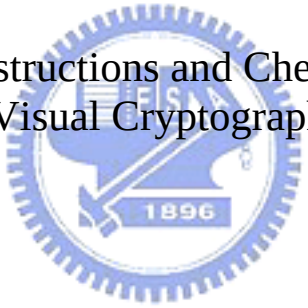
國立交通大學

資訊工程學系

博 士 論 文

視覺密碼學更具效率的設計方法及偽造防範

On Efficient Constructions and Cheating Prevention of Visual Cryptography



研 究 生：胡智明

指導教授：曾文貴　教授

中 華 民 國 九 十 六 年 元 月

視覺密碼學更具效率的設計方法及偽造防範
On Efficient Constructions and Cheating Prevention of Visual Cryptography

研 究 生：胡智明　　Student：Chih-Ming Hu

指導教授：曾文貴 博士　　Advisor：Wen-Guey Tzeng

國立交通大學資訊學院

資訊工程學系

博 士 論 文

A Dissertation Is

Submitted to Department of Computer Science

National Chiao Tung University

for the Degree of

Doctor of Philosophy

in

Computer and Science

Ianuary 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年元月

摘要

視覺密碼學(Visual Cryptography)是一種將秘密的影像加密成數張分享片(Shares)的方法。如此一來只要疊足夠數量的分享片，便能解開那秘密影像。分享片通常以投影片做成，每位參予者(Participant)擁有一片投影片。之前大部分的研究，主要集中在增進兩種參數，像素擴展(Pixel Expansion)及對比(Contrast)。

傳統視覺密碼學定義，要求解開的秘密影像必須比背景黑。然而我們觀察這並不是必要條件，特別是應用在文字影像時，因此我們提出一個較佳的定義。根據這個新定義，我們發現許多傳統視覺密碼方法及其應用，都是可以精進的。根據我們的研究，對傳統視覺密碼學，我們做了以下的貢獻：

1. 我們利用這新定義，研究新視覺密碼學的特質及其界限(Bound)，並提出數個比傳統密碼學較佳的方法。
2. 我們展示三種欺騙的方法，並應用它們來攻擊視覺密碼學。也提出了一種很有效率的方法，將所有視覺密碼學方法轉換成具防欺騙的功能。
3. 根據這新定義，我們提出了一個新方法來完成反轉視覺密碼方法(Visual Cryptography Scheme with Reversing)。跟之前的方法比較，我們的方法僅要求每一位使用者儲存兩張投影片。
4. 我們提出一種新的 2 out of n 的延伸視覺密碼方法(Extended Visual Cryptography Scheme)，雖然這個方法的影像，沒有那麼漂亮，但比起之前的方法則有較佳的對比。

關鍵字：視覺密碼學， 視覺秘密分享法， 存取結構， 偽冒防制， 反轉，最佳對比， 像素擴展。

Abstract

Visual Cryptography (VC) is a method of encrypting a secret image into shares such that stacking a sufficient number of shares reveals the secret image. Shares are usually presented in transparencies. Each participant holds a transparency. Most of the previous research work on VC focuses on improving two parameters: pixel expansion and contrast.

The conventional definition requires that the revealed secret images are always darker than the backgrounds. We observed that this is not necessary, in particular, for the textual images. Therefore, we proposed an improved definition for visual cryptography based on our observation, in which the revealed images may be darker or lighter than the backgrounds. Based on the new definition, we find that many extensions of the original Visual Cryptography Schemes (VCSs) are improvable. According to our study, we improve the results of the original VCSs including the following contributions:

- We studied properties and obtained bounds for visual cryptography schemes based on the new definition. We proposed methods to construct visual cryptography schemes based on the new definition.
- We presented three cheating methods and applied them on attacking existent VC or Extended VC (EVC) schemes. We improved one cheat-preventing scheme. We proposed a generic method that converts a VCS to another VCS that has the property of cheating prevention.
- Based on the new definition, we propose a new ideal VCS with reversing which is compatible and requires fewer stacking and reversing operations, compared to all previous schemes. Each participant is required to store only two transparencies .
- We propose a $(2,n)$-EVCS scheme based on the new definition. Although the image of this construction is not "smooth", it has better contrast than previous results.

誌謝

毫無疑問的，首先我最感謝的便是我的指導教授：曾文貴老師。自民國八十三年，受教於他修讀碩士至今博士畢業已十二年整。一般人或許會認爲時間太長，但我個人認爲依本人資質，若非此十二年的訓練，也無法做一位夠資格的交大畢業生。從一位不具厚實資訊知識的職業軍人轉變爲一位博士，多虧了曾教授對我的指導。他不僅對我循循善誘，教導我寫作論文及從事研究之技能及態度。最重要的從他身上，更體會了做學問所必須具備之嚴謹態度。所謂一日爲師，終身爲師。這正是我現在所體驗的，老師也將成爲我一輩子學習的對象。

在口試時，口試委員(賴溪松、陳玲慧、蔡錫鈞、洪國寶、雷欽隆、顏嵩銘等教授)，給我諸多建設性之意見，也感謝口試委員指導，讓我論文更臻於完善。

再來，最感謝的就是母親多年來的教誨。母親雖僅小學畢業，然而她做事的堅毅、執著，深深感染我，讓我能完成學業。還有父親的惕勵，背後默默的支持，我想這學位的拿到，也是我對他們的多年來辛勞的一種回饋，也祝福他們身體健康。感謝兩位弟弟的支持、包容，也祝他們家庭美滿。

最後我還要感謝朱成康，他在我這一段期間給我許多幫忙，也希望他早日畢業。還有資訊安全實驗室的成員翁御舜、志嘉、阿田等給我許多研究上的意見討論。最後，我也要感謝砲校的長官及資訊中心的同仁，因爲他們對我的支持，才得以讓我完成學業。雖然我將離開軍中，但我受國家及軍中的栽培實在太多了，我永遠以曾爲一職業軍人爲榮，也希望對我的國家能有些貢獻。

謝謝大家

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Following the remarkable advance of computer technology, the theory and applications of computer security are also making progress at a tremendous pace. Powerful cryptographic algorithms and protocols are designed to meet security requirements of various applications. However, using a computer to decrypt secrets is infeasible in some situations. For example, a security guard checks the badge of an employee or a secret agent recovers an urgent secret at some place where no electronic devices are available. In these situations the human visual system is one of the most convenient and reliable tools to do checking and secret recovery. Therefore, Naor and Shamir [20] invented the Visual Cryptography (VC) in which a secret image (printed text, picture, etc) is encrypted in a perfectly secure way such that the secret can be decoded directly by the human visual system.

VC is a method of encrypting a *secret image* into *shares* such that stacking a sufficient number of shares reveals the secret image. Shares are usually presented in transparencies. Each participant holds a transparency (share). Unlike conventional cryptographic methods, VC needs no complicated computation for recovering the secret image. The act of decryption is to stack shares and view the image that appears on the stacked shares simply. A (k, n)-Visual Cryptography Scheme (denoted as (k, n)-VCS) is a visual *secret sharing scheme* [23, 24] such that stacking any k or more shares reveals the secret image, but stacking fewer than k shares reveals not any information about the secret image.

A VCS would be helpful if the shares are meaningful or identifiable to every participant. A VCS with this extended characteristic is called Extended VCS (EVCS) [2, 20]. A (k, n)-EVCS is like a (k, n)-VCS except that each share displays a meaningful image, which will be called *share image* hereafter. In order to identify the transparencies (shares), some images or symbols are needed to appear

on the transparencies. Different shares may have different share images.

A VCS is called *perfect black* (*white* resp.) if all the subpixels associated to a black (white resp.) pixel is black (white resp.). An image with optimal contrast is called *ideal contrast*. That is all the subpixels associated to a black and white pixels are perfect reconstructed. Let h (l resp.) be the number of white subpixels in a white (black resp.) pixel. Then, an image is of ideal contrast if $h = m$ and $l = 0$. A VCS is perfect black if the value l of the reconstructed image is 0. For the characteristic of contrast, the equation $m \geq h > l \geq 0$ must be satisfied if one should identify the secret image. The most concerned issue for the reconstructed image is contrast [20]. Since the share held by each participant should consist of same number of white subpixels and black subpixels (for the reason of computationally secure), it is impossible to recover a secret image with ideal contrast in VC. Therefore, Viet and Kurosawa [27] proposed a VCS, called VCS with Reversing (VCSR), which adopted a simple tool (copy machine) to improve the contrast of the reconstructed image. For most copy machines nowadays, to reverse black and white pixels in a paper is already a fundamental function.

There are quite many new results and extensions of the original work [1, 2, 4, 5, 6, 7, 9, 11, 14, 19, 21]. We briefly describe them as follows.

1.1 Previous Works

Naor and Shamir [20] defined visual cryptography formally and proposed an optimal visual cryptography scheme for the (n, n)-threshold access structure. They also extended the work for the (k, n)-threshold access structures. Many improvements and extensions follows [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 16, 17, 25, 26, 27, 29]. For example, Ateniese, et al. [1] proposed an elegant VCS for general access structures based on the cumulative array method. They analyzed structures of visual cryptography schemes and proved bounds for the size of the shares. Hofmeister, et al. [11] proposed a visual cryptography scheme for (k, n)-threshold access structures, which achieves the best contrast by solving a simple linear program. Visual cryptography schemes for color images were given in [18, 22].

Extended visual cryptography defines that each share shows an image, but their combinations show the real secret image. Naor and Shamir [20] proposed an extended visual cryptography scheme for the $(2, 2)$-threshold access structure. Droste [9] proposed a very general method to construct an extended visual cryptography scheme for an arbitrary access structure, which is not necessarily monotonic.

Ateniese, et al. [2] proposed a hyper-coloring technique to construct extended visual cryptography schemes. It is possible that each share shows a different image initially and a different combination of shares shows a different secret image. Kim, et al. [15] discussed negative images for access structures.

Viet and Kurosawa [27] proposed a VCS with *reversing*, with which the reconstructed secret image obtains almost ideal contrast. They adopted a tool (copy machine) to improve the contrast of the reconstructed image. Before long, S. Cimato et al. [8] proposed two elegant schemes to construct VCSs with reversing. In their first scheme, each participant stores m transparencies, where m is the pixel expansion (the number of subpixels in each pixel). They proposed another VCS, using as a building block a *binary secret sharing scheme* (BSS). This scheme reduces the number of transparencies held by each participant to r, where r is the number of bits in the binary representation of the largest share. Yang et al. [31] applied a cyclic shift operation of subpixels to the Viet and Kurosawa's scheme and obtain a new efficient VCSR.

Naor and Pinkas [19] showed some methods of authentication and identification for VC. Yang and Laih [30] proposed two cheat-preventing methods. Their first method needs an on-line TA (Trusted Authority) to verify the shares of participants. Horng also et al. [12] proposed a cheating method against some VC schemes.

1.2 Motivations

For the interesting characteristic, Visual Cryptography is a quite improvable topic to study. Our improvements on Visual Cryptography include proposing a new definition, studying the cheating behaviors on VC, and doing some improvements on EVCS and VCS with Reversing. We describe the motivations of these works as follows.

The previous work we mentioned above, use the definition of Naor and Shamir, i.e., when recovered, the secret image is darker than the background. However, in many situations, what the human visual system cares about is "contrast", no matter whether the image is darker or lighter than the background. For example, we can get the textual secret image "5" from either ⑤ or ❺. Therefore, we give a new definition for visual cryptography based on the above observation.

