An Improved Visual Cryptography with Cheating Prevention

Yu-Chi Chen[1], Kunhan Lu[2], Raylin Tso[2(✉)], and Mu-En Wu[3]

[1] Institute of Information Science, Academia Sinica,
Taipei, Taiwan
wycchen@ieee.org
[2] Department of Computer Science,
National Chengchi University, Taipei, Taiwan
101753018@nccu.edu.tw, raylin@cs.nccu.edu.tw
[3] Department of Mathematics, Soochow University,
Taipei, Taiwan
mn@scu.edu.tw

Abstract. Visual cryptography was firstly proposed by Naor and Shamir in 1994, which has been extended into many applications, including image encryption, information hiding, visual authentication, and visual identification. One important security issue in visual cryptography is the cheating prevention. However, in 2006, Horng et al. introduced the cheating problem in visual cryptography, where some dishonest participants, cheaters, can deceive other participants, victims, using forged transparencies. Since that, many cheating prevention works have been done in this area. In this paper, we introduce a new cheating prevention scheme which is secure and more efficient than previous schemes.

Keywords: Cheating prevention · Visual cryptography

1 Introduction

Visual cryptography is a field of cryptography proposed by Naor and Shamir in 1994 [15] to realize secret sharing without any computation, and therefore it is also called visual secret sharing (VSS). In a VSS scheme, participants only need to overlap image transparencies with each other to generate a reconstructed image that can be found by using the human vision's natural ability to perceive incomplete pictures and reveal a secret image. Compared to traditional secret sharing [1, 8], VSS does not require a computer to calculate any complex cryptographic operation. However, it only depends on stacking the transparencies with each other to decrypt the message. Based on this concept, many different research studies have been introduced, such as image encryption [6, 12], visual authentication and identification [14], steganography [4, 21], or some non-binary secret images, i.e. gray-scale images [2, 13] and color images [8, 17]. On the other hand, some studies focus on enhancing the contrast of the reorganization image and improving the pixel expansion [3, 19].

In addition to the above relative studies, in 2006, Horng et al. first showed how visual secret images can be forged [9]. The scenario is like that some dishonest

Y.-Q. Shi et al. (Eds.): IWDW 2014, LNCS 9023, pp. 444–454, 2015.
DOI: 10.1007/978-3-319-19321-2_34

participants collude together, and then they can calculate the shared images of other honest participants. Finally, they are able to generate forged shared images to deceive the others. Since then, how to prevent the cheating problem in visual secret sharing has attracted lots amount of attention. Therefore, cheating prevention visual secret-sharing (CPVSS) schemes have come into limelight [5, 7, 10, 11, 16, 20, 22]. In this paper, we focus on cheating prevention in VSS. Recently, [5] pointed out that the method in [10] was insecure, and put forward a proposal to improve this fault. However, this proposal requires a significant amount of pixel expansion which significantly reduces the clarity of the secret image. Therefore, [11] proposed a new improvement method to minimize pixel expansion. However, it still has some problems so we introduce a new scheme to remedy.

2 Visual Cryptography

2.1 Model

In 1994, visual cryptography techniques were proposed by Naor and Shamir [15]. This technique used a VSS mechanism to encrypt a secret image into n shared images. If the shared images are superimposed over at least k pieces, it is possible to decrypt the original secret information. This is the so called k out of n scheme. For example, a two out of two mechanism encrypts a secret image into two shared images, and by superimposing the two shared images, secret information can be obtained (Fig. 1).

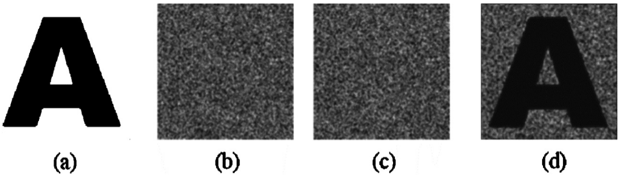

Fig. 1. Two out of two scheme: (a) secret image, (b) shared image 1, (c) shared image 2, and (d) stacked result

In a VSS scheme, first the input secret image is encrypted. The conventional process of encryption uses pixel expansion. Assuming a pixel of the secret image is white, then one row from the white section of Fig. 2 is randomly selected, and the 2 × 2 blocks of pixels are written to shared images 1 and 2, respectively, such that an image of two black and two white pixels results after superimposition. Conversely, for a black pixel of the secret image, one row from the black section of Fig. 2 is randomly selected, and the 2 × 2 pixel blocks are written to shared images 1 and 2, respectively, such that an all-black image results after superimposition. Based on human visual characteristics, the block of two black and two white pixels will appear gray, and have 50 % chromatic aberration with respect to the all-black blocks. Hence, the original secret information can be obtained after the images being superimposed.

