HOW TO CONSTRUCT PROGRESSIVE VISUAL CRYPTOGRAPHY SCHEMES

Wenjuan Wang and Hachiro Fujita

Division of Information and Communications Systems
Tokyo Metropolitan University

ABSTRACT

A visual cryptography scheme (VCS) is an encryption method for images that does not need a computer to decode a secret image. In this paper we propose a simple, progressive VCS which is constructed from a traditional threshold VCS. Our progressive VCS has better decoded image quality and flexibility than the Fang–Lin scheme, a previous progressive VCS. We also present a block-wise progressive VCS which has jigsaw puzzle like decryption, and an XOR-based progressive VCS without pixel expansion. We give some experimental results which show that our scheme is superior to the Fang–Lin scheme with respect to the decoded image quality and the decoding speed controllability.

***Index Terms*—** Contrast, pixel expansion, progressive scheme, threshold scheme, visual cryptography

1. INTRODUCTION

In this paper[1] we propose a simple, progressive VCS that is constructed from a traditional threshold VCS [2]. Our scheme is secure and guarantees the decoded image quality. Further, our decryption has not only a threshold structure but also a progressive refinement: the number of shares pooled is smaller than the threshold, then stacking the shares reveals nothing about the secret image, but if the number of shares pooled exceeds the threshold and becomes larger and larger, then the secret image reveals progressively.

The paper is organized as follows: In Section 2 we give the details of our progressive VCS and analyze our scheme in terms of *average contrast*, a generalization of the conventional contrast. In Section 3 we show an XOR-based progressive VCS without pixel expansion. In Section 4 we show some experimental results and compare our scheme with the Fang–Lin scheme [3], one of previous progressive VCSs (see also [4, 5, 6]). The results show that our scheme is superior to the Fang–Lin scheme with respect to the decoded image quality and the decoding speed controllability.

We generally follow the notation and terminology used in the literature of VC (see, e.g., [7]). The proofs of the lemmas in the text are omitted due to lack of space.

[1]This paper is an extended and improved version of [1].

2. PROPOSED PROGRESSIVE VCS

2.1. Review of Threshold VC

Our scheme is constructed from a (t, n)-threshold VCS. Before giving the details of our scheme we briefly review the threshold VCS of Naor and Shamir [2]. To illustrate the concept of threshold VC we restrict ourselves to the $(2, 2)$-threshold VCS. In this scheme two participants and a dealer (a trusted third party) are involved. The dealer encodes a secret image to two shares and distributes each of them to each one of the participants. Let

$$A_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

We call the above two matrices, A_0 and A_1, the *basis matrices* of the $(2, 2)$-threshold VCS. Using the basis matrices A_0 and A_1, we define C_0 (resp. C_1) as the set of all the matrices obtained by permuting the columns of A_0 (resp. A_1). The encryption algorithm of the $(2, 2)$-threshold scheme is as follows. If a pixel in the secret image is white, then the dealer randomly chooses one of matrices in C_0 and distributes the first row of the chosen matrix to one participant and the second row to the other. A black pixel is encrypted in the same way using C_1 instead of using C_0. Now each pixel in the secret image is expanded into a 2×2 block of subpixels and the images generated by the encryption algorithm are called *shares*, which are four times larger than the original image (the $(2, 2)$-threshold scheme has *pixel expansion* four). The secret image can be recovered by simply stacking two shares, which corresponds to ORing the two rows of A_0 (resp. A_1). The relative difference between blackness of decoded black and white pixels, $\alpha = (4 - 2)/4 = 1/2$, is called *contrast*. For more details on the formal definition and useful properties of threshold schemes see [2]. For later use we give some examples of the parameters of a (t, n)-threshold VCS with pixel expansion four. (i) $(t, n) = (2, 4)$: A_1 is the 4×4 identity matrix and A_0 is the 4×4 matrix whose rows are copies of $(1, 0, 0, 0)$. (ii) $(t, n) = (3, 3)$:

$$A_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

2.2. (t, n)-Threshold Progressive VCS

In this section we give the details of our progressive VCS. Let $2 \le t \le n_0 \le n$ be positive integers and assume that there exists a (t, n_0)-threshold VCS. Below we will construct a (t, n)-threshold progressive VCS that encrypts a secret image to n shares and has the decryption property that if the number of shares pooled is less than threshold t then the stacking of the shares pooled reveals nothing about the secret image, but if the number of shares pooled exceeds threshold t and becomes larger and larger then the stacking reveals the secret image progressively. Here we consider the case $t = 2, 3$ only.