VC has been studied intensively since the pioneer work [20] of Noar and Shamir [5, 6, 8, 10, 11, 17, 26]. In these cases, all participants who hold shares are assumed to be semi-honest, that is, they won't present *false* or *fake shares* during the phase

of recovering the secret image. Thus, the image shown on the stacking of shares is considered as the *real secret image*. Nevertheless, cryptography is supposed to guarantee security even under the attack of malicious adversaries who may deviate from the scheme in any way. We have seen that it is possible to cheat [12, 19, 28, 30] in VC, though it seems hard to imagine. For cheating, a cheater presents some fake shares such that the stacking of fake and genuine shares together reveals a fake image. With the property of unconditional security, VC is suitable for sending to highly-classified orders to a secret agent when computing devices may not be available. The secret agent carried some shares, each with a pre-determined order, when departing to the hostile country. When the headquarter decides to execute a specific order, it can simply send another share to the agent so that the agent can recover what the order is. We can see that it would be terrible if the dispatched share cannot be verified due to a cheater's attack.

At first glance, it seems very difficult to cheat in EVCS because the cheater does not know the share images that appear on the genuine shares and, thus, has no information about the distributions of black and white pixels of the share images. This information is crucial for cheating in VC. However, we show that it is still possible to cheat in EVC.

A VCS with *reversing* (VCSR) [27] is a VCS where every participant is allowed to change black pixels on the transparency into white pixels and vice-versa. A practical material for constructing VC is the transparency. However, due to the contiguous black and white pixels on each transparency, the reconstructed secret image will become much more ambiguous after every stacking if the transparencies are not superimposed precisely. As a result, reducing the stacking and reversing operations is important for VCSs with reversing. Therefore, we propose a compatible ideal contrast VCSR with only two *runs*. In other words, each participant only need to use two shares in the reconstruction phase.

Extended Visual Cryptography [2, 20] stipulates that each share shows an image, and their combinations show the real secret image. Based on the new definition, we find that the pixel expansion of a $(2,n)$-EVCS can be reduced to a smaller number than that of a $(2,n)$-EVCS based on the original definition.

1.3 Our Contributions

1.3.1 A More General and Efficient Definition

With the more general definition, we propose various visual cryptographic schemes. Our schemes have better pixel expansion than previous results in some cases. In Chapter 3, we obtain the following results:

- We propose an improved definition for visual cryptography.
- We study properties and obtain bounds for visual cryptography schemes based on the new definition.
- We propose methods to construct visual cryptography schemes based on the new definition. The experiment results show that our constructions provide better pixel expansion in average.

1.3.2 Cheating Behaviors and Prevention

In Chapter 4, we study the cheating problem in VC and EVC. We present three cheating methods and apply them on existent VC or EVC schemes. Our attacks are to reveal fake images to cheat honest participants.

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

- We propose three cheating methods against VC or EVC schemes. The first two methods are applied to attack VC schemes and the third one is applied to attack EVC schemes. These three methods are easy to implement and satisfy the cheating definition for cheating traditional secret sharing schemes.
- We review some previously proposed cheat-preventing VC or EVC schemes and demonstrate that those schemes are either not robust enough (still cheatable) or improvable.
- We propose some necessary criteria for a VCS to be secure against cheating robustly. By these criteria, we propose a generic method that converts any VCS to another VCS with the property of cheating prevention. Our conversion is very efficient and incurs little overhead compared with the original VCS. The degression in contrast of the converted VCS is almost optimal. For each pixel

of the secret image, we add two additional subpixels to the encoded subpixels only, no matter how many the encoded subpixels are.

1.3.3 More Efficient Compatible VCSs with Reversing Based on the New Definition

In Chapter 5, we show how to

- construct three ideal contrast VCSs with fewer reversing and stacking operations while maintaining compatibility.
- reduce the number of transparencies held by each participant to two. It is an improvement on all properties when compared to the schemes of S. Cimato et al [8], except for the property of pixel expansion.

1.3.4 EVCS Based on the New Definition

With the new definition, we propose a new (2, n)-EVCS. Our schemes have much better contrast than previous results in some cases. In Chapter 6, we show our contributions including:

- an improved definition for extended visual cryptography.
- a new (2, n)-EVCS scheme that has better contrast than the scheme based on the new definition.

Chapter 2

Preliminaries

2.1 Model and Notation

Access structure. We consider arbitrary access structures. Let $P = \{1, 2, \ldots, n\}$ be a set of participants. $\Gamma = (P, Q, F)$ is an access structure if both Q and F are subsets of 2^P and $Q \cap F = \emptyset$. Each $X \in Q$ is a qualified set of participants and each $Y \in F$ is a forbidden (non-qualified) set of participants. We call (P, Q, F) *complete* if $F = 2^P - Q$, which is denoted by (P, Q) in short. (P, Q) is a (k, n)-*threshold* access structure if all k- or more-element subsets of P are in Q. Q is monotonically increasing if $X \in Q$ implies that for all $X' \supseteq X$, $X' \in Q$. F is monotonically decreasing if $X \in F$ implies that for all $X' \subseteq X$, $X' \in F$. We say that $\Gamma = (P, Q, F)$ is *monotonic* if Q is monotonically increasing and F is monotonically decreasing. We remark that Q is not necessarily monotonically increasing and F is not necessarily monotonically decreasing for an arbitrary access structure (P, Q, F).

Notation. Let B be a Boolean matrix and B_i be the ith row vector of B. Let $B_i + B_j$ be the bit-wise OR of vectors B_i and B_j. Let X be a subset $\{i_1, i_2, \ldots, i_q)$ of a participant set P. We define $OR(B, X)$, $AND(B, X)$ and $XOR(B, X)$ to be the vector of "OR", "AND" and "XOR" resp. of rows $i_1, i_2, \ldots, i_q$ of B. Let GREY(GP)= |black subpixels| $/m$ be the grey level of a white (or black) pixel, where m is the *pixel expansion* of the pixel. That is, $OR(B, X) = B_{i_1} + B_{i_2} + \cdots + B_{i_q}$. Let $w(v)$ be the Hamming weight of row vector v. For brevity, we let $w(B, X) = w(OR(B, X))$. Let $A||B$ denote the concatenation of two matrices A and B of the same number of rows. Let $|X|$ be the number of elements in set X.

Bit Operations. We use "$S_i + S_j$" to denote "the stacking of shares S_i and S_j". The "stacking" corresponds to the bitwise-OR operation "+" of subpixels in shares S_i and S_j. Let S_i' denote the complement share (transparency) of S_i for participant i, in other words, we obtain S_i' by computing one reversing operation on S_i. Let

$S_i + S_j$, $S_i \times S_j$ and $S_i \oplus S_j$ be the bit-wise OR, AND, and XOR of the corresponding supixels on transparencies S_i and S_j.

It is well known that any Boolean operation can be performed solely by the combination of OR and NOT gates. Therefore, using a VCS with reversing we can denote more bit operations than in a traditional VCS. For example, an XOR operation is equal to four NOT and three OR operations, i.e. four reversing and three stacking operations.

$$S_i \oplus S_j = OR((OR(S_i', S_j))', (OR(S_i, S_j'))')$$

Probabilistic VCS. Let $p_b(S) = w(v)/m$, where v is a black pixel in share S and m is the dimension of v. Similarly, $p_w(S) = w(v)/m$, where v is a white pixel in share S. Note that all white (or black) pixels in a share have the same Hamming weight.

2.2 Visual Cryptography Scheme

In visual cryptography, a secret image consists of a collection of black and white pixels. Each pixel in the image is considered separately. A pixel is divided into pixel shares. Each pixel share consists of m subpixels and is given to a participant such that a qualified set of participants can recover the pixel by stacking their pixel shares and a set of forbidden participants cannot get any information about the pixel, that is, the subpixel patterns of the pixel shares of the black pixel are the same as those of the white pixel. An *image share* (or share) of an image consists of all the pixel shares of its pixels.

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

The resultant shares need satisfy the properties of visual cryptography. The conventional definition for VCS [1] is as follows.

Definition 2.2.1. *Let* $\Gamma = (P, Q, F)$ *be an access structure. Two collections* (multisets) C^0 *and* C^1 *of* $n \times m$ *Boolean matrices constitute a* (Γ, m)*-VCS if there exist a value* $\alpha(m) > 0$ *and a set* $\{(X, t_X)\}_{X \in Q}$ *satisfying:*

1. *Any qualified set* $X = \{i_1, i_2, \ldots, i_q\} \in Q$ *can recover the secret image by stacking their shares. Formally, for any* $M \in C^0$, $w(M, X) \leq t_X - \alpha(m) \times m$; *whereas, for any* $M' \in C^1, w(M', X) \geq t_X$.

2. *Any forbidden set* $Y = \{i_1, i_2, ..., i_q\} \in F$ *has no information on the secret image. Formally, the two collections* $C^t, t \in \{0, 1\}$, *of* $q \times m$ *matrices obtained by restricting each* $n \times m$ *matrix in* $M \in C^t$ *to rows* $i_1, i_2, ..., i_q$, *are indistinguishable in the sense that they contain the same matrices with the same frequencies.*

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

The following shows an example of VC.

Example 2.2.1. *Let* $P = \{1, 2, 3\}, Q = \{(1, 2), (2, 3), (1, 2, 3)\}$ *and then* $F = \{1, 2, 3, (1, 3), ()\}$. *The two basis matrices*

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

form a $(\Gamma, 4)$*-VCS with contrast* $\alpha(m) = 1/4$. *The shares* S_1, S_2 *and* S_3, *and the stackings of them are given in Figure 2.1.*

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

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

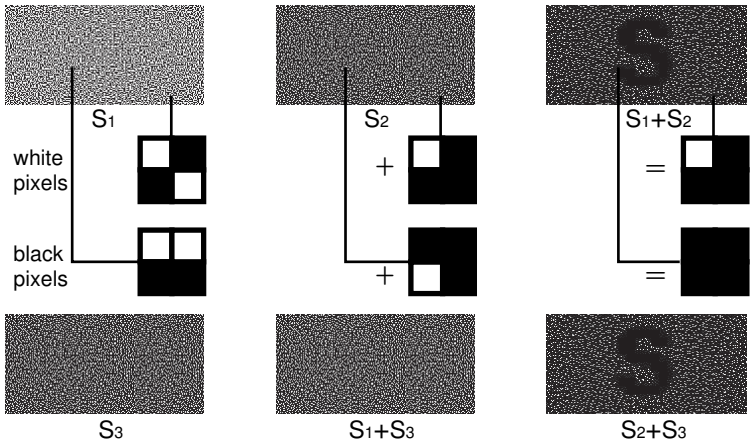


Figure 2.1: A $(\Gamma, 4)$-VCS and the structures of subpixels.

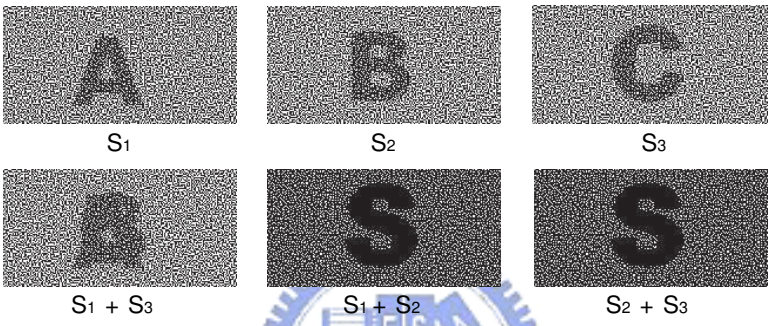


Figure 2.2: A $(\Gamma, 4)$-EVCS.

