Encrypting More Information in Visual Cryptography Scheme

Feng Liu[1], Peng Li[2] and ChuanKun Wu[1]
[1]State Key Laboratory Of Information Security,
Institute of Information Engineering,
Chinese Academy of Sciences, Beijing 100190, China
[2]Department of Computer Science and Technology,
Harbin Institute of Technology, Harbin 150001, China
Email: fengliu.cas@gmail.com, lipeng955@gmail.com
Homepage: http://iscas.ac.cn/∼liufeng/

Abstract

The visual cryptography scheme (VCS) is a scheme which encodes a secret image into several shares, and only qualified sets of shares can recover the secret image visually, other sets of shares cannot get any information about the content of the secret image. From the point of view of encrypting (carrying) the secret information, the traditional VCS is not an efficient method. The amount of the information that a VCS encrypts depends on the amount of secret pixels. And because of the restrictions of the human eyes and the pixel expansion and the alignment problem of the VCS, a VCS perhaps can only be used to encrypt a small secret image. VCS requires a random number generator to guide the generation of the shares. As we will show in this paper, the random input of VCS can ba seen as a subchannel which helps carrying more secret information. We propose a general method to increase the amount of secret information that a threshold VCS can encrypt by treating the pseudo-random inputs of the VCS as a subchannel, i.e. the Encrypting More Information Visual Cryptography Scheme (EMIVCS). We also study the bandwidth of the proposed EMIVCS. The disadvantage of the proposed scheme is that, the decoding process is computer aided. However, compared with other computer aided VCS, the proposed scheme is more efficient.
Keywords: Secret Sharing; Visual Cryptography; Covert data; Sub-channel

1 Introduction

Visual Cryptography Scheme (VCS) was first formally introduced by Naor and Shamir [1], which encodes a secret image into n shares (printed on transparencies) for n participants. The merit of VCS is that the decoding process is computation-free. In general, a (k, n)-threshold VCS, where $k \leq n$, encodes a secret image into n shares. The original image can be recovered by stacking any k shares visually. However any $k-1$ or fewer shares will gain no information other than the size of the secret image. The underlying operation of the stacking is the logic operation *OR*. Many studies focused on the novel applications of VCS [2–5]. Recently, a book covering an extensive range of topics related to VCS is published [6].

For the traditional VCS, the amount of secret information it encrypts depends on the amount of secret bits (pixels) in the recovered secret image. However, because of the following three reasons, the amount of secret information is severely constrained: the first is the big pixel expansion of the shares, this implies that the secret image cannot be too big

(a big transparency is inconvenient for the participant). And second, the human eyes can only identify the patterns in the secret image when the patterns are clear enough, i.e. the lines and dots in the patterns should be a block of pixels rather than a single pixel. Third, because of the alignment problem [7], the pixels in the shares cannot be too small. So, for the traditional VCS, it perhaps can only be used to encrypt a small secret image. Many studies try to increase the amount of the information that the VCS encrypts, such as to encrypt a plural number of secret images in one VCS [8, 9] and the method that uses rotate shares [10], and the color VCS [11, 12]. However, these methods do not increase the amount of secret bits encrypted if we consider the ratio $R = \frac{t}{m}$, where t is the number of secret bits that are encrypted by every m sub-pixels, and for the color VCS, it usually degrades the recovered secret image severely. In this paper, we measure the amount of secret information of a VCS by the amount of the secret bits it encrypts.

The VCS requires a random number generator to guide the generation of the shares. In the real application, we make use of pseudo-random number generator for the random inputs. The main idea of our scheme is that, the pseudo-random inputs of the VCS can be used to carry covert data, and the outputs (ciphertext) of the symmetric encryption algorithms (in practical we can choose the AES, Twofish etc.) can be considered as the pseudo-random inputs of the VCS, hence increase the amount of the secret information encrypted by the VCS. By using Shamir's [13] secret sharing scheme, the encryption key is shared into n sub-keys, which are concatenated with the corresponding shares. We call such scheme the Encrypting More Information Visual Cryptography Scheme (EMIVCS). The secret information encrypted by the EMIVCS contains two parts: the secret image and the covert data encrypted in the random input. For example, for the EMIVCS in Example 1 with share size 129×128, we can encrypt additional 2^{13} bits covert data besides the secret image S_I.

However, to decrypt the covert data, we need to employ computational devices, i.e. EMIVCS is computer aided. By a (k, t, n)-EMIVCS ($k \leq t$), we mean a VCS that carries additional covert data where any k out of n participants can visually recover the secret image by stacking their shares, and any t out of n participants can get the additional covert data by computation. There are two other computer aided VCS's [14, 15], called two-in-one image secret sharing scheme (TiOISSS), which can reveal a vague image by stacking the shares and reveal a much finer gray image by computation. The comparisons between the EMIVCS and two TiOISSS's will be given in Section 4. Yang et al. [5, 16] also tried to make use of the pseudo-random inputs to carry confidential data. Unfortunately, their scheme is only for (2,2) access structure and is also computer aided when decoding the confidential data.

The proposed (k, t, n)-EMIVCS is a multi-threshold secret sharing scheme. A minority participants with at least k members can share one secret image, whereas a majority participants with at least t members can access an additional secret information. The EMIVCS can be applied to many scenarios. We give two examples.

First, consider the scenario of the cheating problem [17], i.e. the cheaters show false shares during recovering the secret image. So, in order to verify the validity of the shares, one will require additional authentication information. The covert data of the EMIVCS can be used to take the authentication information. In such a case, if the participants doubt the authenticity of the recovered secret image, they can convene $t(\geq k)$ participants to verify the shares.

Second, imagine that some soldiers are attending a secret meeting in the battle field,

each one holds a share, the recovered secret image of the shares is the secret order from the commander, and there also is covert data which is the detailed plan of action. Because that in the battle field their decryption device may be destroyed, in such cases, the soldiers can at least get the secret order from their commander. And if their decryption device is available, then they can get the detailed plan of action.

There are other message delivering method requiring some decoding devices, such as some standard 2D encoding methods. By comparing the EMIVCS and the 2D encoding methods [18], we have that, for the 2D encoding method, the decoding of the secret information totally relies on the decoding device. Hence, if the participants are in a scenario where there is no decoding device, they cannot extract any information. But with our method, the participants can stack the shares and get part of the secret information. Hence from this point of view, the EMIVCS can have wider application scenarios.

In this paper, we propose a specific construction of general (k, t, n)-EMIVCS by taking the VCS proposed in [9] and secret sharing scheme in [13] as the building block. We also studied some related problems about EMIVCS such as the pseudo-randomness that the input of VCS requires, and the sufficient conditions to uniquely determine a share matrix, and the bandwidth of the proposed EMIVCS, and we also proposed an efficient decoding method to decode the EMIVCS. At last, comparisons with other computer aided VCS's are given such as TiOISSS in [14, 15].

This paper is organized as follows. In the next section, we give some preliminary results. And in Section 3, we propose a general construction of the EMIVCS based on the construction of VCS in [9] and study some related problems about EMIVCS. In Section 4, we compare the proposed scheme with TiOISSS. At last we give a short conclusion in Section 5.

2 Preliminaries

In this section, we give some definitions about VCS, and introduce the Droste's [9] construction of (k, n)-VCS and Shamir's [13] secret sharing scheme (i.e. polynomial-based secret sharing scheme, PSSS), which are building blocks of our scheme.