Our construction is very simple: we extend the basis matrices of a (t, n_0)-threshold VCS to larger matrices in a probabilistic way. Let A_0 and A_1 be the $n_0 \times m$ basis matrices of a (t, n_0)-threshold VCS. For practical applications and for fair comparison with the Fang–Lin scheme we restrict ourselves to $m = 4$, i.e., a scheme with pixel expansion four. We extend $n_0 \times m$ matrices A_0 and A_1 to $n \times m$ matrices $\hat{A}_0$ and $\hat{A}_1$ by attaching new $n - n_0$ rows to A_0 and A_1, respectively. To save space we consider the transposes A_0^T and A_1^T and attach new columns to A_0^T and A_1^T. Let $A_1^T = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{n_0}]$, where $\boldsymbol{a}_i$, $1 \le i \le n_0$, is the ith column of the A_1^T. We generate $n - n_0$ columns, $\boldsymbol{a}_j$, $n_0 + 1 \le j \le n$, to be attached to A_1^T in the following way:

$$\boldsymbol{a}_j = \begin{cases} \boldsymbol{a}_1 & \text{with probability } p, \\ \cdots & \cdots \\ \boldsymbol{a}_{n_0} & \text{with probability } p, \\ \mathbf{0} & \text{with probability } 1 - n_0 p, \end{cases}$$

where $0 \le p \le 1/n_0$ and $\mathbf{0}$ denotes the all-zero vector of length m. This results in $\hat{A}_1^T = [A_1^T, \boldsymbol{a}_{n_0+1}, \ldots, \boldsymbol{a}_n]$. In the same way we construct $\hat{A}_0^T$. Note that in the threshold VCS with $t = 2$ the extension of A_0 is just to attach some copies of the row of fixed pattern or the all-zero vector. The parameter p introduced above makes the decoding speed variable: if p becomes large then the decoding accelerates.

Our encryption algorithm is almost the same as that of the traditional threshold VCS. In our scheme, however, we permute not only the columns but also the rows of the extended basis matrices $\hat{A}_0$ and $\hat{A}_1$. The row permutation is crucial since, if the order of the rows is fixed, then the statistics of the shares corresponding to the rows attached to the underlying basis matrix differs from that of the shares corresponding to the original rows of the basis matrix, which distinguishes important shares from less important ones. Consider, for example, the extremal case $p = 0$. In this case the attached rows are the all-zero vectors and the corresponding shares have nothing to do with the secret image.

We remark that the extended basis matrices must be constructed for each pixel encryption: if we use the same extended basis matrices for all black/white pixels then progressive decoding does not work. To make the scheme probabilistic we need a large number of samples that obey a given distribution. This is similar to the law of large numbers: the empirical distribution of a large number of samples approximates the true probability distribution.

In the next section we will show that the extended basis matrices constructed above not only generate a secure share but also guarantee the decoded image quality which is the same as that of the underlying (t, n_0)-threshold VCS (e.g., if we use the $(2, 4)$-threshold VCS as an underlying scheme then the final conventional contrast is $3/4$).

2.3. On the Contrast in Our Scheme

In our (t, n)-threshold progressive VCS, if the number, k, of shares pooled exceeds the threshold t, then the stacking result reveals the secret image in a probabilistic manner. In this section we investigate the visual quality of a decoded image as a function of k.

We first consider the $(2, n)$-threshold progressive VCS constructed from the $(2, 2)$-threshold VCS, where $n \ge 2$. Let $p_i(k)$ (resp. $q_i(k)$) denote the probability that a decoded block corresponding to a black (resp. white) pixel in the secret image has i black subpixels (hence $4 - i$ white subpixels) after stacking k shares. In fact, in the $(2, n)$-threshold progressive VCS, $p_i(k)$ is defined for $i = 0, 2, 4$. Recall the basis matrix A_1 of the $(2, 2)$-threshold VCS. Each row of A_1 has (Hamming) weight 2 and the ORing of the two rows has weight 4, while an attached all-zero vector has weight 0. Similarly, $q_i(k)$ is defined for $i = 0, 2$. For notational simplicity we define $p_i(k)$ (resp. $q_i(k)$) for all $0 \le i \le 4$ by simply setting $p_i(k) = 0$ (resp. $q_i(k) = 0$) if i is not a possible value. Note that there is no relation between $p_i(k)$ and $q_i(k)$ since these are probabilities conditional on exclusive events.