Example 2.2.2. *Figure 2.2 shows an EVCS for the access structure* Γ *of Example 2.2.1. The share images of* S_1*,* S_2 *and* S_3 *are* **A**, **B** *and* **C**, *respectively. Note that* $S_1 + S_3$ *shows no information about the secret* $\boldsymbol{S}$.

We consider general access structures. An access structure is non-monotonic if some forbidden set contains a qualified set. Non-monotonic access structures have some applications. For example, it may be that a participant x has the right to veto the decision of a qualified set X, such that $X \cup \{x\}$ is a forbidden set. We point out that the participants may not know Q and F. When some participants come together, all they do is to stack their shares and get the image revealed by their stacked shares. Therefore, non-monotonic access structures have some physical meaning.

We observe that by the definition only monotonic access structures have visual cryptography schemes. To see this, assume that a forbidden set $X \in F$ contains a qualified set $Y \in Q$. Then, X's corresponding D^0 and D^1 are distinguishable by

observing the matrices of D^0 and D^1 restricted to the rows of Y.

We can see that by Definition 2.2.1, recovered images are always darker than backgrounds. As explained above, we give a new definition for visual cryptography that stresses "contrast". That is, some recovered images are darker than backgrounds and some are lighter than backgrounds.

Definition 2.2.2. *Let $\Gamma = (P, Q, F)$ be an access structure. Two collections (multi-sets) C^0 and C^1 of $n \times m$ Boolean matrices constitute a visual cryptography scheme (Γ, m)-VCS if there exist value $\alpha(m) > 0$ and the set $\{(X, t_X)\}_{X \in Q}$ satisfying:*

1. *Any qualified set $X = \{i_1, i_2, \ldots, i_q\} \in Q$ can recover the shared image by stacking their shares. Formally, for any $M \in C^0$, $w(M, X) = t_X$; whereas, for any $M' \in C^1$, $w(M', X) \geq t_X + \alpha(m) \cdot m$ or for any $M' \in C^1$, $w(M', X) \leq t_X - \alpha(m) \cdot m$.*

2. *Any forbidden set $X = \{i_1, i_2, \ldots, i_q\} \in F$ has no information on the shared image. Formally, let $D^t, t \in \{0, 1\}$, be two collections of $q \times m$ matrices obtained by restricting each $n \times m$ matrix in $M \in C^t$ to rows $i_1, i_2, \ldots, i_q$, such that*

   (a) *If X does not contain any qualified set in Q, D^0 and D^1 are indistinguishable in the sense that they contain the same matrices with the same frequencies.*

   (b) *If X contains a qualified set in Q, the two collections $V^t, t \in \{0, 1\}$, of $1 \times m$ vectors obtained by OR-ing all rows of each $q \times m$ matrix in D^t are indistinguishable in the sense that they contain the same vectors with the same frequencies.*

Our definition changes the property of contrast, in which the revealed images may be darker or lighter than backgrounds. We fix the threshold associated to $M \in C^0$ and adjust the threshold associated to $M \in C^1$. In defining security, 2(b) deals with the case of non-monotonic access structures. We require that the "stacked shares" (the OR vector of the corresponding rows) reveal no information about the image.

We shall use VCS_1 for a VCS based on Definition 2.2.1 and VCS_2 for a VCS based on Definition 2.2.2. We give an example in Appendix to show that this definition may reduce the pixel expansion rate. We can see that the secret image "CRYPTOLOGY" is either darker or lighter than the background. The basis matrices of our VCS_2

construction have $m = 4$ and $\alpha(m) = 1/4$. However, by the previous definition, any VCS_1 for the access structure needs at least $m = 12$ and $\alpha(m) = 1/12$.

2.3 Visual Cryptography Schemes with Reversing

With the extra *reversing* operation, we slightly modify the definition for VCS [1] to meet the requirements of VCS with reversing as follows.

Definition 2.3.1. *Let $\Gamma = (P, Q, F)$ be an access structure. Two collections (multisets) C^0 and C^1 of $n \times m$ Boolean matrices constitute a (Γ, m)-VCS with reversing if there exist a value $\alpha(m) > 0$ and a set $\{(X, t_X)\}_{X \in Q}$ satisfying:*

1. *Any qualified set $X = \{i_1, i_2, ..., i_q\} \in Q$ can recover the shared image by stacking or reversing their transparencies. Formally, for any $M \in C^0$, $w(M, X) \leq t_X - \alpha(m) \times m$; whereas, for any $M' \in C^1$, $w(M', X) \geq t_X$.*

2. *Any forbidden set $X = \{i_1, i_2, ..., i_q\} \in F$ has no information on the shared image. Formally, the two collections $C^t, t \in \{0, 1\}$, of $q \times m$ matrices obtained by restricting each $n \times m$ matrix in $M \in C^t$ to rows $i_1, i_2, ..., i_q$, are indistinguishable in the sense that they contain the same matrices with the same frequencies.*

2.4 Extended Visual Cryptography Schemes

We follow in the footsteps of the work of Ateniese et al. [2]. An $(\Gamma_{Qual}, \Gamma_{Forb}, m)$-EVCS, with pixel expansion m, for an access structure $(\Gamma_{Qual}, \Gamma_{Forb})$ on a set of n participants, is similar to VCS except for every share must show some innocent looking image. The quantities α_F and α_S measure the contrast of the reconstructed image and the contrast of the shares respectively. We will refer to the color of a white (black) pixel as a w pixel (b pixel). Let $C_C^{C_1 \cdots C_n}$, where $c, c_1, ..., c_n \in \{b, w\}$, be the collection of matrices from which the dealer chooses a matrix to encode, for $i = 1, ..., n$, a c_i pixel in the image associated to participants i in order to obtain a c pixel when the shares associated to a set $X \in \Gamma_{Qual}$ are stacked together. Therefore, in order to implement an EVCS we must construct 2^n pairs of such collections $(C_w^{c_1 \cdots c_n}, C_b^{c_1 \cdots c_n})$, one for each possible combination of white and black pixels in the n original images.

The conventional definition for EVCS consists of the following properties.

Definition 2.4.1. *[2] Let* $(\Gamma_{Qual}, \Gamma_{Forb})$ *be an access structure on a set of* n *participants. A family of* 2^n *pairs of collections (multisets) of* $n \times m$ *boolean matrices* $\{(C_w^{c_1 \ldots c_n}, C_b^{c_1 \ldots c_n})\}_{c_1,\ldots,c_n \in \{b,w\}}$ *constitute a weak* $(\Gamma_{Qual}, \Gamma_{Forb}, m)$*-EVCS if there exist values* $\alpha(m)$ *and* $\{t_X\}_{X \in \Gamma_{Qual}}$ *satisfying:*

1. Any (qualified) set $X \in \Gamma_{Qua}$ can recover the shared image.
   Formally, for any $X \in \Gamma_{Qua}$ and for any $c_1, ..., c_n \in \{b, w\}$ the threshold t_X and the relative difference $\alpha(m)$ are such that for any $M \in C_w^{c_1 \ldots c_n}$ we have that $w(M_X) \leq t_X - \alpha(m) \times m$; whereas, for any $M \in C_b^{c_1 \ldots c_n}$ it results that $w(M_X) \geq t_X$.

2. Any (forbidden) set $X = \{i_1 ... i_p\} \in \Gamma_{Forb}$ has no information on the shared image. Formally, for any $c_{i_1}, ..., c_{i_p} \in \{b, w\}$ the pair of collections $\cup_{i \in \{1,\ldots,n\} \setminus X} \cup_{C_i \in \{b,w\}} D_t^{c_1,\ldots,c_n}$ with $t = \{b, w\}$, where $D_t^{c_1,\ldots,c_n}$ is obtained by restricting each $n \times m$ matrix in $C_t^{c_1,\ldots,c_n}$ to rows $i_1, ..., i_p$, are indistinguishable in the sense that they contain the same matrices with the same frequencies.

3. After the original innocent looking images are encoded they are still meaningful, that is, any user will recognize the image on his transparency.
   Formally, for any $i \in \{1, ..., n\}$ and any $c_1, ..., c_{i-1}, c_{i+1}, ..., c_n \in \{b, w\}$, it results that
   $$\mathbf{min}_{M \in M_b} w(M_i) > \mathbf{max}_{M \in M_w} w(M_i)$$
   where $M_b = \cup_{c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n \in \{b,w\}} C_w^{c_1 \ldots c_{i-1} b c_{i+1} \ldots c_n}$
   and $M_w = \cup_{c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n \in \{b,w\}} C_w^{c_1 \ldots c_{i-1} w c_{i+1} \ldots cn}$.

The first property is called *contrast*. It ensures that the image can be seen when the transparencies of a qualified set are stacked. The second property, called *security*, ensures that nothing can be recovered when stacking the transparencies of a set in Γ_{Forb}. Finally, the third property called *identification* implies that after encoding the n original innocent looking images by using the 2^n pairs of collections $(C_w^{C_1 \ldots C_n}, C_b^{C_1 \ldots C_n})$, where $c_1, ..., c_n \in \{b, w\}$, any user will recognize the image on his share.

Chapter 3

Improvements on the Original VCS

In this chapter, we studied properties and obtained bounds for visual cryptography schemes based on the new definition. We proposed methods to construct visual cryptography schemes based on the new definition. The experiments showed that visual cryptography schemes based on our definition indeed have better pixel expansion in average.

3.1 Properties of VCS_2

In this section, we study properties about VCS_2 and show how to construct a VCS_2 from smaller VCS_2.

Since VCS_2 is a generalization of VCS_1, any VCS_1 is a VCS_2.

Theorem 3.1.1. *Let $\Gamma = (P, Q, F)$ be an access structure. Any (Γ, m)-VCS_1 is a (Γ, m)-VCS_2.*

Proof. This is trivial since VCS_1 is a special case of VCS_2. □

If basis matrices S^0 and S^1 have a common column, we can delete it from S^0 and S^1 to reduce pixel expansion.

Theorem 3.1.2 (Deletion)**.** *Let $\Gamma = (P, Q, F)$ be an access structure. If S^0 and S^1 are basis matrices for a (Γ, m)-VCS_2, S'^0 and S'^1 are basis matrices for a $(\Gamma, m-k)$-VCS_2, where S'^0 and S'^1 are obtained from S^0 and S^1 by deleting the same k columns.*

Proof. Assume that $b_1, b_2, \ldots, b_k$ are the columns deleted from S^0 and S^1. Let $B = b_1||b_2||\cdots||b_k$. For $X \in Q$, $w(S'^0, X) = w(S^0, X) - w(B, X) = t_X - w(B, X)$

and $w(S'^1, X) = w(S^1, X) - w(B, X) \geq t_X + m \cdot \alpha(m) - w(B, X)$ or $w(S'^1, X) = w(S^1, X) - w(B, X) \leq t_X - m \cdot \alpha(m) - w(B, X)$. Let $t'_X = t_X - w(B, X)$, $m' = m - k$ and $\alpha(m') = m \cdot \alpha(m)/m'$. Then, S'^0 and S'^1 meets the contrast requirement of VCS_2.

For $X \in F$, after deleting the same columns, S'^0 and S'^1 still meet the security requirements of VCS_2. Therefore,S'^0 and S'^1 are basis matrices for a (Γ, m')-VCS_2. □

We can exchange the roles of S^0 and S^1 in a VCS_2. Therefore, if we find a VCS_2 for an access structure, we have another one immediately.