2.1 VCS

We restrict ourselves to images only consisting of black and white pixels, where we denote by 1 for a black pixel and 0 for a white pixel. In this paper, we consider only the threshold (k, n)-VCS. For a vector $v \in GF^m(2)$, we denote by $w(v)$ the Hamming weight of the vector v. A (k, n)-VCS, denoted by (C_0, C_1), consists of two sets (pairwise different collection) of $n \times m$ Boolean matrices C_0 and C_1. To encrypt a white (*resp.* black) pixel, a dealer (the one who sets up the system) randomly chooses one of the share matrices (in the practical sense, the dealer can only choose the share matrices pseudo-randomly), in C_0 (*resp.* C_1) and distributes its rows (shares) to the n participants of the scheme. More precisely, we give a formal definition of the (k, n) VCS as follows:

Definition 1 *Let k, n, m, l and h be nonnegative integers satisfying $2 \leq k \leq n$ and $0 \leq l < h \leq m$. The two sets of $n \times m$ Boolean matrices (C_0, C_1) constitute a (k, n)-VCS if the following properties are satisfied:*

1. *(Contrast) For any $s \in C_0$, the OR of any k out of the n rows of s, is a vector v that,*

satisfies $w(v) \leq l$.

2. (*Contrast*) *For any* $s \in C_1$, *the OR of any* k *out of the* n *rows of* s, *is a vector* v *that, satisfies* $w(v) \geq h$.

3. (*Security*) *For any* $i_1 < i_2 < \cdots < i_t$ *in* $\{1, 2, \cdots, n\}$ *with* $t < k$, *the two collections of* $t \times m$ *matrices* F_j *for* $j \in \{0, 1\}$, *obtained by restricting each* $n \times m$ *matrix in* C_j *to rows* $i_1, i_2, \cdots i_t$, *are indistinguishable in the sense that they contain the same matrices with the same frequencies.*

In the above definition, m is called the pixel expansion of the shares. A pixel of the original secret image is represented by m sub-pixels in the recovered secret image. In general, we are interested in schemes with m being as small as possible.

In Definition 1, the first two properties ensure that any k participants will be able to distinguish the black and white pixels, and the third property ensures the security of the scheme that any $k-1$ or fewer participants can gain no information about the content of the secret image.

In order to share a complete image, the pixel scheme has to be applied to all the pixels in the image. In the traditional VCS models, the encryption is applied to the secret pixels one at a time. However, we extend this method to encrypt q secret pixels at a time, and call this model the *q-pixel encryption model*. The traditional model is the 1-*pixel encryption model*. The difference between the 1-*pixel encryption model* and the *q-pixel encryption model* is that: in the 1-*pixel encryption model*, the dealer generates one pseudo-random number which guides the choice of a share matrix at a time. However, in the *q-pixel encryption model*, the dealer generates one pseudo-random number which guides the choice of q share matrices at a time.

We consider VCS where C_0, C_1 are constructed from a pair of $n \times m$ matrices M_0, M_1, which are called *basis matrices*. The set C_i $(i = 0, 1)$ consists of the matrices obtained by permuting all the columns of M_i. This approach of VCS construction will have small memory requirements (it only keeps the basis matrices) and high efficiency (to choose a matrix in C_0 $(resp.\ C_1)$ as it only needs to generate a permutation of the basis matrix). We will use the basis matrices to simplify the discussions in this paper. And when the set of a VCS C_0 $(resp.\ C_1)$ can be generated by the basis matrix, we call such VCS as the *basis matrix VCS*. Many studies in the literatures proposed the construction of the *basis matrix VCS*, such as [9, 19, 20].

Recall that, by definition, the share matrices in C_0 $(resp.\ C_1)$ are pairwise different. Denote the different columns in the basis matrix M_i as $c_1, c_2, \cdots, c_e$ and the multiplicities of these columns are $a_1, a_2, \cdots, a_e$, we have that the number of share matrices in C_i is $|C_i| = \frac{(\sum_{i=1}^{e} a_i)!}{\prod_{i=1}^{e} a_i!}$, for $i \in \{0, 1\}$ (these share matrices are pairwise different). In order to choose a share matrix in C_i pseudo-randomly, the length of the pseudo-random input for one secret pixel should be at least $\log_2 |C_i|$ bits.

2.2 Droste's construction of (k, n)-VCS

In this paper, we take the construction of Droste [9] as the building block, and we recall his construction as follows:

Construction of (k, n)-VCS proposed in [9]:

Setup Let M_0 and M_1 be two empty matrices, where the basis matrices M_0 and M_1 are considered as the collections of their columns;

step 1 For all even $p \in \{0, 1, \ldots, k\}$, call **ADD**($p$,$M_0$);

step 2 For all odd $p \in \{0, 1, \ldots, k\}$, call **ADD**($p$,$M_1$);

step 3 Define P_0 (resp. P_1) be the collection consisting of all columns of every restriction of k rows of M_0 (resp. M_1), and define S_0 (resp. S_1) be the set consisting of all k-length boolean columns with an even (resp. odd) number of 1's. Define the remaining of M_0 (resp. M_1) be $P_0 \backslash S_0$ (resp. $P_1 \backslash S_1$), and define the *rest* of M_0 (resp. M_1) be the columns in the remaining of M_0 (resp. M_1), but not in the remaining of M_1 (resp. M_0), i.e. the *rest* of M_0 is $\{P_0 \backslash S_0\} \backslash \{P_1 \backslash S_1\}$ and the *rest* of M_1 is $\{P_1 \backslash S_1\} \backslash \{P_0 \backslash S_0\}$. If the *rests* are not empty:

(a) If p is an even number, add to M_0 all columns adjusting the *rest* of M_1 by calling **ADD**(p,M_0), where p is the number of 1's in column l, and $l \in \{P_1 \backslash S_1\} \backslash \{P_0 \backslash S_0\}$.

(b) If p is an odd number, add to M_1 all columns adjusting the *rest* of M_0 by calling **ADD**(p,M_1), where p is the number of 1's in column l, and $l \in \{P_0 \backslash S_0\} \backslash \{P_1 \backslash S_1\}$.

where the subroutine ADD is: **ADD**(p,M)

1 If $p \leq k - p$, add every column with $q = p$ 1's to M.

2 If $p > k - p$, add every column with $q = p + n - k$ 1's to M.

2.3 PSSS

Shamir[13] introduced the (t, n)-PSSS ($t \leq n$) to share the secret data into n shares. Any t shares can be used to reconstruct the secret, but any $t - 1$ or less shares get no information about the secret. To divide the secret data into shares, it randomly generates a $(t - 1)$-degree polynomial using modular arithmetic:

$$f(x) = (a_0 + a_1 x^1 + \ldots + a_{t-1} x^{t-1}) \bmod p$$

in which a_0 is replaced by the secret data, p is a prime number larger than a_0 and n. The coefficients $a_1, a_2, \ldots, a_{t-1}$ are randomly chosen from a uniformly distribution over the integers in $[1, p)$. Then we could generate n shares $(x_i, f(x_i)), i = 1, 2, \ldots, n$. Later, with any t out of the n shares, one can evaluate all the coefficients, particularly the coefficient a_0, of the polynomial $f(x)$ by Lagrange's interpolation. However, any $t - 1$ or fewer shares cannot get any information about the secret.

3 EMIVCS

In this section, we first propose a construction of EMIVCS by taking the pseudo-random inputs as a subchannel, and then study some related problems about the EMIVCS: first, the pseudo-randomness that the input of VCS requires; second, the sufficient conditions to uniquely determine a share matrix in the set C_i for $i = 0, 1$; third, the bandwidth of the subchannel; fourth, the decoding method to decode the ciphertext of EMIVCS.

3.1 Construction of EMIVCS

The main idea of the proposed scheme is to treat the symmetric encryption algorithm as the pseudo-random generator of VCS. Thus the VCS can, naturally, carry the additional covert data encrypted by the symmetric encryption algorithm. In this paper, we take the VCS proposed in [9] as the building block. And in practical application, the symmetric encryption algorithm can be the AES, Twofish etc. The cipher block chaining (CBC) [21] encryption mode is employed. The encryption key S_{Key} in EMIVCS is shared by (t, n)-PSSS into n sub-keys $SK_1, SK_2, \ldots, SK_n$. Therefore, any t or more sub-keys can be used to reveal the secret key, while any $t-1$ or less sub-keys cannot get any information about the secret key.