Definition 1. The *k-th average contrast* is defined as

$$\alpha(k) = \frac{\sum_{i=0}^{4} i \cdot p_i(k) - \sum_{i=0}^{4} i \cdot q_i(k)}{4}.$$

Note that $\sum_{i=0}^{4} i \cdot p_i(k)$ (resp. $\sum_{i=0}^{4} i \cdot q_i(k)$) is the average number of black subpixels in a decoded block corresponding to a black (resp. white) pixel in the secret image.

We first give the exact expression for $q_i(k)$ where possible values of i are 0 and 2, more precisely, if $k = 1, \ldots, n - 2$, then i can take value 0 or 2 (note that there is a possibility that all the k shares are the all-zero vectors), and if $k = n - 1, n$, then $i = 2$ (at least one of k shares has weight 2).

Lemma 1.

$$q_0(k) = \begin{cases} (1 - 2p)^k \binom{n-2}{k} \binom{n}{k}^{-1}, & k = 1, \ldots, n - 2, \\ 0, & k = n - 1, n; \end{cases}$$

$$q_2(k) = \begin{cases} 1 - q_0(k), & k = 1, \ldots, n - 2, \\ 1, & k = n - 1, n. \end{cases}$$

Similarly, we can give the exact expression for $p_i(k)$:

Lemma 2. *$p_0(k) = q_0(k)$, $k = 1, \dots, n$;*

$$p_2(k) = \begin{cases} 1 - p_0(1), & k = 1, \\ 2\big[(1-p)^{k-1}\binom{n-2}{k-1} & \\ \quad + \big((1-p)^k - (1-2p)^k\big) & \\ \quad \times \binom{n-2}{k}\big]\binom{n}{k}^{-1}, & k = 2, \dots, n-2, \\ 2n^{-1}(1-p)^{n-2}, & k = n-1, \\ 0, & k = n; \end{cases}$$

$$p_4(k) = \begin{cases} 0, & k = 1, \\ 1 - p_0(k) - p_2(k), & k = 2, \dots, n-1, \\ 1, & k = n. \end{cases}$$

Using Lemmas 1 and 2 we can compute the average contrast of the $(2, n)$-threshold progressive VCS.

Proposition 1. *A share generated by the scheme is secure. In particular, the first average contrast of the scheme is zero.*

Proof. Since $p_0(1) = q_0(1)$ and $p_2(1) = q_2(1)$ (and also $p_4(1) = 0$), an encryption of a black pixel has the same statistics as that of a white pixel and one cannot distinguish a black pixel encryption from a white one. Also from this fact the 1st average contrast of the scheme is shown to be 0. □

We compare our scheme with the Fang–Lin scheme [3] (see also [5, 6]), the first progressive VCS, in terms of average contrast. (The explicit expression for the average contrast of the Fang–Lin scheme is omitted for lack of space.) In Fig. 1 we show the average contrast of our $(2, 6)$-threshold progressive VCS based on the $(2, 2)$-threshold VCS with varying parameter p. For comparison we also show the average contrast of the Fang–Lin scheme. As shown in Proposition 1 a share generated by our scheme has average contrast 0 and the final stacking result has contrast 0.5, irrespective of the value of p, whereas a share of the Fang–Lin scheme has a nonzero value of average contrast and the final stacking result has average contrast less than 0.45.

Similarly, we can derive the explicit expression for the average contrast of the $(3, n)$-threshold progressive VCS based on the $(3, 3)$-threshold VCS, where $n \geq 3$, and show that the stacking of up to any two shares reveals nothing about a secret image, although the details are omitted for lack of space.

2.4. Extension to a Block-wise Progressive VCS

The scheme described in Section 2.2 encrypts each pixel in the secret image to a 2×2 block of subpixels. It is easy to extend the scheme to a block-wise encryption scheme. In the block-wise scheme we divide the secret image of size $N \times M$ into NM/L^2 blocks of size $L \times L$ where L divides N and M, and we take parameter p to be zero. The block-wise encryption algorithm is almost the same as the pixel-wise one except that the row permutation is fixed for each $L \times L$ block in the secret image, which leads to a jigsaw puzzle like decryption.