Theorem 3.1.3 (Inverse)**.** *Let $\Gamma = (P, Q, F)$ be an access structure. If S^0 and S^1 are basis matrices for a (Γ, m)-VCS_2, S'^0 and S'^1 are basis matrices for a (Γ, m)-VCS_2, where $S'^0 = S^1$ and $S'^1 = S^0$.*

Proof. For each $X \in Q$, we set t'_X to be $t_X + m \cdot \alpha(m)$ if $w(S^1, X) \geq t_X + m \cdot \alpha(m)$ and to be $t_X - m \cdot \alpha(m)$ if $w(S^1, X) \leq t_X - m \cdot \alpha(m)$. Then, for each $X \in Q$, $w(S'^1, X) = w(S^0, X) \leq t'_X - m \cdot \alpha(m)$ or $w(S'^1, X) = w(S^0, X) \geq t'_X - m \cdot \alpha(m)$.

The security requirements are not affected by exchanging S^0 and S^1. □

We can add a participant such that Q is augmented.

Theorem 3.1.4. *Let $\Gamma = (P, Q, F)$ be an access structure and $x \notin P$. If there exists a (Γ, m)-VCS_2 with bases, there exists a (Γ', m)-VCS_2 with bases, where $\Gamma' = (P \cup \{x\}, Q \cup \{\{x\}\}, F)$.*

Proof. Without loss of generality, let x be the $(n+1)$-th element in $P \cup \{x\}$. Let S^0 and S^1 be the basis matrices for a (Γ, m)-VCS_2. It is easy to see that

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

are basis matrices for a (Γ', m)-VCS_2. □

Theorem 3.1.5. *Let $\Gamma = (P, Q)$ be a complete access structure and $x \notin P$. If there exists a (Γ, m)-VCS_2 with bases, there exists a (Γ', m)-VCS_2 with bases, where $\Gamma' = (P \cup \{x\}, Q \cup \{X \cup \{x\} | X \in Q\})$.*

Proof. Without loss of generality, let x be the $(n+1)$-th participant in $P \cup \{x\}$. Let S^0 and S^1 be the basis matrices for a (Γ, m)-VCS_2. It is easy to see that

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

are basis matrices for a (Γ', m)-VCS_2. □

Theorem 3.1.6. *Let $\Gamma = (P, Q, F)$ be an access structure and $x \notin P$. If there exists a (Γ, m)-VCS_2 with bases, there exists a $(\Gamma', m+1)$-VCS_2 with bases, where $\Gamma' = (P \cup \{x\}, Q \cup \{X \cup \{x\} | X \subseteq P\}, F)$.*

Proof. Without loss of generality, let x be the $(n+1)$-th element in $P \cup \{x\}$. Let S^0 and S^1 be the basis matrices for a (Γ, m)-VCS_2. Let

$$S'^0 = \begin{bmatrix} & & & 0 \\ & S^0 & & \vdots \\ & & & 0 \\ 1 & \cdots & 1 & 0 \end{bmatrix}, S'^1 = \begin{bmatrix} & & & 0 \\ & S^1 & & \vdots \\ & & & 0 \\ 1 & \cdots & 1 & 1 \end{bmatrix} \text{ and } \alpha(m+1) = 1/(m+1).$$

For every $X \in Q' = Q \cup \{X \cup \{x\} | X \subseteq P\}$, if $X \in Q$, we have $w(S'^0, X) = w(S^0, X)$ and $w(S'^1, X) = w(S^1, X)$. If $x \in X$, we have $w(S'^0, X) = m$ and $w(S'^1, X) = m+1$, where $t_X = m$. Thus, S'^0 and S'^1 meet the contrast property. Since all forbidden sets are in F, S'^0 and S'^1 meet the security requirement. Therefore, S'^0 and S'^1 are basis matrices for a $(\Gamma', m+1)$-VCS_2. □

We can construct a VCS_2 for Γ' from a VCS_2 for Γ when Γ' is obtained by adding an additional participant x to Γ such that some sets containing x are forbidden.

Theorem 3.1.7. *Let $\Gamma = (P, Q, F)$ be an access structure and $x \notin P$. If there exists a (Γ, m)-VCS_2 with bases, there exists a (Γ', m)-VCS_2 with bases, where $\Gamma' = (P \cup \{x\}, Q, F \cup \{X \cup \{x\} | X \in F\})$.*

Proof. Without loss of generality, let x be the $(n+1)$-th element in $P \cup \{x\}$. Let S^0 and S^1 be the basis matrices for a (Γ, m)-VCS_2. It is easy to see that

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

are basis matrices for a (Γ', m)-VCS_2. □

Corollary 3.1.1. *Let $\Gamma = (P, Q, F)$ be an access structure and $x \notin P$. If there exists a (Γ, m)-VCS_2 with bases, there exist a (Γ', m)-VCS_2 with bases and a (Γ'', m)-VCS_2 with bases, where $\Gamma' = (P \cup \{x\}, Q, F \cup \{\{x\}\})$, and $\Gamma'' = (P \cup \{x\}, Q, F)$.*

We can concatenate the basis matrices of two VCS_2's if their access structures satisfy some conditions.

Theorem 3.1.8 (Composition)**.** *Let $\Gamma_1 = (P, Q_1, F_1)$ and $\Gamma_2 = (P, Q_2, F_2)$ be two access structures. Assume that $Q_1 \cap Q_2 = \emptyset$. If there exist a (Γ_1, m_1)-VCS_2 with bases and a (Γ_2, m_2)-VCS_2 with bases, there exists a $(\Gamma, m_1 + m_2)$-VCS_2 with bases, where $\Gamma = (P, Q_1 \cup Q_2, F_1 \cap F_2)$.*

Proof. Let S_1^0 and S_1^1 be basis matrices for a (Γ_1, m_1)-VCS_2 and S_2^0 and S_2^1 be basis matrices for a (Γ_2, m_2)-VCS_2. We show that $S^0 = S_1^0||S_2^0$ and $S^1 = S_1^1||S_2^1$ with $m = m_1+m_2$ and $\alpha(m) = \min\{m_1 \cdot \alpha(m_1), m_2 \cdot \alpha(m_2)\}/(m_1+m_2)$ are basis matrices for a (Γ, m)-VCS_2.

Let $Q = Q_1 \cup Q_2$ and $F = F_1 \cap F_2$. For $X \in Q$, if $X \in Q_1 \cap F_2$, we have

$$\begin{aligned} |w(S^0, X) - w(S^1, X)| &= |w(S_1^0, X) + w(S_2^0, X) - w(S_1^1, X) - w(S_2^1, X)| \\ &\geq |w(S_1^0, X) - w(S_1^1, X)| \\ &\geq m \cdot \alpha(m); \end{aligned}$$

if $X \in F_1 \cap Q_2$, we have

$$\begin{aligned} |w(S^0, X) - w(S^1, X)| &= |w(S_1^0, X) + w(S_2^0, X) - w(S_1^1, X) - w(S_2^1, X)| \\ &\geq |w(S_2^0, X) - w(S_2^1, X)| \\ &\geq m \cdot \alpha(m). \end{aligned}$$

Thus, S^0 and S^1 meet the contrast requirement.

For $X \in F$, since $X \in F_1 \cap F_2$, the matrix obtained by restricting S^t to rows of X is that obtained by restricting S_1^t and S_2^t to rows of X, $t \in \{0, 1\}$. Since S_1^0 and S_1^1 (S_2^0 and S_2^1) meet the security requirement, S^0 and S^1 meet the security requirement. □

Even if the participant sets are not the same, we can modify the basis matrices a bit and concatenate them.

Corollary 3.1.2. *Let $\Gamma_1 = (P_1, Q_1, F_1)$ and $\Gamma_2 = (P_2, Q_2, F_2)$ be two access structures. Assume that $Q_1 \cap Q_2 = \emptyset$. If there exist a (Γ_1, m_1)-VCS_2 with bases and a (Γ_2, m_2)-VCS_2 with bases, there exists a $(\Gamma, m_1 + m_2)$-VCS_2 with bases, where $\Gamma = (P_1 \cup P_2, Q_1 \cup Q_2, F_1 \cap F_2)$.*

Proof. By Theorem 3.1.7, we can construct basis matrices for (Γ_1', m_1)-VCS_2 and (Γ_2', m_2)-VCS_2, where $\Gamma_1' = (P_1 \cup P_2, Q_1, F_1)$ and $\Gamma_2' = (P_1 \cup P_2, Q_2, F_2)$. Then, by Theorem 3.1.8, we concatenate the basis matrices of (Γ_1', m_1)-VCS_2 and (Γ_2', m_2)-VCS_2. □

3.2 Some Results

We now present some results that are useful for constructing VCS_2 for general access structures.

3.2.1 Optimal VCS_2 for (n, n)-Threshold Access structure

Let S^0 be the $n \times 2^{n-1}$ matrix whose columns are those that have exactly an even number of $1's$ and S^1 be the $n \times 2^{n-1}$ matrix whose columns are those that have exactly an odd number of $1's$. Then, S^0 and S^1 are the optimal basis matrices for a VCS_1 for the (n, n)-threshold access structure. This construction is optimal for VCS_2, too, that is, any VCS_2 with bases must have $n \times m$ basis matrices with $m \geq 2^{n-1}$ and $\alpha(m) \leq 1/2^{n-1}$.

Theorem 3.2.1. *[20] Any VCS_2 with bases for the (n, n)-threshold access structure must have $m \geq 2^{n-1}$ and $\alpha(m) \leq 1/2^{n-1}$.*

3.2.2 Q with a Single Qualified Set

Let $\Gamma = (P, Q)$ be a complete access structure such that Q contains a single set $X = \{i_1, i_2, \ldots, i_q\}$ only. We construct $n \times 2^{q-1}$ matrices S^0 and S^1 for a $(\Gamma, 2^{q-1})$-VCS_2 from a VCS_2 for the (q, q)-threshold access structure.

Theorem 3.2.2. *Let $\Gamma = (P, \{X\})$ be a complete access structure with $X = \{i_1, i_2, \ldots, i_q\}$. There exist basis matrices for a $(\Gamma, 2^{q-1})$-VCS_2.*

Proof. Let P_X be the set of participants in X. $\Gamma' = (P_X, \{X\})$ is a (q, q)-threshold access structure. Let S'^0 and S'^1 be the optimal basis matrices for a $(\Gamma', 2^{q-1})$-VCS_2, as shown in Section 3.2.1. By Theorem 3.1.7, we add the participants of $P - P_X$ to the participant set one by one and get $n \times 2^{q-1}$ basis matrices S^0 and S^1 for a $(\Gamma, 2^{q-1})$-VCS_2, where the i_jth row of S^t is the jth row of S'^t, $1 \leq j \leq q$, and all other rows are 1's, $t \in \{0, 1\}$. □

3.2.3 The Cumulative Array Method

We review the cumulative array method that constructs a VCS_1 for a complete monotonic access structure $\Gamma = (P, Q)$ [1, 24]. Assume that $P = \{1, 2, \ldots, n\}$. We define Z_{MF} to be the collection of the maximal forbidden sets in $F = 2^P - Q$, i.e.,

$$Z_{MF} = \{B \in F | B \cup \{i\} \in Q \text{ for all } i \in P \backslash B\}.$$

Assume that $Z_{MF} = \{z_1, z_2, \ldots, z_m\}$. We define the $n \times m$ Boolean matrix

$$CA_{Z_{MF}} = [a_{i,j}]_{n \times m}, \text{ where } a_{i,j} = 0 \text{ if and only if participant } i \in z_j.$$

Let $A_i = \{j | a_{i,j} = 1, 1 \leq j \leq m\}$, $1 \leq i \leq n$. Let S'^0 and S'^1 be the optimal $m \times 2^{m-1}$ basis matrices for a VCS_1 of the (m, m)-threshold access structure. Then, S^0 and S^1 constitute basis matrices for a VCS_1 for Γ, where

the ith row of S^t is $OR(S'^t, A_i)$,

for $1 \leq i \leq n$ and $t \in \{0, 1\}$.