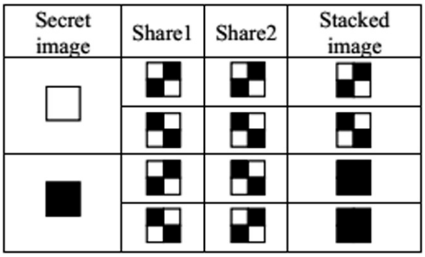

Fig. 2. Sharing and stacking scheme of black and white pixels.

Given a secret image, pixel expansion can be used to generate n shared images that are given to n secret participants, and as long as there are k or more participants to superimpose the shared images, hidden secrets will be recovered. The above mechanism is called a (k, n)-threshold VSS mechanism (or scheme) [15].

A VSS scheme is a special variant of a k-out-of-n secret-sharing scheme, where the shares given to participants are copied onto transparencies. Therefore, a share is also called a transparency. If X is a qualified subset of participants, then the participants in X can visually recover the secret image by stacking their transparencies without performing any cryptographic computation. Usually, the secret is an image. To create the transparencies, each pixel, either black or white, of the secret image is separately handled. It appears as a collection of m black and white subpixels in each of the n transparencies. We say that these m subpixels together form a block. This block is referred to as a black (or white) block if the pixel to be shared is black (or white). Therefore, a pixel of the secret image corresponds to $n \times m$ subpixels. We can describe the $n \times m$ subpixels by an $n \times m$ Boolean matrix, called a *base matrix*, $S = \left[S_{ij}\right]$ such that $S_{ij} = 1$ if and only if the j-th subpixel of the i-th share is black and $S_{ij} = 0$ if and only if the j-th subpixel of the i-th share is white. The gray level of the stack of k-shared blocks is determined by the Hamming weight H(V) of the ORed m-vector V of the corresponding k rows in S. This gray level is interpreted by the visual system of the participants as black if $H(V) \geq d$ and as white if $H(V) \leq d - \alpha \times m$ for some fixed threshold d and relative difference α. Usually, m and α are referred to as the pixel expansion factor and the scheme contrast, respectively. We would like m to be as small as possible and α to be as large as possible.

More formally, a solution to a k-out-of-n VSS scheme consists of two collections C^0 and C^1 of $n \times m$ base matrices. To share a white pixel, the dealer randomly chooses one of the matrices from C^0, and to share a black pixel, the dealer randomly chooses one of the matrices from C^1. The chosen matrix determines the m subpixels in each one of the n transparencies. The solution is considered valid if the two conditions are met.

Contrast conditions:

1. For any matrix S^0 in C^0, V of any k of the n rows satisfies $H(V) \leq d - \alpha \times m$.
2. For any matrix S^1 in C^1, V of any k of the n rows satisfies $H(V) \geq d$.

Security condition:

3. For any subset $\{i_1, i_2, \ldots, i_q\}$ of $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{n}\}$ with q < k, the two collections D^0 and D^1 of q × m matrices obtained by restricting each n × m matrix in C^0 and C^1 to rows $i_1, i_2, \ldots, i_q$ are indistinguishable, in the sense that they contain the same matrices with the same frequencies.

In the black-and-white VSS mechanism, first we assume that S^0 and S^1 are the two fundamental matrices of size n × m used to generate the shared image, where S^0 represents a white point and S^1 represents a black point. For example, in a (k, n)-threshold VSS mechanism, dealer assume that the secret image at each pixel in an image share S_i (where i = 1, 2, 3, …, n) is a pixel expansion of m points, where S^0 and S^1 are defined as follows.