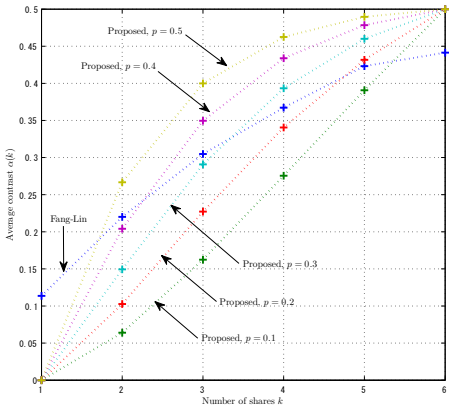


Fig. 1. Comparison of our scheme with the Fang–Lin scheme.

3. XOR-BASED PROGRESSIVE VCS WITHOUT PIXEL EXPANSION

The proposed scheme in the previous section has pixel expansion, which is undesirable in practical applications. In this section, assuming that optical devices are available, we give a progressive VCS without pixel expansion, which is based on the conventional XOR-based $(2, 2)$-threshold VCS (see, e.g., [7, Chap. 6]) whose basis matrices are given by

$$C_0 = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}, \quad C_1 = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}.$$

The construction of the extended basis matrices of the XOR-based $(2, n)$-threshold progressive VCS is almost the same as the OR-based $(2, n)$-threshold progressive VCS, where $n \geq 2$, except that parameter p is taken to be 0 (that is, we simply attach some zeros to a basis matrix), and so we omit the details. The encryption algorithm is also the same as that of the OR-based one. On the other hand, the decryption of an XOR-based scheme uses the XORing instead of the stacking (ORing).

We can define the average contrast of an XOR-based scheme in the same way as in the previous section. Let $p(k)$ (resp. $q(k)$) denote the probability that a decoded pixel corresponding to a black (resp. white) pixel in the secret image has a value of 1 (i.e., decoded pixel is black) after XORing k shares. Note that $p(n) = 1$ and $q(n) = 0$ are obvious from the construction. In fact, it is easy to show the following.

Lemma 3.

$$p(k) = \begin{cases} \binom{n-1}{k-1}\binom{n}{k}^{-1}, & k = 1, \dots, n-1, \\ 1, & k = n; \end{cases}$$

$$q(k) = \begin{cases} \binom{n-2}{k-1}\binom{n}{k}^{-1}, & k = 1, \dots, n-1, \\ 0, & k = n. \end{cases}$$

Note that $p(k) \geq q(k)$ for $k = 1, \ldots, n$. We define the k-th average contrast in the XOR-based scheme as $\alpha(k) = p(k) - q(k)$.

Proposition 2. *$\alpha(k) = \frac{k(k-1)}{n(n-1)}$ for $k = 1, \ldots, n$. In particular, $\alpha(1) = 0$. So a share generated by the scheme reveals nothing about a secret image.*

Proof. Use Lemma 3 and the definition of $\alpha(k)$. □

Note that $\alpha(k)$ is an increasing function of k. Since the proposed XOR-based progressive VCS has no pixel expansion and the final average contrast (i.e., conventional contrast) attains the maximum value of 1, it is advantageous against the OR-based scheme with a pixel expansion factor of 4 and a final average contrast of at most $3/4$. Furthermore, since XOR operations are easy to perform using the polarization of light, an XOR-based progressive VCS can be implemented with low complexity (see, e.g., [7, Chap. 6] and references therein).

Finally, we remark that an XOR-based progressive VCS can be extended to a block-wise progressive one in the same way as in Section 2.4.

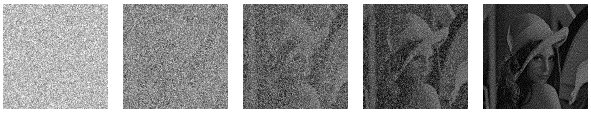
4. EXPERIMENTAL RESULTS

In this section we experimentally examine the following three points: (i) effect of parameter p of an OR-based progressive VCS on the decoding speed; (ii) effect of block-wise processing on the decoded image; and (iii) comparison of our XOR-based scheme with the Hou–Quan scheme [6]. We have implemented in Matlab our OR-based scheme constructed from the $(2, 2)$-threshold VCS that has contrast $1/2$. The secret image was a binary image obtained from the 256×256 grayscale image "Lena" by using the dither function of Matlab. Figures 2 and 3 show the results of our scheme with parameter $p = 0.1$ and $p = 0.5$, respectively. To save space we have rescaled the size of the images. The results show that if parameter p becomes large, the decoding accelerates as shown by the contrast analysis in Fig. 1, so we can control the decoding speed, which is an advantage of our scheme against previous progressive VCSs.

Note that the security of our scheme means secure shares and says nothing about more than one share (stacking two shares already reveals something about a secret image).