3.2.4 An Upper Bound for 2-out-n Access Structure

We now give an upper bound for pixel expansion of any VCS_2 for the special 2-out-n access structures. $\Gamma = (P, Q)$ is the 2-out-n access structure if $|P| = n$ and $Q = \{X \subseteq P : |X| = 2\}$. We present a VCS_2 with bases for the 2-out-n access structure.

Theorem 3.2.3. *There is a VCS_2 with pixel expansion $m(n)$ and contrast $1/m(n)$ for the 2-out-n access structure such that*

$$m(n) = \begin{cases} \frac{(n-1)(n+3)}{4} & \text{if } n \text{ is odd} \\ \frac{n(n+2)}{4} & \text{if } n \text{ is even} \end{cases}$$

Proof. Let $b_{i,j}$ be the n-dimensional column vector whose ith and jth entries are 0 and all other entries are 1, $1 \leq i < j \leq n$. Let c_i be the n-dimensional column vector whose ith entry is 0 and all other entries are 1. Let $\vec{1}$ be the n-dimensional vector of all entries being 1.

For the case $n = 2m + 1$, we let S^0 contain all $b_{i,j}$'s with $i + j$=odd and S^1 contain all $b_{i,j}$'s with $i + j$=even. Furthermore, we add 2 copies of c_i to S^1 for even i, $1 \leq i \leq n$, and m copies of $\vec{1}$ to S^0. For example, the following are basis matrices of a VCS_2 for the 2-out-5 access structure:

$$S^0 = \begin{bmatrix} 00111111 \\ 01001111 \\ 11010111 \\ 10110011 \\ 11101011 \end{bmatrix}, \; S^1 = \begin{bmatrix} 00111111 \\ 11010011 \\ 01101111 \\ 11011100 \\ 10101111 \end{bmatrix}$$

There are $m^2 + 2m$, which is $(n-1)(n+3)/4$, columns in S^0 and S^1.

We now consider the contrast and security properties of this construction. Since there is only one $b_{i,j}$ column in either S^0 or S^1, for any two participants i and j, we have $|w(S^0, \{i, j\}) - w(S^1, \{i, j\})| = 1$. For any X containing 3 or more participants $i_1, i_2, \ldots, i_k$, $k \geq 3$, we have $w(S^0, \{i_1, i_2, \ldots, i_k\}) = w(S^1, \{i_1, i_2, \ldots, i_k\}) = m(n)$ since each column has at most two 0's. For any X containing only one participant i, row i of S^0 contains m 0's if i is odd and $m + 1$ 0's if i is even. This holds for S^1 also. Therefore, any single participant computes absolutely no information about the secret from his share.

For the case $n = 2m$, we let S^0 contain all $b_{i,j}$'s with $i + j$=odd and S^1 contain all $b_{i,j}$'s with $i + j$=even. Furthermore, we add a copy of c_i to S^1, $1 \leq i \leq n$, and m copies of $\vec{1}$ to S^0. For example, the following are basis matrices of a VCS_2 for the 2-out-4 access structure:

$$S^0 = \begin{bmatrix} 001111 \\ 010111 \\ 110011 \\ 101011 \end{bmatrix}, \; S^1 = \begin{bmatrix} 010111 \\ 101011 \\ 101101 \\ 011110 \end{bmatrix}$$

There are $m^2 + m$, which is $n(n + 2)/4$, columns in S^0 and S^1.

We can discuss the contrast and security properties for this construction similarly. This completes the proof. □

Droste's VCS_1 construction for the 2-out-n access structure has the pixel expansion $m = C_2^n \cdot \sum_{i=1}^{n} (2^i \cdot C_i^n)$ [9]. By the cumulative array method, the VCS_1 construction for the 2-out-n access structure has pixel expansion $m = 2 \cdot C_2^n$. We are aware that there are (2,n)-threshold VCS_1 that have pixel expansion $m = 2\lceil \log n \rceil$ [1]. However, the 2-out-n access structure is different from the $(2, n)$-threshold access structure. The later one allows more than two participants to reveal the secret, while the former one does not.

3.3 Partition of Access Structures

For a given access structure $\Gamma = (P, Q, F)$, we can decompose it into smaller access structures $\Gamma_1 = (P, Q_1, F_1), \Gamma_2 = (P, Q_2, F_2), \ldots, \Gamma_k = (P, Q_k, F_k)$ such that

1. $Q_1 \cup Q_2 \cup \cdots \cup Q_k = Q$;
2. $Q_i \cap Q_j = \emptyset$ for $1 \leq i \neq j \leq k$;
3. $F_1 \cap F_2 \cap \cdots \cap F_k = F$.

We call such decomposition as a *partition* of Γ. By generalizing Theorem 3.1.8, we can concatenate the smaller basis matrices for (Γ_i, m_i)-VCS_2's to form basis matrices for a (Γ, m)-VCS_2.

Theorem 3.3.1 (Partition)**.** *Let $\Gamma_1, \Gamma_2, \ldots, \Gamma_k$ be a partition of the access structure Γ. Assume that S_i^0 and S_i^1 are basis matrices for a (Γ_i, m_i)-VCS_2. Then, $S_1^0||S_2^0||\cdots||S_k^0$ and $S_1^1||S_2^1||\cdots||S_k^1$ are basis matrices for a $(\Gamma, \sum_{i=1}^{k} m_i)$-$VCS_2$.*

Proof. This is proved by induction on k, $k \geq 2$. The induction basis holds by Theorem 3.1.8. The induction step follows easily. □

3.3.1 An Upper Bound for General Access Structures

By the results in Theorems 3.2.2 and 3.3.1, we give an upper bound on pixel expansion for any access structure.

Theorem 3.3.2. *Let $\Gamma = (P, Q, F)$ be an access structure. There exists a (Γ, m)-VCS_2 with bases, where $m = \sum_{X \in Q} 2^{|X|-1}$.*

Proof. Let Q be $\{X_1, X_2, \ldots, X_k\}$ and $\Gamma' = (P, Q)$. Since any (Γ, m)-VCS_2 is a (Γ', m)-VCS_2, we consider only $\Gamma' = (P, Q)$. We partition $\Gamma' = (P, Q)$ into $(P, \{X_1\}), (P, \{X_2\}), \ldots, (P, \{X_k\})$. For each $\Gamma_i = (P, \{X_i\})$, we construct $n \times 2^{|X_i|-1}$ basis matrices for a VCS_2 of Γ_i. Since $2^P - Q = \bigcap_{i=1}^{k} 2^P - \{X_i\}$, by Theorem 3.3.1 we concatenate these basis matrices to get basis matrices for a (Γ', m)-VCS_2, where $m = \sum_{i=1}^{k} 2^{|X_i|-1}$. □

3.4 VCS_2 Construction for General Access Structure

We present two methods of constructing basis matrices for a VCS_2 of an arbitrary access structure. Without loss of generality, we consider a complete access structure $\Gamma = (P, Q)$, where $P = \{1, 2, \ldots, n\}$ is the set of participants. In case that the input access structure is not complete, we add the "don't care" participant sets into F and form a complete access structure.

3.4.1 Top-Down Approach

The idea of our first construction is to partition Q into maximal monotonic subsets Q_i, $1 \leq i \leq k$, and use the methods in Sections 3.2.2 and 3.2.3 to construct the basis matrices for these access structures (P, Q_i). Then, by Theorem 3.3.1, we concatenate these basis matrices for a (Γ, m)-VCS_2.

Our algorithm A1 is in Figure 3.1. We first pick a qualified set X with a maximum number of participants and incorporate as many qualified sets under X as possible. That is, for each picked X, we find the maximum monotonic collection Z_{MMQ} of qualified sets under X:

$$Z_{MMQ}(X, Q) = \{X' | X' \in Q, \text{ there is no } Y \in 2^{P_X} - Q \text{ such that } X' \subset Y \subset X\}.$$

Input: $\Gamma = (P, Q)$, where $F = 2^P - Q$.
1. if $Q = \emptyset$, return $S^0 = \mathbf{0}_{n\times 1}$ and $S^1 = \mathbf{0}_{n\times 1}$;
2. $A \leftarrow Q$; $i \leftarrow 0$;
3. while $A \neq \emptyset$ do
4. $\quad i \leftarrow i + 1$;
5. $\quad$ let X_i be the maximum set in A; (break tie randomly)
6. $\quad Z_i \leftarrow Z_{MMQ}(X_i, A)$;
7. $\quad A \leftarrow A - Z_i$;
8. $k \leftarrow i$;
9. construct basis matrices S_i^0 and S_i^1 for $\Gamma_i = (P_{X_i}, Z_i)$
   and extend them to T_i^0 and T_i^1 for $\Gamma_i' = (P, Z_i)$, $1 \le i \le k$;
10. return $S^0 = T_1^0||T_2^0||\cdots||T_k^0$ and $S^1 = T_1^1||T_2^1||\cdots||T_k^1$.

Figure 3.1: A1: Partition Q and find basis matrices.

Let $\Gamma_1 = (P_X, Z_{MMQ}(X, Q))$. Note that by our definition, Γ_1 is monotonic. We then subtract $Z_{MMQ}(X, Q)$ from Q and continue to find Γ_2, and so on. This process does not stop until Q becomes empty.

We give an example to illustrate this partition. Let $P = \{1, 2, 3, 4, 5\}$, $Q = \{\{1, 3\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{1, 2, 3\}, \{1, 3, 4, 5\}, \{2, 3, 4, 5\}, \{1, 2, 3, 4, 5\}\}$ and $F = 2^P - Q$. First, we choose the maximum set $X_1 = \{1, 2, 3, 4, 5\}$ and set $Z_1 = Z_{MMQ}(X_1, Q) = \{\{1, 3, 4, 5\}, \{2, 3, 4, 5\}, \{1, 2, 3, 4, 5\}\}$. Therefore, $\Gamma_1 = (P_{X_1}, Z_1)$. Then, we subtract Z_1 from Q. Q becomes $\{\{1, 3\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{1, 2, 3\}\}$. We select $X_2 = \{1, 2, 3\}$ and set $Z_2 = Z_{MMQ}(X_2, Q) = \{\{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. Therefore, $\Gamma_2 = (P_{X_2}, Z_2)$. This process continues and we get $\Gamma_3 = (P_{X_3}, Z_3)$ and $\Gamma_4 = (P_{X_4}, Z_4)$, where $X_3 = \{3, 4\}$, $X_4 = \{4, 5\}$, $Z_3 = \{\{3, 4\}\}$ and $Z_4 = \{\{4, 5\}\}$.