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

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

In this case, n = m = 3, k = 2, and S_i is generated as follows:

Step 1: If the pixel of secret image is white, three bits of S^0 are put into the i-th row into S_i.

Step 2: If the pixel of secret image is black, three bits of S^1 are put into i-th row into S_i.

2.2 Cheating

The issue of cheating is well studied and understood in secret-sharing schemes [18]. Since Visual Cryptography (VC) is a variant of secret sharing, it is natural to also consider this issue. Most cheating attacks in VC are known plaintext attacks where the cheaters know the secret image and are able to infer the blocks of the victim's transparency based on the base matrices. Let us consider a 2-out-of-3 VSS scheme as an example. Assume Alice, Bob, and Carol are three participants in a 2-out-of-3 VSS scheme. In the following, we refer to an image as a message since each image represents a password. A secret message is transformed into three distinct shared images, denoted by S_A, S_B, and S_C. They are then delivered to Alice, Bob, and Carol, respectively. Stacking two of the three shares will reveal the secret message. Figure 3 shows the overall cheating process.

Alice and Bob are assumed to be the collusive cheaters who intend to deceive the victim Carol. The related parameters used are $B_v = 2$, $W_v = 1$, $H(S^0) = 1$, $H(S^1) = 1$, and m = 3, where:

m: the number of subpixels in a block.

B_V: the number of black subpixels in a block that represents a single black pixel of the reconstructed secret image.

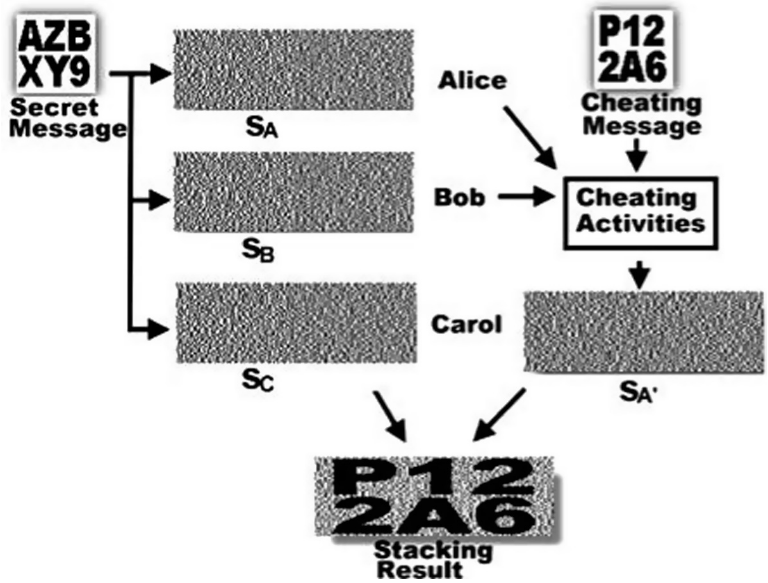

Fig. 3. Horng et al. in 2006 [9]: cheating in a visual cryptographic scheme.

W_V: the number of black subpixels in a block that represents a single white pixel of the reconstructed secret image.
$H(S^0)$: the number of black subpixels of any block in C^0.
$H(S^1)$: the number of black subpixels of any block in C^1.

Let

$$C^o = \begin{bmatrix} C_1^0 \\ C_2^0 \\ C_3^0 \end{bmatrix} = \left\{ \text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \right\}$$

$$C^1 = \begin{bmatrix} C_1^1 \\ C_2^1 \\ C_3^1 \end{bmatrix} = \left\{ \text{all the matrices obtained by permuting the columns of } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\}$$