Before showing the construction, we need to give an assumption that: the participants all know the access structure they belong to, i.e. the i-th participant knows by himself that he is the i-th participant. Usually, the access structure of a VCS is not secret information, so this assumption is reasonable.

Construction 1

Encryption process:

Input: *The secret image S_I, covert data $S_{plaintext}$ and the secret key S_{Key}.*

Output: *n shares.*

Step 1: *Encrypt the covert data $S_{plaintext}$ by using the key S_{Key},$S_{ciphertext} = En(S_{Key}, S_{plaintext})$;*

Step 2: *Encrypt the secret image S_I into n shares $V_1, V_2, \ldots, V_n$ by using the (k, n)-VCS, where the encrypted data from the Step 1 is employed as the pseudo-random input of the (k, n)-VCS;*

Step 3: *Share S_{Key} into n sub-keys $SK_1, SK_2, \ldots, SK_n$ by using (t, n)-PSSS, then convert these sub-keys into binary images $I_1, I_2, \ldots, I_n$, and concatenate I_i $(i = 1, 2, \ldots, n)$ with share V_i to get the final share S_i.*

Decryption process:

Input: *Any t shares where $k \leq t$.*

Output: *The secret image S_I and the covert data $S_{plaintext}$.*

Step 1: *Stack any k shares to get the recovered secret image S_I;*

Step 2: *Determine the share matrices which are used to encrypt the secret image for each pixel by t shares, and hence get the ciphertext $S_{ciphertext}$;*

Step 3: *Extract t sub-keys from t shares, then reconstruct the secret key S_{Key} by Lagrange's interpolation.*

Step 4: *Decrypt the ciphertext $S_{ciphertext}$ by using the S_{Key}, $S_{plaintext} = De(S_{Key}, S_{ciphertext})$.*

Figure 1 and Figure 2 illustrate the procedures of Construction 1.

Remarks: In the practical application, the key length of the AES and Twofish, usually, is 128 bits. Therefore, each sub-key is generated and converted into a 128 bits binary image which only takes a small area in the share.

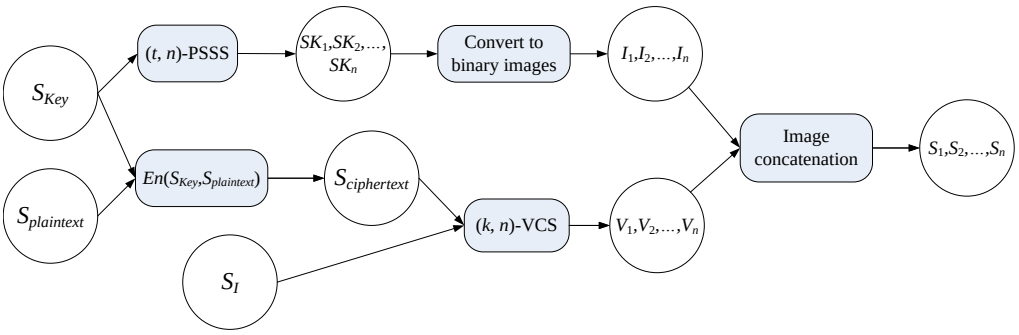

Figure 1: The encryption process of the (k, t, n)-EMIVCS

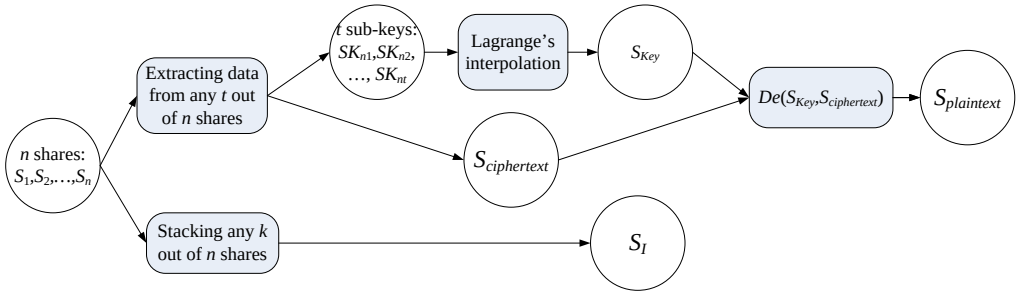

Figure 2: The decryption process of the (k, t, n)-EMIVCS

For the (k, t, n)-EMIVCS, by stacking k shares we can reconstruct the secret image S_I. And if one obtains t rows, he can uniquely determine a share matrix and hence obtain the ciphertext, where "can uniquely determine a share matrix" means that there only exists one share matrix in C_i $(i = 0, 1)$ that contains these t rows (and "cannot uniquely determine" means there exist more than one share matrices that contain these t rows, hence we cannot determine which one is chosen by the dealer when encrypting the secret pixel). In another word, in order to get the ciphertext one needs t shares.

The security of the (k, t, n)-EMIVCS are based on the security of the symmetric encryption algorithm, the security of VCS and the security of PSSS. Particularly, if an attacker wants to know the secret image, he needs at least k shares; if he wants to know the covert data encrypted by the symmetric encryption algorithm, he needs at least t shares to extract the ciphertext and the secret key.

The VCS requires inputting pseudo-random numbers to guide the choice of the share matrices. Denote the share matrices in C_i as $S_0^i, \cdots, S_{|C_i|-1}^i$, and denote $P(S_j^i)$ for $i = 0, 1$ and $j = 0, 1, \cdots, |C_i| - 1$ as the probability that choosing the share matrix S_j^i. Hence the inputted of the pseudo-random numbers should guarantee that

$$P(S_0^i) = P(S_1^i) = \cdots = P(S_{|C_i|-1}^i) \quad \cdots\cdots\cdots\cdots \quad (1)$$

In order to choose a share matrix pseudo-randomly in C_i, the dealer needs at least

$\log_2 |C_i|$ bits pseudo-random numbers (we will consider the case that $\log_2 |C_i|$ is not an integer in a later time). Denote $B(j)$ as the binary representation of integer j with length $\log_2 |C_i|$, i.e. $B(j)$ is the binary string that represents j. Without loss of generality, we assume that when the inputted pseudo-random number is $B(j)$, the dealer chooses the share matrix S_j^i to encrypt the secret pixel i. And denote $P(B(j))$ as the probability of generating the binary string $B(j)$ by the pseudo-random generator. According to the equation (1), we have

$$P(B(0)) = P(B(1)) = \cdots = P(B(|C_i| - 1)) \quad \cdots\cdots\cdots\cdots \quad (2)$$

In fact the ciphertexts of the AES and Twofish satisfy the equation (2), because they have passed the serial test in [22]. Hence, we can take the AES and Twofish as the pseudo-random generator. (This also is the very reason that we do not use the covert data directly to guide the generation of shares.)