After finding a partition Γ_i, $1 \le i \le k$, of Γ, we construct a VCS_2 for each $\Gamma_i = (P_{X_i}, Z_i)$. If Z_i contains only a single qualified set X_i, we use the method in Section 3.2.2 to construct basis matrices S_i^0 and S_i^1 for a (Γ_i, m_i)-VCS_2, where $m_i = 2^{|X_i|-1}$. If Z_i contains two or more qualified sets, we use the cumulative method in Section 3.2.3 to construct S_i^0 and S_i^1 for a (Γ_i, m_i)-VCS_2, where m_i is the parameter implied by the cumulative method. By Theorem 3.1.7, we extend S_i^0 and S_i^1 to basis matrices T_i^0 and T_i^1 for a (Γ_i', m_i)-VCS_2, where $\Gamma_i' = (P, Z_i)$. Note that Γ_i and Γ_i' differ on the participant set.

We continue the example and compute

$$T_1^0 = \begin{bmatrix} 00010111 \\ 00010111 \\ 00101011 \\ 01001101 \\ 01110001 \end{bmatrix}, \quad T_1^1 = \begin{bmatrix} 00010111 \\ 00010111 \\ 00101011 \\ 01001101 \\ 10001110 \end{bmatrix}, \quad T_2^0 = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 11 \\ 11 \end{bmatrix}, \quad T_2^1 = \begin{bmatrix} 10 \\ 10 \\ 01 \\ 11 \\ 11 \end{bmatrix},$$

$$T_3^0 = \begin{bmatrix} 11 \\ 11 \\ 10 \\ 10 \\ 11 \end{bmatrix}, \quad T_3^1 = \begin{bmatrix} 11 \\ 11 \\ 10 \\ 01 \\ 11 \end{bmatrix}, \quad T_4^0 = \begin{bmatrix} 11 \\ 11 \\ 11 \\ 10 \\ 10 \end{bmatrix}, \quad \text{and} \quad T_4^1 = \begin{bmatrix} 11 \\ 11 \\ 11 \\ 10 \\ 01 \end{bmatrix}.$$

By concatenating these basis matrices, we get basis matrices S^0 and S^1 for a (Γ, m)-VCS_2 with $m = 14$, $\alpha(m) = 1/14$,

$$S^0 = \begin{bmatrix} 00010111101111 \\ 00010111101111 \\ 00101011101011 \\ 01001101111010 \\ 01110001111110 \end{bmatrix} \quad \text{and} \quad S^1 = \begin{bmatrix} 00010111101111 \\ 00010111101111 \\ 00101011011011 \\ 01001101110110 \\ 10001110111101 \end{bmatrix}.$$

If we use Droste's method [9] directly to construct basis matrices for a (Γ, m)-VCS_1, we get $m = 44$ and $\alpha(m) = 1/44$. In the next section, we apply the techniques implied in Theorems 3.1.2 and 3.1.3 to improve this m and $\alpha(m)$ to 6 and 1/6, respectively.

We now show correctness of our construction.

Theorem 3.4.1. *The algorithm A1 in Figure 3.1 outputs basis matrices for a* (Γ, m)-VCS_2.

Proof. We only have to show that $\Gamma'_1, \Gamma'_2, \ldots, \Gamma'_k$ form a partition of $\Gamma = (P, Q)$ and T_i^0 and T_i^1 are the basis matrices for a (Γ'_i, m)-VCS_2. The later one holds by the constructions in Sections 3.2.2 and 3.2.3. For the former one, by the definition of $Z_{MMQ}(X, Q)$, $\Gamma_i = (P_X, Z_{MMQ}(X, Q)$ is a complete access structure over P_X. By the algorithm, the next Γ_{i+1} is computed from Q', where $Q' = Q - Z_{MMQ}(X, Q)$. Therefore, Γ'_i, $1 \leq i \leq k$, form a partition for Γ. □

3.4.2 Further Improvement

By Theorem 3.1.3, if S^0 and S^1 are basis matrices for a (Γ, m)-VCS_2, S'^0 and S'^1 are also basis matrices for a (Γ, m)-VCS_2, where $S'^0 = S^1$ and $S'^1 = S^0$. In Step 9 of A1 in Figure 3.1, for each Γ'_i, we actually have two VCS_2's with bases: one is (T_i^0, T_i^1)

and the other is $(T_i'^0, T_i'^1)$, where $T_i'^0 = T_i^1$ and $T_i'^1 = T_i^0$. Therefore, we have 2^k (Γ, m)-VCS_2's in total. By searching among these schemes and removing redundant columns, we can find a VCS_2 with better contrast. For example, continuing the example of the previous section, we let

$$S^0 = T_1^1||T_2^0||T_3^1||T_4^0 = \begin{bmatrix} 000101111011111 \\ 000101111011111 \\ 001010111010111 \\ 010011011101110 \\ 100011101111110 \end{bmatrix}$$

and

$$S^1 = T_1^0||T_2^1||T_3^0||T_4^1 = \begin{bmatrix} 000101111011111 \\ 000101111011111 \\ 001010110110111 \\ 010011011110101 \\ 011100011111101 \end{bmatrix}.$$

By Theorem 3.1.2, we delete equal columns from S^0 and S^1 and get

$$S'^0 = \begin{bmatrix} 000111 \\ 000111 \\ 001010 \\ 010001 \\ 100011 \end{bmatrix} \text{ and } S'^1 = \begin{bmatrix} 001011 \\ 001011 \\ 010100 \\ 000110 \\ 011001 \end{bmatrix},$$

which have $m = 6$ and $\alpha(m) = 1/6$.

Lemma 3.4.1. *Any $S^0 = T_1^{t_1}||T_2^{t_2}||\cdots||T_k^{t_k}$ and $S^1 = T_1^{\bar{t}_1}||T_2^{\bar{t}_2}||\cdots||T_k^{\bar{t}_k}$ are basis matrices for a (Γ, m)-VCS_2, where $t_i \in \{0, 1\}$ and $\bar{t}_i$ is the complement of t_i, $1 \leq i \leq k$.*

Proof. By Theorem 3.1.3, (T_i^0, T_i^1) and (T_i^1, T_i^0) are both basis matrix pair for a (Γ_i', m_i)-VCS_2, $1 \leq i \leq k$. By Theorem 3.3.1 for composition of a partition, this lemma holds. □

Though to find S^0 and S^1 with minimal pixel expansion among the 2^k VCS_2's is NP-complete, we provide a dynamic programming-type heuristic method to find a reasonable one.

We assume a canonical order $b_1, b_2, \cdots, b_{2^n}$ for n-dimensional Boolean vectors. Let $f_i^t = (i_1, i_2, \ldots, i_{2^n})$ be the *column spectrum* of T_i^t, $t \in \{0, 1\}, 1 \leq i \leq k$, such that i_j is the number of b_j in columns of T_i^t. For example, if

$$T_i^0 = \begin{bmatrix} 000011 \\ 010001 \\ 010001 \end{bmatrix},$$

Input: $T_i^0, T_i^1, 1 \le i \le k$;
1. compute f_i^0 and f_i^1, $1 \le i \le k$;
2. for $z = 0$ to $k-1$ do
3. for $i = 1$ to $k-z$ do
4. compute $m(i, i+z)$ and record t_l, $i \le l \le i+z$;
5. let $t_l, 1 \le l \le k$, be the indices by which $m(1,k)$ is computed;
6. return $S^0 = T_1^{t_1}||T_2^{t_2}||\cdots||T_k^{t_k}$ and $S^1 = T_1^{\bar{t}_1}||T_2^{\bar{t}_2}||\cdots||T_k^{\bar{t}_k}$.

Figure 3.2: Search a VCS_2 with better pixel expansion.

then $f_i^0 = (3,1,0,0,0,0,1,1)$ is its column spectrum, where $b_1 = [0\ 0\ 0]^T$, $b_2 = [1\ 0\ 0]^T$, etc. For a spectrum $f = (i_1, i_2, \ldots, i_{2^n})$, let $|f| = \sum_{j=1}^{2^n} |i_j|$. Let $m(i,j)$ denote the differential column spectrum between

$$S_{i,j}^0 = T_i^{t_i}||T_{i+1}^{t_{i+1}}||\cdots||T_j^{t_j} \text{ and } S_{i,j}^1 = T_i^{\bar{t}_i}||T_{i+1}^{\bar{t}_{i+1}}||\cdots||T_j^{\bar{t}_j}$$

for some $t_l \in \{0,1\}, i \le l \le j$, where $m(i,j)$ is defined recursively as follows:

$$m(i,j) = \begin{cases} f_i^0 - f_i^1 & \text{if } i = j \\ \min_{i \le l \le j}\{m(i,l) + m(l+1,j)m(i,l) - m(l+1,j)\} & \text{if } i > j, \end{cases}$$

where $\min\{v_1, v_2, \ldots, v_r\} = v_i$ if $|v_i| \le |v_j|$ for all j, $1 \le j \le r$ (we break tie randomly). That is, $m(i,j)$ is the difference of the column spectrums of $S_{i,j}^0$ and $S_{i,j}^1$. We can see that the smaller $|m(i,j)|$ is, the smaller the pixel expansion $S_{i,j}^0$ and $S_{i,j}^1$ have after deleting equal columns. Our goal is to find smaller $|m(1,k)|$. The search algorithm is shown in Figure 3.2. During computing $m(i, i+z)$, we keep track the choice of t_l, $i \le l \le i+z$, in order to compute the indices for $m(1,k)$.

3.4.3 Bottom-Up Approach

Our second method uses the bottom-up approach. For a qualified set $X \in Q$, we define the collection of the qualified sets Y that contain X such that all sets between X and Y are qualified:

$$M(X,Q) = \{Y | X \subseteq Y, \text{ for all } X' \subseteq Y - X, X \cup X' \in Q\}.$$

$M(X,Q)$ is not empty since $X \in M(X,Q)$. For any $Y \in M(X,Q)$, let $B(X,Y) = \{X' | X \subseteq X' \subseteq Y\}$.

Input: $\Gamma = (P, Q)$, where $F = 2^P - Q$.
1. if $Q = \emptyset$, return $S^0 = \mathbf{0}_{n\times 1}$ and $S^1 = \mathbf{0}_{n\times 1}$;
2. $A \leftarrow Q$; $i \leftarrow 0$;
3. while $A \neq \emptyset$ do
4. $\quad i \leftarrow i + 1$;
5. $\quad$ let X_i be the minimum set in A; (break tie randomly)
6. $\quad$ let Y_i be the maximum set in $M(X_i, A)$; (break tie randomly)
7. $\quad A \leftarrow A - Q(X_i, Y_i)$;
8. $k \leftarrow i$;
9. construct basis matrices S_i^0 and S_i^1 for $\Gamma_i = (P, Q(X_i, Y_i))$,
$\quad$ as shown in Lemma 3.4.2;
10. return $S^0 = S_1^0||S_2^0||\cdots||S_k^0$ and $S^1 = S_1^1||S_2^1||\cdots||S_k^1$.

Figure 3.3: A2: Bottom-up partition Q and find basis matrices.