Based on C^o and C^1, it produces three shares S_A, S_B, and S_C. If the i-th pixel in the secret message is white, a matrix M^0 is chosen randomly from C^0 and M_1^0, M_2^0, and M_3^0 are assigned to S_{Ai}, S_{Bi}, and S_{Ci}, respectively. Conversely, if the i-th pixel is black, a matrix M^1 is chosen randomly from C^1 and M_1^1, M_2^1, and M_3^1 are assigned to S_{Ai}, S_{Bi}, and S_{Ci}, respectively. This operation will repeat until every pixel of the secret message is encoded. Intuitively, collusive cheaters can derive the exact values from their shares. The secret message is composed of many white or black blocks. If the cheaters intend to cheat someone, it is necessary for them to change the construction of their shares. First, they predict the positions of black and white subpixels in the victim's share. Then, based on this prediction, they change the positions of the black and white subpixels in the forged shares. Finally, after stacking the forged shares with the victim's shares, the forged message will be revealed instead of the real secret message. The main problems for cheaters are how to predict the positions of black and white subpixels in the victim's share and rearrange the new positions of black and white subpixels in the cheaters' shares. There are four possible cases, as listed in Table 1.

Table 1. Horng et al. [9]: the basic concept of cheating in 2-out-of-3 VSS.

	Pixel in Secret message	Block in Share S_A	Block in Share S_B	Block in Share S_C	Pixel in Cheating message	Block in Share S'_A	Block in Share S'_B
Case 1	white	[1 0 0]	[1 0 0]	[1 0 0]	white	[1 0 0]	[1 0 0]
Case 2	white	[1 0 0]	[1 0 0]	[1 0 0]	black	[0 1 0]	[0 0 1]
Case 3	black	[1 0 0]	[0 1 0]	[0 0 1]	white	[0 0 1]	[0 0 1]
Case 4	black	[1 0 0]	[0 1 0]	[0 0 1]	black	[1 0 0]	[0 1 0]

3 The Proposed Cheating Prevention Scheme

The cheating prevention scheme proposed by Hu et al. [10] seems to be secure until 2012. In the year Chen et al. found a new attack technique [5]. Chen et al. also introduced a new scheme as a remedy. However, the pixel has been expanded to twice its original width, hence in order to reduce the required space, Liu et al. proposed an improved scheme [11] that requires minimal pixel expansion to prevent the attack. However, we found that in Liu et al.'s scheme, malicious participants can generate a forged shared image to cheat on honest participants using the black regions of the secret image. In order to solve this problem such that malicious participants cannot generate a forged shared image, we propose a new scheme.

Let S^0 and S^1 be the n × m-sized basic matrices for shared image generation in a VSS method, where S^0 and S^1 are for white and black pixels, respectively. Furthermore, each participant P_i holds shared image S_i ($i = 1, 2, \ldots, n$) and a pixel in a secret image is expanded to m subpixels in a shared image.

First, dealer create five n × (m + 3)-sized basic matrices T^0, T^1, R^0, R^1, and R^2 as follows:

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

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

$$R^1 = \left[\begin{array}{ccc|c} 0 & 1 & 0 & \\ & \vdots & & 0 \\ 0 & 1 & 0 & \end{array}\right], \quad R^2 = \left[\begin{array}{ccc|c} 0 & 0 & 1 & \\ & \vdots & & 0 \\ 0 & 0 & 1 & \end{array}\right],$$

where, T^0 and T^1 are used to generate shared image S_i, as in [10]. In our scheme, participants can choose their desired verification image. The generation of verification shared-image is divided into four cases:

Case 1: The focal pixels in the secret and verification images are white.

Case 2: The focal pixels in the secret and verification images are black and white, respectively.

Case 3: The focal pixels in the secret and verification images are white and black, respectively.

Case 4: The focal pixels in the secret and verification images are black.

Furthermore, each (m + 3)-length subpixel in the verification shared image V_i is generated as follows:

Case 1: As in [11], r_i^0 is put into V_i. In addition, where party-dependent (m + 3)-length row vector r_i^0 is obtained from t_i^0, the i-th row of T^0 (where i = 1, 2, …, n), and t_i^0 is defined by the following formula

$$t_i^0 = \left[100|s_i^0\right],$$

where s_i^0 is the i-th row of S^0, the number of ones in s_i^0 is x (where $0 < x < m$), and the number of ones in t_i^0 is (x + 1). The position of a one is randomly chosen from the (x + 1) existing ones, and other ones are set to zero to obtain a new (m + 3)-length row vector r_i^0. For example, when $t_i^0 = [1\,0\,0\,1\,0\,0]$, $r_i^0 = [1\,0\,0\,0\,0\,0]$ or $r_i^0 = [0\,0\,0\,1\,0\,0]$.