To make things clearer, we give the following example for $(2, 2, 2)$-EMIVCS:

Example 1 *The sets of share matrices of* $(2, 2, 2)$*-EMIVCS are as follows:*

$$C_0 = \left\{ \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \begin{bmatrix} 01 \\ 01 \end{bmatrix} \right\} \text{ and } C_1 = \left\{ \begin{bmatrix} 10 \\ 01 \end{bmatrix}, \begin{bmatrix} 01 \\ 10 \end{bmatrix} \right\}$$

The principle of choosing share matrix is that: if the pseudo-random input is 0*, we choose the first share matrix in* C_0 *or* C_1*, and if the pseudo-random input is* 1*, we choose the second.*

Figure 3 gives an illustration for the $(2, 2, 2)$*-EMIVCS.*

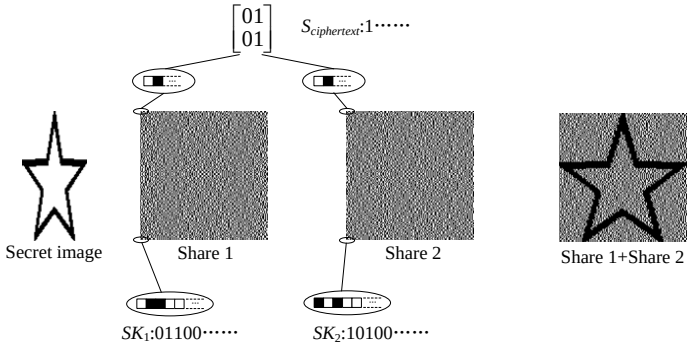

Figure 3: The procedure of the $(2, 2, 2)$-EMIVCS

A secret image with 64×128 *pixels is encoded into Share 1 and Share 2. The size of the shares and the recovered secret image is* 129×128*. Since the length of each sub-key is* 128 *bits, it only takes one line at the bottom of each share to attach the sub-key. The length of the ciphertext* $S_{ciphertext}$ *encrypted in the shares is* 2^{13} *bits, i.e. the subchannel can be used to carry extra* 2^{13} *bits of covert data. In the first step of the reconstruction process, the secret image can be visually decoded by stacking two shares. In the second step, two sub-keys are extracted from the last row of two shares, and then we can reveal the secret key* S_{Key} *by Lagrange's interpolation. With further observation of the share blocks of two shares, the ciphertext can be obtained by the uniquely determined share matric by share blocks. For*

example, the first block of share 1 *is constituted by two sub-pixels '0' and '1', and the first block of share* 2 *is also constituted by two sub-pixels '0' and '1'. Therefore, we can determine the share matrix, which is the second share matrix* C_0*, and the recovered ciphertext is '1'. Finally, we get the covert data* $S_{plaintext}$ *by decrypting the ciphertext* $S_{ciphertext}$*.* □

3.2 Uniquely determine a share matrix

For the (n, n)-VCS, if one has all the n shares, he can uniquely determine the share matrices used when encrypting the secret image S_I, and hence get to know the ciphertext.

We then focus our discussion on the (k, n)-VCS with $k < n$: we find that, for the VCS in Section 2, $n-1$ rows can uniquely determine a share matrix in the set C_0 (resp. C_1). The following theorem shows this result:

Theorem 1 *Denote* M_0 *and* M_1 *be the basis matrices constructed by* (k, n)*-VCS in [9], and denote* C_0 *and* C_1 *be the sets of share matrices generated from* M_0 *and* M_1*, respectively. If every* t *rows of a share matrix in* C_i $(i = 0, 1)$ *can uniquely determine a share matrix in* C_i*, then* $t \geq n - 1$*.*

Proof: First, for the case of $t = n$, it obviously can uniquely determine an n-row matrix from its all n rows.

Second, we show any $n-1$ rows can uniquely determine a share matrix. According to the construction in [9], the number of the 1's of each column in the basis matrix M_0 is from the set $T_0 = \{a | 0 \leq a \leq \lfloor \frac{k}{2} \rfloor, a \mod 2 = 0\} \bigcup \{a + n - k | \lfloor \frac{k}{2} \rfloor < a \leq k, a \mod 2 = 0\}$, and the number of 1's of each column in the basis matrix M_1 is from the set $T_1 = \{a | 0 \leq a \leq \lfloor \frac{k}{2} \rfloor, a \mod 2 = 1\} \bigcup \{a + n - k | \lfloor \frac{k}{2} \rfloor < a \leq k, a \mod 2 = 1\}$. (here and hereafter, $\lfloor x \rfloor$ is the largest integer that is no larger than x, and $\lceil x \rceil$ is the smallest integer no smaller than x.)

Because $k < n$, when one has $n-1$ rows of a share matrix M, he can stack k shares and hence knows the secret pixel. Without loss of generality, suppose the secret pixel is black. We determine the last row of the share matrix M as follows: for the column p_i of M, where $i \in \{1, \cdots m\}$, denote the number of 1's of the $n-1$ rows in column p_i as h, then we have the entry of the last rows of column p_i be 0 if $h \in T_1$ and be 1 if $h + 1 \in T_1$. Hence, the last row can be uniquely determined by the $n-1$ rows, and because the participants know the access structure they belong to, then the share matrix can be uniquely determined.

Third, we prove any $n-2$ rows cannot uniquely determine a share matrix. Consider the construction in [9], we have that the basis matrix M_1 contains all the columns with hamming weight equaled to 1. Let A be a share matrix in C_1. Without loss of generality, there exist two different columns c_1 and c_2 in A, whose hamming weights are equaled to 1. Denote the position of 1 in column c_1 (resp. c_2) be p_1 (resp. p_2), we have $p_1 \neq p_2$. Let $X = \{1, 2, \ldots, n\} \backslash \{p_1, p_2\}$. Then, by restricting all the rows of columns c_1 and c_2 in X, we get two same sub-columns. Suppose B is a matrix generated by exchanging the positions of columns c_1 and c_2 in A, then B is also a share matrix in C_1. Therefore, by restricting all the rows of A and B in X, we get two same sub-matrices. In another word, the $n-2$ rows of share matrix A (the rows restricted in X) cannot uniquely determine a share matrix. Obviously, it also cannot uniquely determine a share matrix from less than $n-2$ rows. □

Theorem 1 gives an explicit method to uniquely determine a share matrix in C_i $(i = 0, 1)$, and in light of the above discussion, we have the following theorem:

Theorem 2 *Let $t = n - 1$, then Construction 1 generates a $(k, n-1, n)$-EMIVCS.* □

For general basis matrix visual cryptography (C_0, C_1), denote C_i^{All} as the set of all the possible columns that appear in the share matrices of C_i $(i = 0, 1)$. For any set of participants $X \subseteq P$, denote M' as a sub-matrix which is generated by restricting to the rows in X of a share matrix in C_i. First we have the following lemma:

Lemma 1 *For every column c' of M', if there exists only one column $c \in C_i^{All}$ such that $c[X] = c'$, then the sub-matrix M' can uniquely determine a share matrix in C_i, where $c[X]$ is the sub-column generated by restricting to the rows in X of c.*

Proof: Reduction to absurdity: Suppose M' cannot uniquely determine a share matrix in C_i, i.e. there exist two different share matrices, denoted by M_a and M_b, such that $M_a[X] = M_b[X] = M'$, where $M_a[X]$ is the sub-matrix generated by restricting to the rows in X of M_a. Since M_a and M_b are different share matrices, there exists at least one column that is different for M_a and M_b. Denote this column in M_a is c_a and that in M_b is c_b, i.e. $c_a \neq c_b$. And because $M_a[X] = M_b[X]$, we have $c_a[X] = c_b[X]$, which is contradict to the assumption that there exists only one column $c \in C_i^{All}$ such that $c[X] = c'$. Hence, M' can uniquely determine a share matrix in C_i. □

According to Lemma 1, we give a general discussion for basis matrix (k, n)-VCS, denote $c_p, c_q \in C_i^{All}$ as two different columns, and denote $X_{pq}^i (\subset P)$ as the set of the participants such that for each $x \in X_{pq}^i$ satisfying $c_p[x] = c_q[x]$, where $c_p[x]$ is the x-th entry of c_p. Then we have the following theorem:

Lemma 2 *Let $t = max\{|X_{pq}^i| + 1\}$ for $p \neq q$, $1 \leq p, q \leq m$ and $i = 0, 1$, then a sub-matrix of t rows of a share matrix in C_i can uniquely determine a share matrix in C_i.*

Proof: Let c' be a column of the sub-matrix M' which is generated by restricting t rows of a share matrix in C_i $(i = 0, 1)$. Denote the set of the participants of these t rows as X, i.e. $|X| = t$, where $t = max\{|X_{pq}| + 1\}$. We prove that, there only exists one column c of M such that $c[X] = c'$.

Reduction to absurdity: Suppose there exist two columns c_a and c_b such that $c_a[X] = c_b[X] = c'$. We have that c_a and c_b have t entries that have the same values, i.e. $t = |X_{ab}|$, which is impossible because $t = max\{|X_{pq}| + 1\}$ which implies $t > |X_{ab}|$.