Lemma 3.4.2. *$\Gamma' = (P, B(X, Y))$ have a VCS_2 with $n \times 2^{|X|-1}$ basis matrices S^0 and S^1, where the rows of S^0 (S^1) for X is the S'^0 (S'^1) of the optimal $(|X|, |X|)$-VCS_1, the rows of S^0 (S^1) for $Y - X$ are all 0 and the rows of S^0 (S^1) for $P - Y$ are all 1.*

Proof. By Theorem 3.1.5, we extend $\Gamma' = (P_X, \{X\})$ to $\Gamma'' = (P_Y, B(X, Y))$ and by Theorem 3.1.7, we extend $\Gamma'' = (P_Y, B(X, Y))$ to $\Gamma = (P, B(X, Y))$. The basis matrices S^0 and S^1 are constructed accordingly. □

For example, for $\Gamma = (\{1, 2, 3, 4\}, \{\{2, 3\}, \{1, 2, 3\}, \{2, 4\}\})$ and $X = \{2, 3\}$, $M(X) = \{\{1, 2, 3\}\}$ and $\Gamma' = (\{1, 2, 3, 4\}, \{\{2, 3\}, \{1, 2, 3\}\})$ has a VCS_2 with

$$S^0 = \begin{bmatrix} 00 \\ 01 \\ 01 \\ 11 \end{bmatrix} \text{ and } S^1 = \begin{bmatrix} 00 \\ 10 \\ 01 \\ 11 \end{bmatrix}.$$

The algorithm A2 based on bottom-up partition is shown in Figure 3.3. We reduce the pixel expansion by applying the algorithm in Figure 3.2.

3.5 Experiments and Comparison

We compare the results of our two methods on random access structures with those of the Droste's method, which is the most efficient method of constructing VCS_1 for arbitrary access structures. The experimental results show that our VCS_2's indeed have better pixel expansion (contrast) in average.

We implement A1, A2 and the Droste's method for arbitrary access structures. The columns of the basis matrices produced by A1 and A2 are reduced by the search algorithm in Figure 3.2. We also remove redundant columns in basis matrices produced by the Droste's method. For a particular number of participants, we run these algorithms on a number of randomly chosen access structures. The results are shown in Tables 3.1, 3.2 and 3.3. In Table 3.1, we randomly choose access structures with $|Q| \approx 2^{n-1}$. In Table 3.2, we randomly choose access structures with $|Q| \approx 2^n/3$. For both cases, the average pixel expansion of our VCS_2 for a random access structure is only one half of that of the VCS produced by the Droste's method. In Table 3.3 for monotonic access structures, the A1 algorithm takes the whole Q as a partition and produces the same result as that of the Droste's method. But, the A2 algorithm produces VCS_2 with much better pixel expansion. Table 3.4 shows two access structures that have better pixel expansion based on our definition.

the number n of participants	the number of random Γ	average pixel expansion m		
		A1	A2	Droste's
3	50	2.1	2.0	2.8
4	100	3.9	4.2	6.6
5	150	8.2	8.8	15.9
6	200	17.2	18.5	38.8
7	300	39.0	41.1	93.9
8	400	87.6	92.1	224.4

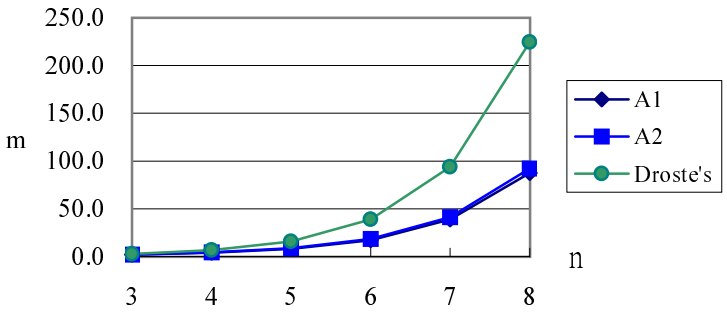


Table 3.1: Comparison of three methods with $|Q| \approx 2^{n-1}$.

the number n of participants	the number of random Γ	average pixel expansion m		
		A1	A2	Droste's
3	50	1.9	2.0	2.6
4	100	3.8	4.0	6.1
5	150	8.2	8.7	15.7
6	200	17.2	18.9	38.5
7	300	38.5	41.9	93.3
8	400	88.2	101.9	230.1

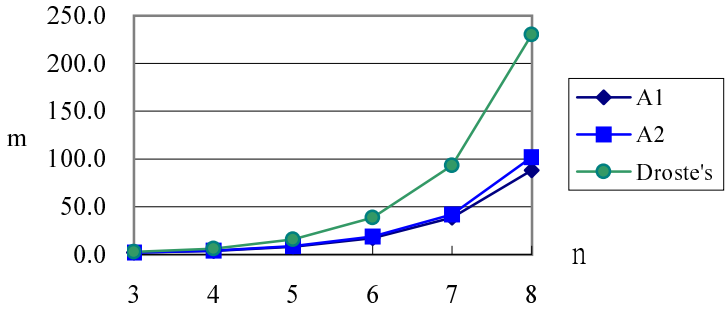


Table 3.2: Comparison of three methods with $|Q| \approx 2^n/3$.

the number n of participants	the number of random Γ	average pixel expansion m		
		A1	A2	Droste's
3	50	2.0	2.0	2.0
4	100	4.1	3.9	4.1
5	150	10.0	7.8	10.0
6	200	25.1	15.5	25.1
7	300	64.4	31.7	64.4
8	400	187.3	73.5	187.3

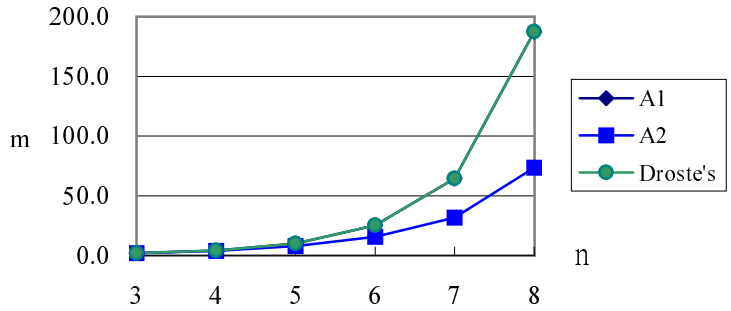


Table 3.3: Comparison of three methods with monotonic Γ.

	$P = \{1,2,3\}$, $Q = (\{1\},\{2,3\},\{1,2,3\})$, $F = 2^P - Q$
Our VCS_2	$S^0 = \begin{bmatrix} 01 \\ 01 \\ 01 \end{bmatrix}$, $S^1 = \begin{bmatrix} 00 \\ 10 \\ 01 \end{bmatrix}$
Droste's VCS	$S^0 = \begin{bmatrix} 000 \\ 101 \\ 101 \end{bmatrix}$, $S^1 = \begin{bmatrix} 001 \\ 011 \\ 101 \end{bmatrix}$
	$P = \{1,2,3,4\}$, $F = 2^P - Q$ $Q = (\{1,2\},\{1,3\},\{2,3\},\{2,4\},\{1,3,4\},\{1,2,3,4\})$
Our VCS_2	$S^0 = \begin{bmatrix} 0011 \\ 0100 \\ 0100 \\ 0011 \end{bmatrix}$, $S^1 = \begin{bmatrix} 0011 \\ 0001 \\ 1000 \\ 0101 \end{bmatrix}$
Droste's VCS	$S^0 = \begin{bmatrix} 01111100011101 \\ 01101000101011 \\ 11101101110001 \\ 11111001000111 \end{bmatrix}$, $S^1 = \begin{bmatrix} 10111110100011 \\ 01100111000101 \\ 11111100010110 \\ 11011001001111 \end{bmatrix}$

Table 3.4: Two examples of comparing our methods with Droste's.

Chapter 4

Cheating Prevention in VC

In this chapter we studied the cheating problem in VC and extended VC. We considered the attacks of malicious adversaries who may deviate from the scheme in any way. We presented three cheating methods and applied them on attacking existent VC or EVC schemes. We improved one cheat-preventing scheme. We proposed a generic method that converts a VCS to another VCS that has the property of cheating prevention. The overhead of the conversion is near optimal in both contrast degression and pixel expansion.

4.1 Cheating in VC

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

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

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

In order to cheat successfully, honest participants who present their shares for recovering the secret image should not be able to distinguish fake shares from genuine shares. A reconstructed image is *perfect black* if the subpixels associated to a black pixel of the secret image are all black. Most proposed VC schemes have the property of perfect blackness. For example, the reconstructed secret images **S** in Example 2.2.1 are all perfect black.

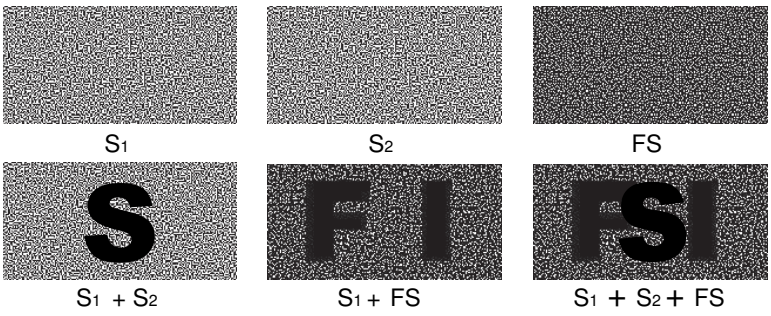


Figure 4.1: An example of cheating a $(2, 2)$-VCS.

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

Example 4.1.1. *Figure 4.1 shows how to cheat participants in a* $(2, 2)$*-VCS. Since* $S_1 + FS$ *reveals the fake image* **FI**, *Participant 1(*P_1 *for short, hereafter) is cheated to believe that the secret image is* **FI**. *Although* $S_1 + S_2 + FS$ *successfully reveals the fake image, the real secret image* **S** *also appears on* $S_1 + S_2 + FS$ *due to the property of perfect blackness for secret images. The participants of a qualified set,* $(1, 2)$ *in this example, cannot be cheated.*

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

Definition 4.1.1. *For a* (Γ, m)*-VCS with basis matrices* S^0 *and* S^1, *an* **MP** *or an* **MO** *cheats successfully if it finds a fake image and generates fake shares satisfying the following:*

1. *For* $Y = \{i_1, i_2, \ldots, i_q\} \notin Q$, *the stacking of their shares and the fake shares reveals the fake image. If the cheater is an* **MP**, *some* i_j *is the cheater,* $1 \leq j \leq q$.

2. *The fake shares cannot be distinguished from the genuine shares. Formally, for each fake share FS, there is a share* S_i *such that the subpixels of FS are identically distributed as those of* S_i.

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

1. For each white pixel of the fake image, copy the corresponding subpixels of the pixel in S_1 to each fake share.

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

Image reconstruction phase (the fake image):
Let $Y = \{1, i_1, i_2, \ldots, i_q\}$ be a set of participants. If $Y \notin Q$, the stacking of genuine shares $S_1, S_{i_1}, S_{i_2}, \ldots, S_{i_q}$ and fake shares $FS_1, FS_2, \ldots, FS_r$ shall reveal the fake image.

Figure 4.2: Cheating method **CA-1**, initiated by an **MP**.

4.2 Three Cheating Methods

Our first cheating method is initiated by an **MP**, while the second cheating method is initiated by an **MO**. Both of them applies to attack VC. Our third cheating method is initiated by an **MP** and applies to attack EVC.

4.2.1 Cheating a VCS by an MP

The cheating method **CA-1**, depicted in Figure 4.2, applies to attack any VCS. Without loss of generality, we assume that P_1 is the cheater. Since the cheater is an **MP**, he uses his genuine share as a template to construct a set of fake shares which are indistinguishable from its genuine share. The stacking of these fake shares and S_1 reveals the fake image of perfect blackness. We see that, for $Y = \{1, i_1, i_2, \ldots, i_q\} \notin Q$ the stacking of their shares reveals no images. Thus, the stacking of their shares and the fake shares reveals the fake image due to perfect blackness of the fake image.

