tzeng received the B.S. degree in computer science and information engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1985, and the M.S. and Ph.D. degrees in computer science from the State University of New York at Stony Brook in 1987 and 1991, respectively.

He joined the Department of Computer and Information Science (now the Department of Computer Science), National Chiao-Tung University, Hsinchu, Taiwan, in 1991. His current research interests include cryptology and network security.