Case 2: As in [16], the i-th row of R^0 is put into V_i as (m + 3)-length subpixels.

Case 3: First, dealer randomly select a V_i from V_1 to V_n. If the point happens to be in case 3, then dealer put the i-th row of R^1 into V_i as in [10]. For other participants $P_j (j \neq i)$ and V_j is in case 3, then dealer put the j-th row of R^2 into V_j. In other words, only one participant's V is generated by R^1, all the other participants' Vs are generated by R^2. For example, if there are five participants with verification images V_1 to V_5, respectively. First, assume dealer randomly selected V_2 from V_1 to V_5. In addition, assume V_1, V_2, and V_4 happened to be in case 3. As a result, dealer would put the 2-nd row of R^1 into V_2, and put the 1-st row of R^2 into V_1 and the put 4-th row of R^2 into V_4.

Case 4: The procedure for case 4 is the same as for case 3. First, dealer randomly select a V_i from V_1 to V_n. If that point happens to be in case 4, then dealer put the i-th row of R^1 into V_i as in [10]. For other participants $P_j(j \neq i)$ and V_j is in case 4, then dealer put the j-th row of R^2 into V_j. In other words, only one participant's V is generated by R^1, all the other participants' Vs are generated by R^2. For example, if there are five participants with verification images V_1 to V_5, respectively. First, assume dealer randomly selected V_2 from V_1 to V_5. In addition, assume V_1, V_2, and V_4 happened to be in case 4. As a result, dealer would put the 2-nd row of R^1 into V_2, and put the 1-st row of R^2 into V_1 and the put 4-th row of R^2 into V_4.

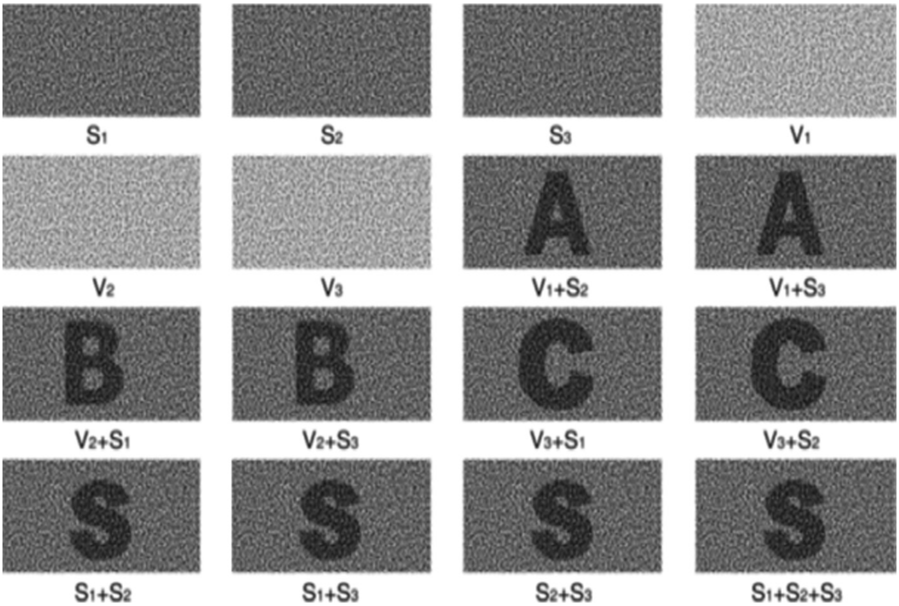

Fig. 4. Example of our proposed scheme 2 on a (2, 3)-threshold VSS method.

Figure 4 is an example of our proposed scheme 2 on a (2, 3)-threshold VSS method. Three participants P_1, P_2, and P_3 have their own verification images A, B, and C, respectively. For P_1, if S_2 and S_3 are stacked with V_1, respectively, and verification image A appears, then it can be guaranteed that S_2 and S_3 are the correct shared images. Similarly, P_2 and P_3 can also use the same method to confirm whether they have the correct shared image. All the verification pixels cannot be accurately estimated, so it is impossible to generate a forged shared image.

4 Security

Here, we analyze the security of our scheme in four cases.