According to Lemma 1, we have that a sub-matrix M' with t rows can uniquely determine a share matrix in C_i. □

For a (k, n)-VCS, any $k-1$ or less shares cannot get any information of the secret image. In another word, any $t (t < k)$ shares cannot decide the t-row sub-matrix is from C_0 or C_1, and hence can not uniquely determine the share matrix. Therefore, it is reasonable to assume $t \geq k$. Further with Lemma 2, we get the following theorem immediately:

Theorem 3 *For a basis matrix (k, n)-VCS, there exists a (k, t, n)-EMIVCS where $t = max\{k, |X_{pq}^i| + 1\}$, $p \neq q$, $1 \leq p, q \leq m$ and $i = 0, 1$.* □

According to Theorem 3, we also examined other two known constructions of (k, n)-VCS in [20, 23], and found that, the two constructions both have $t = n - 1$ too (the same as the

results in Theorem 1). Because they both take the canonical matrices as building block, where the canonical matrices means the matrices that all the columns of a given weight occur with the same frequency. And for the canonical matrices which has a column c_i with x 1's and $n-x$ 0's where $0 < x < n$, there exists a column c_j such that have only two entries are different from c_i, which implies $|X_{ij}| = n-2$, and hence $t = n-1$.

3.3 Bandwidth of EMIVCS

We define the bandwidth of EMIVCS as the maximum amount of covert data it carries through its subchannel. In this section, we show the bandwidth of the EMIVCS.

Denote the columns in the basis matrix M_i as $c_1, \cdots, c_e$ and the multiplicities of these columns are $a_1, \cdots, a_e$, recall that, we have the number of share matrix in C_i being $|C_i| = \frac{(\sum_{i=1}^{e} a_i)!}{\prod_{i=1}^{e} a_i!}$, for $i \in \{0,1\}$. To choose a share matrix in C_i, one needs at least $\log_2 |C_i|$ pseudo-random bits theoretically. And by determining the share matrix which is chosen when encrypting the secret image in C_i, one can determine at most $\log_2 |C_i|$ bits information theoretically. Hence, the amount of the additional covert data that can be carried by the secret pixel i is at most $\log_2 |C_i|$ bits theoretically. We list the number of the share matrices $|C_i|$ of the VCS constructed in [9] in the Table 1 and 2 as follows:

k \ n	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10
3		4!	$\frac{6!}{2!}$	$\frac{8!}{3!}$	$\frac{10!}{4!}$	$\frac{12!}{5!}$	$\frac{14!}{6!}$	$\frac{16!}{7!}$	$\frac{18!}{8!}$
4			8!	$\frac{15!}{3!2!}$	$\frac{24!}{6!3!}$	$\frac{35!}{10!4!}$	$\frac{48!}{15!5!}$	$\frac{63!}{21!6!}$	$\frac{80!}{28!7!}$
5				16!	$\frac{30!}{3!(2!)^6}$	$\frac{48!}{6!(3!)^7}$	$\frac{70!}{10!(4!)^8}$	$\frac{96!}{15!(5!)^9}$	$\frac{126!}{21!(6!)^{10}}$
6					32!	$\frac{70!}{4!(2!)^{21}3!}$	$\frac{128!}{10!(3!)^{28}6!}$	$\frac{210!}{20!(4!)^{36}10!}$	$\frac{320!}{35!(5!)^{45}15!}$
7						64!	$\frac{140!}{4!(2!)^{28}(3!)^8}$	$\frac{256!}{10!(3!)^{36}(6!)^9}$	$\frac{420!}{20!(4!)^{45}(10!)^{10}}$
8							128!	$\frac{315!}{5!(3!)^{36}(2!)^{36}4!}$	$\frac{640!}{15!(6!)^{45}(3!)^{45}10!}$
9								256!	$\frac{630!}{5!(3!)^{45}(2!)^{120}(4!)^{10}}$
10									512!

Table 1: The number of share matrices in C_0

Actually, in the practical sense, a pseudo-random number generator can only generate integer number of pseudo-random bits, and the ciphertexts are also represented by integer number of bits. However, the values of $\log_2 |C_i|$ are rarely integers, which means that some share matrices cannot be chosen by integer number of the pseudo-random bits, and it is hard to determine all the $\log_2 |C_i|$ ciphertext bits, hence results in the waste of the pseudo-random bit resources. So in the practical sense, the amount of the covert data carried by the EMIVCS cannot reach the theoretical value.

In fact, if one encrypts the secret pixels one at a time, in order to choose a share matrix pseudo-randomly in C_i, one needs at least $\lceil \log_2 |C_i| \rceil$ pseudo-random bits, and the length of the ciphertext can be at most $\lfloor \log_2 |C_i| \rfloor$ bits. To fully make use of the pseudo-random bits resources, we propose to encrypt q secret pixels at a time, i.e. the *q-pixel encryption model*. Let $q = a_0 + a_1$, where denote a_0 as the number of white secret pixels and a_1 as the number of black secret pixels, the effectiveness of using *q-pixel encryption model* rather

than 1-*pixel encryption model* is as follows:

k \ n	2	3	4	5	6	7	8	9	10
2	2!	3!	4!	5!	6!	7!	8!	9!	10!
3		4!	$\frac{6!}{2!}$	$\frac{8!}{3!}$	$\frac{10!}{4!}$	$\frac{12!}{5!}$	$\frac{14!}{6!}$	$\frac{16!}{7!}$	$\frac{18!}{8!}$
4			8!	$\frac{15!}{(2!)^5}$	$\frac{24!}{(3!)^6}$	$\frac{35!}{(4!)^7}$	$\frac{48!}{(5!)^8}$	$\frac{63!}{(6!)^9}$	$\frac{80!}{(7!)^{10}}$
5				16!	$\frac{30!}{3!(2!)^6}$	$\frac{48!}{6!(3!)^7}$	$\frac{70!}{10!(4!)^8}$	$\frac{96!}{15!(5!)^9}$	$\frac{126!}{21!(6!)^{10}}$
6					32!	$\frac{70!}{(3!)^7(2!)^7}$	$\frac{128!}{(6!)^8(3!)^8}$	$\frac{210!}{(10!)^9(4!)^9}$	$\frac{320!}{(15!)^{10}(5!)^{10}}$
7						64!	$\frac{140!}{4!(2!)^{28}(3!)^8}$	$\frac{256!}{10!(3!)^{36}(6!)^9}$	$\frac{420!}{20!(4!)^{45}(10!)^{10}}$
8							128!	$\frac{315!}{(4!)^9(2!)^{84}(3!)^9}$	$\frac{640!}{(10!)^{10}(3!)^{120}(6!)^{10}}$
9								256!	$\frac{630!}{5!(3!)^{45}(2!)^{120}(4!)^{10}}$
10									512!

Table 2: The number of share matrices in C_1

First: the number of pseudo-random bits required to choose the share matrices when use the *q-pixel encryption model* is $\lceil a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rceil$, and satisfies

$$\lceil a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rceil \leq a_0 \lceil \log_2 |C_0| \rceil + a_1 \lceil \log_2 |C_1| \rceil$$

which implies less pseudo-random bits are required by using the *q-pixel encryption model* than the 1-*pixel encryption model*.

Second: the number of pseudo-random bits determined by the share matrices when encrypt q secret pixels at one time is $\lfloor a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rfloor$, and satisfies

$$\lfloor a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rfloor \geq a_0 \lfloor \log_2 |C_0| \rfloor + a_1 \lfloor \log_2 |C_1| \rfloor$$

which implies more pseudo-random bits can be determined by using the *q-pixel encryption model* than the 1-*pixel encryption model*.

An problem for the *q-pixel encryption model* is that, when encrypt more secret pixels at a time, the encryption scheme becomes more complex. So there exists a trade-off for the value of the q.