Figures 4 shows an example of a block-wise progressive VCS based on the $(3, 3)$-threshold VCS with contrast $1/4$. The block size was taken to be 16×16. It can be seen from the figures that the secret image is block-wise recovered.

We have also implemented our XOR-based scheme and the Hou–Quan (OR-based) scheme with $n = 5$, both of which have no pixel expansion. Figures 5 and 6 show the results of our scheme and the Hou–Quan scheme, respectively. Since these schemes are based on different mechanisms, the comparison is meaningless, however, if both are implemented on a computer, they are comparable.

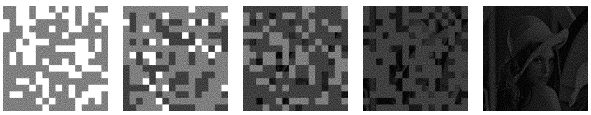


(a) (b) (c) (d) (e)

Fig. 2. Proposed $(2, 5)$-threshold progressive VCS with $p = 0.1$. (a) shows a share; (b), (c), (d) and (e) show the resulting images obtained from stacking k shares, $k = 2, 3, 4$, and 5(all shares), respectively.

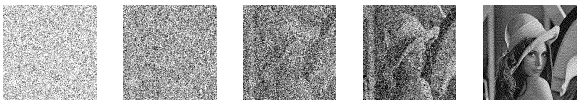
(a) (b) (c) (d) (e)

Fig. 3. Proposed $(2, 5)$-threshold progressive VCS with $p = 0.5$. Figures (a)-(e) correspond to those of Fig.2.



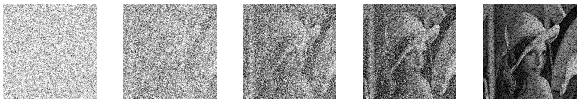
(a) (b) (c) (d) (e)

Fig. 4. Proposed $(3, 5)$-threshold block-wise progressive VCS. Figures (a)-(e) correspond to those of Fig.2.



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

Fig. 5. Proposed XOR-based progressive VCS with $n = 5$. Figures (a)-(e) correspond to those of Fig.2.



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

Fig. 6. The Hou–Quan scheme with $n = 5$. Figures (a)-(e) correspond to those of Fig.2.

5. CONCLUSION

In this paper we have presented some progressive VCSs using traditional threshold VCSs, i.e., the OR-based pixel/block-wise threshold progressive VCSs with pixel expansion 4 and contrast $1/4$, $1/2$, $3/4$, and the XOR-based scheme with no pixel expansion and contrast 1. Compared with the Fang–Lin scheme, our OR-based schemes can control the decoding speed and guarantee the contrast. The XOR-based scheme has an advantage of the contrast againt the Hou–Quan scheme, although their scheme is an OR-based one.

6. REFERENCES

[1] "(t, n)-Threshold progressive visual cryptography scheme," IEICE Tech. Rep., Vol. 113, No. 217, ISEC2013-55, pp. 29–36, presented at the IEICE Technical Meeting on Information Security, Tokyo, Japan, September 2013; "How to construct progressive visual cryptography schemes," presented in the Work-in-Progress track of the IEEE Workshop on Information Forensics and Security, Guangzhou, China, November 2013. (The technical reports of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan are non-reviewed preprints that are circulated only in Japan, and the IEICE permits authors to submit their papers presented at the IEICE Technical Meetings to other conferences/journals.)

[2] M. Naor and A. Shamir, "Visual cryptography," Advances in Cryptology—Eurocrypt'94, Vol. 950 of Lecture Notes in Computer Science, pp. 1–12, Springer, Berlin, 1995.

[3] W.-P. Fang and J.-C. Lin, "Progressive viewing and sharing of sensitive images," *Pattern Recognition and Image Analysis*, Vol. 16, No. 4, pp. 632–636, 2006.

[4] D. Jin, W.-Q. Yan, and M. S. Kankanhalli, "Progressive color visual cryptography," *J. Electron. Imag.*, Vol. 14, No. 3, pp. 1–13, 2005.

[5] W.-P. Fang, "Friendly progressive visual secret sharing," *Pattern Recognition*, Vol. 41, pp. 1410–1414, 2008.

[6] Y.-C. Hou and Z.-Y. Quan, "Progressive visual cryptography with unexpanded shares," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 21, No. 11, pp. 1760–1764, 2011.

[7] S. Cimato and C.-N. Yang (editors), Visual Cryptography and Secret Image Sharing, CRC Press, 2012.