Case 1: In this case, there is no difference between our scheme and the scheme [11]. If the malicious participants wish to cheat together to lead a honest participant into believing that the secret image is black, they will have $1/2$ opportunity of wrongly guessing the position having value 1 in the verification image of the honest participant. Assuming an image size is X × Y, and each pixel has $1/4$ probability to be in case 1, so the probability of successfully generating a forged shared image is $\left(\frac{1}{2}\right)^{\frac{XY}{4}}$.

Case 2: As scheme 2 slightly expands the verification bit and allows only one participant's V to be generated by R^1, where all the other participants' Vs are generated by R^2 when the verification image is black. If the malicious participants choose inverted verification images as in [5] to attack the honest participants, they will have $1/n$ opportunity of wrongly guessing all the positions of the verification bits (where n is the number of participants). For example, in a (2, 3)-threshold VSS, if the malicious participants wish to cheat together and lead the honest participants to believe that the secret image is white, they will have $1/3$ opportunity of wrongly guessing all the positions of the verification bits. In this situation, they will have $1/2$ opportunity of wrongly guessing the position having value 1 in the share image of the honest participant. Hence, the attack will fail with a probability of $\frac{1}{3} \times \frac{1}{2}$. Assuming an image size is X × Y, and each pixel has $1/4$ probability to be in case 2, so the probability of successfully generating a forged shared image is $\left(\frac{5}{6}\right)^{\frac{XY}{4}}$.

Case 3: As in case 2, only one participant's V is generated by R^1, and all other participants' Vs are generated by R^2 when the verification image is black. If the malicious participants choose inverted verification images as in [5] to attack the honest participants, they will have $1/n$ opportunity of wrongly guessing all the positions of the verification bits (where n is the number of participants). For example, in (2, 3)-threshold VSS, if the malicious participants wish to cheat together to lead the honest participants to believe that the secret image is black, they will have $1/3$ opportunity of wrongly guessing all the positions of the verification bits. In this situation, they will have $2/3$ opportunity of wrongly guessing the position having value 1 in the verification image of the honest participant. Hence, the attack will fail with probability $\frac{1}{3} \times \frac{2}{3}$. Assuming an image size is X × Y, and each pixel has $1/4$ probability to be in case 3, so the probability of successfully generating a forged shared image is $\left(\frac{7}{9}\right)^{\frac{XY}{4}}$.

Case 4: As in case 2, scheme 2 slightly expands the verification bits, and only one participant's V is generated by R^1, where and all the other participants' Vs are generated by R^2 when the verification image is black. If the malicious participants choose inverted verification images as in [5] to attack the honest participants, they will have $1/n$ opportunity of wrongly guessing all the positions of the verification bits (where n is the number of participants). For example, in (2, 3)-threshold VSS, if the malicious participants wish to cheat together to lead the honest participants to believe that the secret image is white, they will have $1/3$ opportunity of wrongly guessing all the positions of the verification bits. In this situation, they will have $1/2$ opportunity of wrongly guessing the position having value 1 in the share image of the honest participant. Hence, the attack will fail with a probability of $\frac{1}{3} \times \frac{1}{2}$. Assuming an image size is X × Y, and each pixel has $1/4$ probability to be in case 4, so the probability of successfully generating a forged shared image is $\left(\frac{5}{6}\right)^{\frac{XY}{4}}$.

The report in [7] mentions two kinds of cheating, meaningful cheating and meaningful deterministic cheating. We now discuss these types of cheating with respect to schemes 1 and 2. In scheme 1, for any single point, malicious participants cannot completely construct a forged share point, so the scheme can resist meaningful deterministic cheating. In scheme 2, for any single point, malicious participants in some situations can completely generate a forged share point, so the scheme cannot resist meaningful deterministic cheating. However, for the whole image, malicious participants cannot generate a complete forged shared image, so the scheme can resist meaningful cheating (Table 2).