To make things clearer, we give the following example for a $(2, 2, 3)$-EMIVCS:

Example 2 *For the sets*

$$C_0 = \left\{ \begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix}, \begin{bmatrix} 010 \\ 010 \\ 010 \end{bmatrix}, \begin{bmatrix} 001 \\ 001 \\ 001 \end{bmatrix} \right\} \text{ and}$$

$$C_1 = \left\{ \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}, \begin{bmatrix} 100 \\ 001 \\ 010 \end{bmatrix}, \begin{bmatrix} 010 \\ 100 \\ 001 \end{bmatrix}, \begin{bmatrix} 010 \\ 001 \\ 100 \end{bmatrix}, \begin{bmatrix} 001 \\ 100 \\ 010 \end{bmatrix}, \begin{bmatrix} 001 \\ 010 \\ 100 \end{bmatrix} \right\}$$

We have that, in the theoretical sense, the amount of information bits that can be carried by a white secret pixel is $\log_2 |C_0| = \log_2 3$ *and by a black secret pixel is* $\log_2 |C_1| = \log_2 6$. *And for 10 secret pixels with 5 white secret pixels and 5 black secret pixels the value will be* $5 \log_2 3 + 5 \log_2 6 \approx 20.85$.

However, in the practical sense, the 10-pixel encryption model*, where take* $a_0 = 5$ *and* $a_1 = 5$ *as example, we have the amount of information that can be carried is* $\lfloor \log_2 3^5 + \log_2 6^5 \rfloor = 20$*, which is more than the* 1-pixel encryption model*, where the corresponding value is* $5\lfloor \log_2 3 \rfloor + 5\lfloor \log_2 6 \rfloor = 15$. □

At this point, we can calculate the bandwidth of the EMIVCS as follows:

Theorem 4 *For a secret image* S_I *which consists of* n_w *white pixels and* n_b *black pixels, the bandwidth, denoted by* W*, of the EMIVCS is* $W = \lfloor n_w \log_2 |C_0| + n_b \log_2 |C_1| \rfloor$*, and it is achieved when using the* q_a-pixel encryption model *where* $q_a = n_w + n_b$*.*

Proof: For the *q_a-pixel encryption model* where $q_a = n_w + n_b$, which implies encrypt all the secret pixels in the secret image at a time. And it is clear that the amount of covert data carried by such EMIVCS is $W = \lfloor n_w \log_2 |C_0| + n_b \log_2 |C_1| \rfloor$. We only need to prove that W reaches its maximum when using the *q_a-pixel encryption model*, i.e. if one divides all the pixels in the secret image into several parts, and encrypts these parts separately, the amount of covert data carried is less than the *q_a-pixel encryption model*.

Without loss of generality, let $q_a = q_1 + q_2$ (i.e. divide into two parts), and suppose the encryption of the secret image S_I is realized by using *q_1-pixel encryption model* and *q_2-pixel encryption model*, and let $q_1 = a_0 + a_1$, $q_2 = b_0 + b_1$, where a_0, b_0 are the number of white pixels and a_1, b_1 are the number of black pixels. We have that the total number of pseudo-random bits can be determined is $\lfloor a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rfloor + \lfloor b_0 \log_2 |C_0| + b_1 \log_2 |C_1| \rfloor$, which is no larger than $\lfloor (a_0 + b_0) \log_2 |C_0| + (a_1 + b_1) \log_2 |C_1| \rfloor = \lfloor n_w \log_2 |C_0| + n_b \log_2 |C_1| \rfloor$. Hence, the theorem follows. □

3.4 On decoding the ciphertext

For EMIVCS, in order to encrypt the secret pixels and decode the ciphertext one needs to set a bijection between the set of pseudo-random numbers (ciphertext) and the set of share matrices. A simple way to realize that is to generate a table which contains all the share matrices and their corresponding numbers. When the dealer generates the shares, he needs to generate a pseudo-random number and find the corresponding share matrix by table-lookup, then he can encrypt the shares by using the share matrix. And when decoding the ciphertext, the participants get the share matrices according to Theorem 1, and find the corresponding numbers by table-lookup, hence, they get the ciphertext. The disadvantage of this decoding method is that, the table requires to store all the share matrices in sets C_0 and C_1, and hence has large memory requirements. In this subsection, we propose a decoding method which is more efficient than the above method.

The proposed decoding method contains two subroutines, the first is $MTN(S)$, which takes a share matrix in C_i $(i = 0, 1)$ as its input and outputs a number between 1 and $m!$. And the second is $NTM(N)$, which takes a number between 1 and $m!$ as its input and outputs a share matrix S. The subroutines $MTN(S)$ and $NTM(N)$ form a bijection between the set of the share matrices and the set of numbers between 1 and $m!$.

By using $MTN(S)$ and $NTM(N)$, when the dealer encrypts a secret pixel p, he first generates a pseudo-random number between 1 and $m!$, and then consults to the subroutine $NTM(N)$ to generate a share matrix in C_i $(i = 0, 1)$, and encrypts the secret pixel p by

using the share matrix. And when the participants decode the ciphertext, they first generate the share matrix according to the Theorem 1, and consult to the subroutine $MTN(S)$ to get the ciphertext.

Denote the columns of the basis matrix as $c_1, \cdots, c_m$, first we consider the case that $c_1, \cdots, c_m$ are pairwise different. In this part, we treat a matrix as a set of columns. The subroutine $MTN(S)$ which outputs a number between 1 and $m!$ given a share matrix S as its input is as follows:

Subroutine: MTN(S)

For $i = 1$ to $m - 1$

Find c_i in S, assume that c_i is the J_i-th column of S

Delete c_i from S

Output $N = 1 + \sum_{i=1}^{m-1}((m-i)!)(J_i - 1)$

The subroutine $NTM(N)$ which outputs a share matrix S given a number between 1 and $m!$ as its input is as follows:

Subroutine: NTM(N)

Initial S as an empty matrix

$N_0 \leftarrow N - 1$

For $i = 1$ to $m - 1$

$J_i \leftarrow \lfloor \frac{N_{i-1}}{(m-i)!} \rfloor + 1$

$N_i \leftarrow N_{i-1} - (J_i - 1)((m-i)!)$

Insert c_m to S as its 1-st column

For $i = m - 1$ to 1

Insert column c_i into S as its J_i-th column

Output S

According to the subroutines $MTN(S)$ and $NTM(N)$ above, we have the following theorem:

Theorem 5 *The subroutines $MTN(S)$ and $NTM(N)$ form a bijection between the set of share matrices in C_i $(i = 0, 1)$ and the set of numbers between 1 and $m!$.*

Proof: Because in subroutines $MTN(S)$ and $NTM(N)$, we represent the share matrices by the positions of its columns $(J_1, J_2, \cdots, J_{m-1})$ where $1 \leq J_i \leq m+1-i$ for $i = 1, 2, \cdots, m-1$, we only need to prove that $MTN(S)$ and $NTM(N)$ form a bijection between the sets $X = \{(J_1, J_2, \cdots, J_{m-1}) | 1 \leq J_i \leq m+1-i \ for \ i = 1, 2, \cdots, m-1\}$ and $Y = \{1, 2, \cdots, m!\}$. Denote $f : X \to Y$ as a map from X to Y, we prove that f is a bijection.

First, given a number in Y, according to $NTM(N)$, there exists a $(J_1, J_2 \cdots, J_{m-1})$, hence f is a surjection.

Second, for any two different elements in X, $J = (J_1, J_2, \cdots, J_{m-1})$ and $J' = (J'_1, J'_2, \cdots, J'_{m-1})$ such that $J \neq J'$, we prove that their corresponding numbers $f(J)$ and $f(J')$ are different.

According to $MTN(S)$, we have $f(J) = 1 + \sum_{i=1}^{m-1}((m-i)!)(J_i - 1)$ and $f(J') = 1 + \sum_{i=1}^{m-1}((m-i)!)(J'_i - 1)$. Denote i^* as the smallest number that $J_{i^*} \neq J'_{i^*}$, without loss of generality, we suppose $J_{i^*} > J'_{i^*}$, i.e. $J_{i^*} - J'_{i^*} \geq 1$. We have:

$$\begin{aligned} f(J) - f(J') &= \sum_{i=1}^{m-1}((m-i)!)(J_i - J'_i) \\ &= (m-i^*)!(J_{i^*} - J'_{i^*}) + \sum_{i=i^*+1}^{m-1}((m-i)!)(J_i - J'_i) \\ &\geq (m-i^*)! + \sum_{i=i^*+1}^{m-1}((m-i)!)(J_i - J'_i) \end{aligned}$$

Because $1 \leq J_i, J'_i \leq m+1-i$, we have $-(m-i) \leq J_i - J'_i \leq m-i$, hence

$$\begin{aligned} f(J) - f(J') &\geq (m-i^*)! - \sum_{i=i^*+1}^{m-1}((m-i)!)(m-i) \\ &= (m-i^*)! - ((m-i^*)! - 1) \\ &= 1 \end{aligned}$$

Hence, $f(J) - f(J') \neq 0$, we have f is an injection. Hence, f is a bijection and the theorem follows. □

For the case that there are identical columns in the basis matrix, which means that there are identical share matrices in the $m!$ permutations of the basis matrix. And suppose there are e different columns in the basis matrix, and the multiplicities of these columns are $a_1, a_2, \cdots, a_e$. Denote N_d as the number of the different share matrices in C_i, then we have $N_d = \frac{(\sum_{i=1}^{e} a_i)!}{\prod_{i=1}^{e} a_i!}$, for $i \in \{0, 1\}$. And each share matrix appears $\frac{m!}{N_d}$ times in the $m!$ permutations.

Furthermore, according to the subroutine $MTN(S)$, each permutation corresponds to a number between 1 and $m!$, we can divide these $m!$ numbers into N_d groups, where each group contains $\frac{m!}{N_d}$ numbers, and the numbers in one group correspond to an identical share matrix. We hence can form an array of length N_d by choosing the smallest number of each group. Denote this array as A, and denote $S_1^i, S_2^i \cdots, S_{N_d}^i$ as all the different share matrices in the set C_i, the following subroutine generates A:

Subroutine: MC

Initial an empty array A

For $j = 1$ to N_d

  For $q = 1$ to m

    Find the first c_q in S_j^i from left to right, assume that c_q is the J_q-th column

    of S_j^i

Delete c_q from S_j^i

$A[j] \leftarrow 1 + \sum_{q=1}^{m-1}((m-q)!)(J_q - 1)$

To differentiate the two cases that there exist and do not exist identical columns, we denote MTN-d(S) and NTM-d(N) as the corresponding subroutines for the case that there exist identical columns:

Subroutine: MTN-d(S)

$A \leftarrow MC$

For $q = 1$ to m

Find the first c_q in S_j^i from left to the right, assume that c_q is the J_q-th column of S_j^i

Delete c_q from S_j^i

$N' \leftarrow 1 + \sum_{q=1}^{m-1}((m-q)!)(J_q - 1)$

For $r = 1$ to N_d

if $A[r] = N'$

Output r

Subroutine: NTM-d(N)

$A \leftarrow MC$

$N' \leftarrow A[N]$

$S \leftarrow NTM(N')$

Output S

According to Theorem 5, we have that, each group only has one smallest number. Hence the array A is a bijection from the set $\{1, 2, \cdots, N_d\}$ and the set of smallest numbers in each group. Furthermore, because each group corresponds to a different share matrix we have that the MTN-d(S) and NTM-d(N) form a bijection between the set $\{1, 2, \cdots, N_d\}$ and the set of share matrices $\{S_1^i, S_2^i \cdots, S_{N_d}^i\}$. We summarize this result as the following theorem:

Theorem 6 *The subroutines MTN-d(S) and NTM-d(N) form a bijection between the set of share matrices in C_i ($i = 0, 1$) and the set of numbers between 1 and N_d.* □

The above subroutines are more efficient than the simple table-lookup method. Particularly, for the case that the columns $c_1, c_2, \cdots, c_m$ are pairwise different, the subroutines $MTN(S)$ and $NTM(N)$ are efficient, because they only needs fixed memory requirements. And for the case that there are identical columns in $c_1, c_2, \cdots, c_m$, the memory requirement of the subroutines MTN-d(S) and NTM-d(N) relates to the value of m. And because they only need to store the indexes of the share matrices $A[1], A[2], \cdots, A[N_d]$, they are more efficient than the simple table-lookup method. Furthermore, the table (the array A in Subroutine MC) can be previously generated and reusable.

4 Comparison of EMIVCS and TiOISSS

From the point of view of carrying the secret information, both the EMIVCS and the TiOISSS are computer aided and carry two types of secrets, one is a secret image which can be revealed by stacking the shares, and the other is covert data which is revealed by computation. The two TiOISSSs [14, 15] can be also treated as (k, k, n)-TiOISSS, which means a vague secret image is revealed by stacking any k out of n shares, and further a much finer gray-scale secret image (i.e. the covert data) is revealed by computation with these k shares.

Considering the information carrying capability, we compare the amount of covert data carried by the EMIVCS and two TiOISSSs [14, 15].

First, the covert data carried by the EMIVCS is larger than that in Lin et al.'s TiOISSS [14]. The bandwidth of the proposed EMIVCS has been discussed in Theorem 4, and can be evaluated from Table 1 and Table 2. Lin et al.'s TiOISSS [14] divides the share matrices into different types to carry covert data according to the first row. Let m be the pixel expansion of the basis matrix, which each row contains b '1' and w '0' and m=b+w. There are $\begin{pmatrix} m \\ w \end{pmatrix}$ different types of share matrix, and each secret pixel in VCS can carry $\log_2 \begin{pmatrix} m \\ w \end{pmatrix}$ bits. We list the number of share matrices generated by Droste [9] with different types in Table 3. Note that in order to satisfy the security, we can only choose the type of one row and the remaining $(n-1)$ rows are then determined according to the type of share matrix. Therefore, only $1/n$ part of each share can be used to carry $\log_2 \begin{pmatrix} m \\ w \end{pmatrix}$ bits. Since the covert data of each share is taken from a shadow image generated by polynomial-based secret sharing scheme, the total secret information carried by VCS is $k|S_I| \log_2 \begin{pmatrix} m \\ w \end{pmatrix} /n$ bits, where S_I is the binary secret image of VCS.

Second, there is no fix relationship between the amount of covert data carried by the EMIVCS and that of Yang et al.'s TiOISSS [15]. The TiOISSS [15] replaces the black pixels in the shares with gray pixels generated by polynomial-based secret sharing scheme. Therefore, each row of share matrix carries $8b$ bits, and the total amount of covert data is $8kb|S_I|$ bits, where b is the number of '1' in each row of share matrix. In most cases, especially when n is a small number, the covert data carried by Yang et al.'s TiOISSS [15] is more than that in the EMIVCS. However, in some cases, the EMIVCS can carry more data. For example, for a (2,10)-VCS constructed by Droste [9], we have, $\log_2 |C_0| = 3.32$ and $\log_2 |C_1| = 21.79$, and $b = 1, k = 2$. In the EMIVCS, each white secret pixel carries 3.32 bits and each black secret pixel carries 21.79 bits, while in Yang et al.'s TiOISSS [15], sharing one secret pixel carries $8kb = 16$ bits. With proper proportion of the numbers of

k \ n	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10
3		$\frac{4!}{2!2!}$	$\frac{6!}{3!3!}$	$\frac{8!}{4!4!}$	$\frac{10!}{5!5!}$	$\frac{12!}{6!6!}$	$\frac{14!}{7!7!}$	$\frac{16!}{8!8!}$	$\frac{18!}{9!9!}$
4			$\frac{8!}{4!4!}$	$\frac{15!}{9!6!}$	$\frac{24!}{16!8!}$	$\frac{35!}{25!10!}$	$\frac{48!}{36!12!}$	$\frac{63!}{49!14!}$	$\frac{80!}{64!16!}$
5				$\frac{16!}{8!8!}$	$\frac{30!}{15!15!}$	$\frac{48!}{24!24!}$	$\frac{70!}{35!35!}$	$\frac{96!}{48!48!}$	$\frac{126!}{63!63!}$
6					$\frac{32!}{16!16!}$	$\frac{70!}{40!30!}$	$\frac{128!}{80!48!}$	$\frac{210!}{140!70!}$	$\frac{320!}{224!96!}$
7						$\frac{64!}{32!32!}$	$\frac{140!}{70!70!}$	$\frac{256!}{128!128!}$	$\frac{420!}{210!210!}$
8							$\frac{128!}{64!64!}$	$\frac{315!}{175!140!}$	$\frac{640!}{384!256!}$
9								$\frac{256!}{128!128!}$	$\frac{630!}{315!315!}$
10									$\frac{512!}{256!256!}$