Table 2. Performance comparison

Scheme	Pixel expansion	Security
De Prisco and De Santis [16]	m + 2	Insecure
Wang et al. [21]	m + (n + 1)	Secure
Liu et al. [11]	m + 2	Insecure
Our scheme	m + 3	Secure

ps: m is the number of bits required for presenting a pixel in any VSS scheme without cheating prevention

5 Conclusions

Visual cryptography was proposed by Naor and Shamir in 1994. Cheating is a well-known security issue. In this paper, we have introduced a new CPVSS scheme. As a result, our scheme is secure and more efficient than the existing schemes.

Acknowledgment. This research is supported by the National Science Council of the Republic of China under Contract no. MOST 103-2221-E-004-009 and MOST 103-2218-E-031-001.

References

1. Blakley, G.: Safeguarding cryptographic keys. In: Proceedings. AFIPS National Conference, p. 313 (1979)
2. Blundo, C., De Santis, A., Naor, M.: Visual cryptography for grey level images. Inf. Process. Lett. **75**(6), 255–259 (2000)
3. Blundo, C., D'Arco, P., De Santis, A., Stinson, D.R.: Contrast optimal threshold visual cryptography schemes. SIAM J. Discrete Math. **16**(2), 224–261 (2003)

4. Chang, C.C., Chuang, J.C.: An image intellectual property protection scheme for gray-level image using visual secret sharing strategy. Pattern Recog. Lett. **23**(8), 931–941 (2002)
5. Chen, Y.C., Horng, G., Tsai, D.S.: Comment on 'cheating prevention in visual cryptography'. IEEE Trans. Image Process. **21**(7), 3319–3323 (2012)
6. Chen, T.H., Tsai, D.S.: Owner-customer right protection mechanism using a watermarking scheme and a watermarking protocol. Pattern Recog. **39**(8), 1530–1541 (2006)
7. Chen, Y.C., Tsai, D.S., Horng, G.: Visual secret sharing with cheating prevention revisited. Digit. Signal Proc. **23**(5), 1496–1504 (2013)
8. Hou, Y.C.: Visual cryptography for color images. Pattern Recog. **36**(7), 1619–1629 (2003)
9. Horng, G., Chen, T.H., Tsai, D.S.: Cheating in visual cryptography. Des. Codes Crypt. **38** (2), 219–236 (2006)
10. Hu, C.M., Tzeng, W.G.: Cheating prevention in visual cryptography. IEEE Trans. Image Process. **16**(1), 36–45 (2007)
11. Liu, S.C., Fujiyoshi, M., Kiya, H.: A cheat-prevention visual secret sharing scheme with efficient pixel expansion. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E96-A** (11), 2134–2141 (2013)
12. Lukac, R., Plataniotis, K.N.: Bit-level based secret sharing for image encryption. Pattern Recog. **38**(5), 767–772 (2005)
13. Lin, C.C., Tsai, W.H.: Visual cryptography for gray-level images by dithering techniques. Pattern Recog. Lett. **24**(1–3), 349–358 (2003)
14. Naor, M., Pinkas, B.: Visual authentication and identification. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 322–336. Springer, Heidelberg (1997)
15. Naor, M., Shamir, A.: Visual cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
16. De Prisco, R., De Santis, A.: Cheating immune threshold visual secret sharing. Comput. J. **53**(9), 1485–1496 (2009)
17. Rijmen, V., Preneel, B.: Efficient colour visual encryption for shared colors of Benetton. In: Proceedings of the Rump Session of EUROCRYPTO, Berlin, Germany (1996)
18. Shamir, A.: How to share a secret. Comm. ACM **22**(11), 612–613 (1979)
19. Shamir, A., Naor, M.: Visual cryptography II. In: Lomas, M. (ed.) Security Protocols. LNCS, vol. 1189, pp. 1–12. Springer, Heidelberg (1996)
20. Tsai, D.S., Chen, T.H., Horng, G.: A cheating prevention scheme for binary visual cryptography with homogeneous secret images. Pattern Recog. **40**(8), 2356–2366 (2007)
21. Wang, C.C., Tai, S.C., Yu, C.S.: Repeating image watermarking technique by the visual cryptography. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E83-A**(8), 1589–1598 (2000)
22. Yan, H., Gan, Z., Chen, Z.: Acheater detectable visual cryptography scheme. J Shanghai Jiaotong Univ. **38**(1), 179–196 (2004)