Table 3: The number of share matrices with different types in Lin et al.'s TiOISSS

white secret pixels to black secret pixels, the EMIVCS can carry more covert data than Yang et al.'s TiOISSS [15].

From the point of view of the visual quality, both the EMIVCS and the TiOISSS can visually recover the secret image by stacking shares. The EMIVCS and Lin et al.'s TiOISSS [14] use traditional VCS as the building block, hence the recovered secret image is the same as that of the traditional VCS. In Yang et al.'s TiOISSS [15], the black pixels in shares are replaced by gray pixels, and the contrast of VCS is diminished. Therefore, the visual quality of the recovered secret image by stacking shares is deteriorated in Yang et al.'s TiOISSS [15], which is a disadvantage of their scheme.

Besides, another disadvantage of Yang et al.'s TiOISSS [15] is that, to reconstruct the covert data the participants have to obtain the greyness of each subpixel precisely, which is impractical if the shares are printed on the transparencies. Occasional scrub may change the greyness of the subpixels in the transparencies, which will be impossible to reconstruct the covert data.

Both the EMIVCS and Lin et al.'s TiOISSS [14] carry covert data by choosing different share matrices, hence there is a bijection between the set of pseudo-random numbers (ciphertext) and the set of share matrices. In Lin et al.'s TiOISSS [14], it employs a lookup table to map the different types of share matrices to the set of pseudo-random numbers. The disadvantage of their scheme is that, the table needs to store all the types of share matrices, which has large memory requirements. However, in the EMIVCS, an efficient algorithm is introduced to make the mapping more convenient and memory efficient.

5 Conclusion

In this paper, we propose a construction of the (k, t, n)-EMIVCS, which can carry additional covert data compared with the traditional (k, n)-VCS by treating the pseudo-random inputs as a subchannel. And we analyze some related problems about EMIVCS such as the pseudo-randomness that the input of VCS requires, and the sufficient conditions to uniquely determine a share matrix in the set C_i for $i = 0, 1$, and the bandwidth of the proposed EMIVCS, and we also propose an efficient decoding method to decode EMIVCS. At last, comparisons with some computer aided VCS's are given such as the TiOISSS in [14, 15].

The proposed (k, t, n)-EMIVCS is especially useful for the $(n-1, n-1, n)$-EMIVCS and (n, n, n)-EMIVCS, because in such cases, the qualified participants can get the secret image and the covert data at the same time. The constructions of $(k, n-1, n)$-EMIVCS and (k, n, n)-EMIVCS can be easily implemented by the proposed scheme. For general value of $k < t < n-1$, we leave it as an open problem for further research.

6 Acknowledgement

This work was supported by NSFC grant No. 60903210. The paper was first submitted in 2007, and has been reviewed for several times. During the reviewing procedure, many anonymous reviewers provided many valuable constructive comments. We thank a lot to these anonymous reviewers.

References

[1] M. Naor and A. Shamir. Visual cryptography. In *EUROCRYPT '94, Springer-Verlag Berlin*, volume LNCS 950, pages 1–12, 1995.

[2] B. Surekha, G. Swamy, and K.S. Rao. A multiple watermarking technique for images based on visual cryptography. In *Computer Applications*, volume 1, pages 77–81, 2010.

[3] T. Monoth and B. Anto P. Tamperproof transmission of fingerprints using visual cryptography schemes. In *Procedia Computer Science*, volume 2, pages 143–148, 2010.

[4] J. Weir and W. Yan. Resolution variant visual cryptography for street view of google maps. In *Proceedings of the ISCAS*, pages 1695–1698, 2010.

[5] C.N. Yang, T.S. Chen, and M.H. Ching. Embed additional private information into two-dimensional bar codes by the visual secret sharing scheme. In *Integrated Computer-Aided Engineering*, volume 13, Number 2, pages 189–199, 2006.

[6] S. Cimato and C.N. Yang. *Visual cryptography and secret image sharing*. CRC Press, Taylor & Francis, 2011.

[7] F. Liu, C.K. Wu, and X.J. Lin. The alignment problem of visual cryptography schemes. In *Designs, Codes and Cryptography*, volume 50, pages 215–227, 2009.

[8] M. Iwamoto and H. Yamamoto. A construction method of visual secret sharing schemes for plural secret images. In *IEICE Transactions on Fundamentals*, volume E86-A.NO.10, pages 2577–2588, 2003.

[9] S. Droste. New results on visual cryptography. In *CRYPTO '96, Springer-Verlag Berlin LNCS*, volume 1109, pages 401–415, 1996.

[10] M. Iwamoto, W. Lei, K. Yoneyama, N. Kunihiro, and K. Ohta. Visual secret sharing schemes for multiple secret images allowing the rotation of shares. In *IEICE Transactions on Fundamentals*, volume E89-A. NO.5, pages 1382–1395, 2006.

[11] C.N. Yang and C.S. Laih. New colored visual secret sharing schemes. In *Designs, Codes and Cryptography*, volume 20, pages 325–335, 2000.

[12] S.J. Shyu. Efficient visual secret sharing scheme for color images. In *Pattern Recognition*, volume 39, pages 866–880, 2006.

[13] A. Shamir. How to share a secret. In *Communications of the ACM*, volume 22 (11), pages 612–613, 1979.

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

[15] C.N. Yang and C.B. Ciou. Image secret sharing method with two-decoding-options: Lossless recovery and previewing capability. In *Image and Vision Computing*, volume 28, pages 1600–1610, 2010.

[16] W.P. Fang and J.C. Lin. Visual cryptography with extra ability of hiding confidential data. In *Journal of Electronic Imaging*, volume 15(2), page 023020, 2006.

[17] G.B. Horng, T.H. Chen, and D.S. Tsai. Cheating in visual cryptography. In *Designs, Codes and Cryptography*, volume 38, pages 219–236, 2006.

[18] Patent with US Application No.: 5243655. System for encoding and decoding data in machine readable graphic form. 1993.

[19] C. Blundo, A. De Santis, and D.R. Stinson. On the contrast in visual cryptography schemes. In *Journal of Cryptology*, volume 12(4), pages 261–289, 1999.

[20] H. Koga. A general formula of the (t,n)-threshold visual secret sharing scheme. In *ASIACRYPT '2002, Springer-Verlag LNCS*, volume 2501, pages 328–345, 2002.

[21] W.F. Ehrsam, C.H.W. Meyer, J.L. Smith, and W.L. Tuchman. *Message verification and transmission error detection by block chaining.* 1976.

[22] J. Soto and L. Bassham. Randomness testing of the advanced encryption standard finalist candidates. In *Proceedings AES3, New York, http://csrc.nist.gov/publications/nistir/ir6483.pdf*, 2001.

[23] C. Blundo, A. De Bonis, and A. De Santis. Improved schemes for visual cryptography. In *Designs, Codes and Cryptography*, volume 24, pages 255–278, 2001.