Example 4.2.1. *Figure 4.3 shows how to cheat the participants in a* $(4,4)$*-VCS. There are four shares* S_1*,* S_2*,* S_3 *and* S_4 *in the* $(4,4)$*-VCS.* P_1 *is assumed to be the* **MP**. *By* ***CA-1****, one fake share* FS_1 *is generated. Since* $Y = (1,3,4)$ *(or* $(1,2)$*)* $\notin Q$*, we see that* $S_1 + FS_1 + S_3 + S_4$ *(or* $S_1 + FS_1 + S_2$*) reveals the fake image* **FI**. *Thus,*

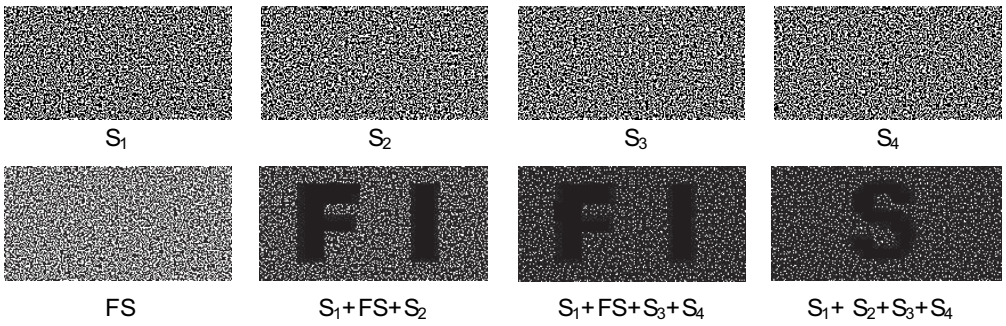


Figure 4.3: An example of cheating a $(4, 4)$-VCS by an MP.

P_3 and P_4 (or P_2) are cheated to believe that **FI** *is the secret image.*

For some prominent (n, n)- and (k, n)-VCS's [5, 6, 20], the numbers of black and white subpixels in a pixel are almost equal. The cheater needs only $r = \lceil \frac{m}{x} \rceil - 1 = 1$ fake share to cheat successfully.

Theorem 4.2.1. *The* **MP** *in* **CA-1** *successfully cheats any VCS.*

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

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

4.2.2 Cheating a VCS by an MO

Our second cheating method **CA-2**, depicted in Figure 4.4, demonstrates that an **MO** can cheat even without any genuine share at hand. The idea is as follows. We use the optimal $(2, 2)$-VCS to construct the fake shares for the fake image. Then, we tune the size of fake shares so that they can be stacked with genuine shares.

Now, the only problem is to have the right share size for the fake shares. Our solution is to try all possible share sizes. In case that the **MO** gets one genuine

Input: none.
Fake share construction phase:
The **MO** chooses a fake image and does the following:

1. Encode the fake image into two fake shares FS_1 and FS_2 with the optimal $(2, 2)$-VCS.

2. Generate enough pairs of fake shares $FS_{1,i}$ and $FS_{2,i}$ with various sizes and subpixel distributions, $1 \leq i \leq r$ for some r.

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

Figure 4.4: Cheating method **CA-2**, initiated by an **MO**.

share, there will be no such problem. It may seem difficult to have fake shares of the same size as that of the genuine shares. We give a reason to show the possibility. The shares of a VCS are usually printed in transparencies. We assume that this is done by a standard printer or copier which accepts only a few standard sizes, such as A4, A3, etc. Therefore, the size of genuine shares is a fraction, such as 1/4, of a standard size. We can simply have the fake shares of these sizes. Furthermore, it was suggested to have a solid frame to align shares [20] in order to solve the alignment problem during the image reconstruction phase. The **MO** can simply choose the size of the solid frame for the fake shares. Therefore, it is possible for the **MO** to have the right size for the fake shares.

Example 4.2.2. *Figure 4.5 shows that an **MO** cheats a $(4, 4)$-VCS. The four genuine shares S_1, S_2, S_3, and S_4 are those in Figure 4.3 and the two fake shares are FS_1 and FS_2. For clarity, we put S_1 here to demonstrate that the fake shares are indistinguishable from the genuine shares. We see that the stacking of fewer than four genuine shares and two fake shares shows the fake image* **FI**.

Theorem 4.2.2. *The* **MO** *in **CA-2** successfully cheats a VCS if the right share size is obtained.*

Proof. Contrast. For $Y = \{i_1, i_2, \ldots, i_q\} \notin Q$, let $Z_Y = S_{i_1} + S_{i_2} + \cdots + S_{i_q}$. Since FS_1 and FS_2 are two shares of the optimal $(2, 2)$-VCS, $p_b(FS_1 + FS_2) = 1$ and $p_w(FS_1 + FS_2) = 1/2$. By **CA-2**, the distribution of subpixels of the genuine shares

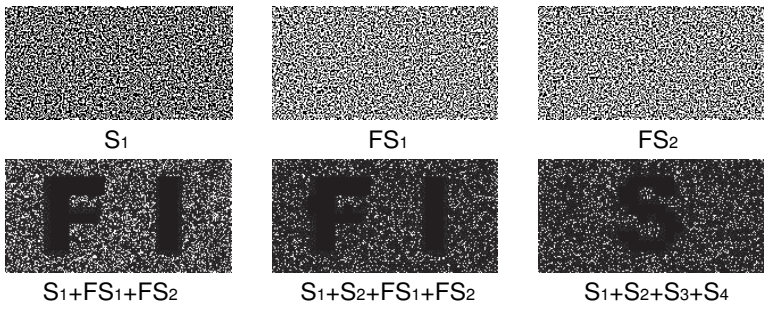


Figure 4.5: An example of cheating a $(4, 4)$-VCS by an **MO**.

are random and independent of that of the fake shares. For the white pixel in $Z_Y + FS_1 + FS_2$, we have, with high probability,

$$p_w(Z_Y + FS_1 + FS_2) = 1 - (1 - p_w(Z_Y))(1 - 1/2) = 1/2 + p_w(Z_Y)/2 < 1.$$

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

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

4.2.3 Cheating an EVCS by an MP

In the definition of VC, it only requires the contrast be non-zero. Nevertheless, we observe that if the contrast is too small, it is hard to "see" the image. Based upon this observation, we demonstrate the third cheating method **CA-3**, depicted in Figure 4.6, against an EVCS. The idea of **CA-3** is to use the fake shares to reduce the contrast between the share images and the background. Simultaneously, the fake image in the stacking of fake shares has enough contrast against the background since the fake image is recovered in perfect blackness.

Let ϵ be the threshold for contrast that human eyes distinguish the image from the background. The value ϵ varies for different sizes, contrasts and types of share images. We do some experiments to obtain ϵ empirically. We consider four types of pictures (in Figure 4.7) with four different sizes ($Z_1 : 200 \times 100$ pixels, $Z_2 : 200 \times 200$ pixels, $Z_3 : 400 \times 200$ pixels, and $Z_4 : 400 \times 400$ pixels) and four different contrasts (1/4, 1/9, 1/16, and 1/25). The values (ϵm) in Table 4.1 represent the number of black subpixels which we should add for each pixel of the fake shares in order to

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

1. Create S_1', which is S_1, but without the share image. The share image of S_1 is removed by changing d black subpixels into white subpixels in each black pixel, where d is the difference between the numbers of black subpixels of a black and a white pixel.

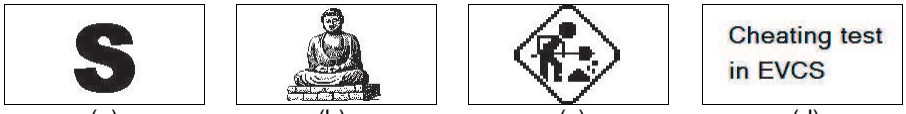
2. Create $r = \lceil \frac{m}{x} \rceil - 1$ temporary fake shares FS_i', $1 \leq i \leq r$, by using S_1' according to **CA-1**.

3. Randomly change d white subpixels into black subpixels of each pixel of the share image in FS_i', $1 \leq i \leq r$.

4. Construct FS_i by randomly adding ϵm black subpixels (changing from white subpixels) to each pixel in FS_i', $1 \leq i \leq r$. The threshold value ϵm, like those in Table 4.1, is obtained by experiments.

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

Figure 4.6: Cheating method **CA-3** against an EVCS.



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

Figure 4.7: Four different types of pictures.

	(a)				(b)				(c)				(d)			
contrast	Z_1	Z_2	Z_3	Z_4	Z_1	Z_2	Z_3	Z_4	Z_1	Z_2	Z_3	Z_4	Z_1	Z_2	Z_3	Z_4
1/4	20	22	23	24	10	13	16	19	15	20	22	24	15	17	18	21
1/9	17	21	21	23	7	8	14	17	14	20	21	22	13	16	17	19
1/16	5	7	5	7	2	2	2	3	2	4	6	9	4	4	3	6
1/25	4	6	4	6	1	1	1	2	1	3	5	8	2	2	1	4

Table 4.1: The number of added black subpixels for the pictures in Figure 4.7 with different sizes and contrasts.

reduce the contrast between the background and the share images to be fewer than ϵ. The larger the size and contrast of the image are, the more black subpixels we need to add to the fake shares. Most EVCS's don't have a large contrast, we can easily cheat them by adding a small number of black subpixels to the pixels of the share images in the fake shares.

Example 4.2.3. *Figure 4.8 shows the results of cheating a* (Γ, m)*-EVCS, where* $P = \{1, 2, 3\}$*, and* $Q = \{(1,2), (2,3), (1,2,3)\}$*. In this example,* P_1 *is the cheater who constructs a fake share* FS_2 *with share image* **B** *in substitute for* P_2 *to cheat* P_3*.* $S_1 + FS_2 + S_3$ *reveals the fake image* **FI***.*

Theorem 4.2.3. *The* **MP** *in* **CA-3** *successfully cheats an EVCS by producing fake shares with meaningful share images if the* ϵ *is correct.*

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

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

Indistinguishability. The proof is the same as that of Theorem 4.2.1 except that we have to show that honest participants cannot identify fake shares. Since share images are used for identification, honest participants will not know the exact shapes of share images. They care only about the content of share images. Therefore, the

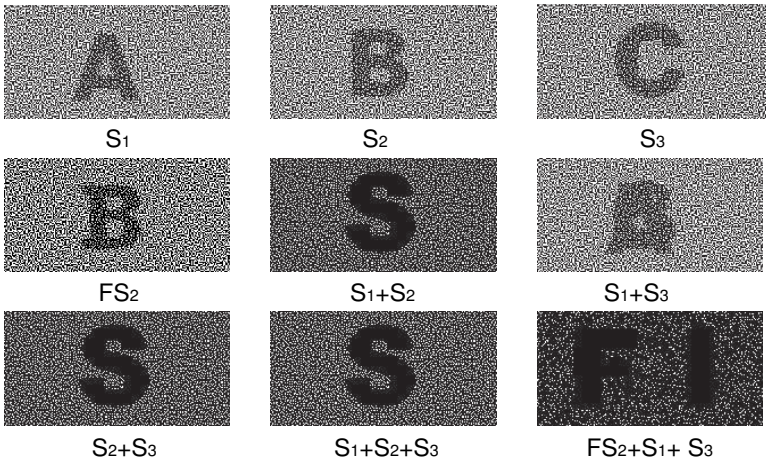


Figure 4.8: An example of cheating a (Γ, m)-EVCS.

cheater who is a legitimate participant can create reasonable share images on fake shares according to his own share to cheat other participants. □

4.3 Attacks and Improvement on Previous Cheat-Preventing Methods

There are two types of cheat-preventing methods [30]. The first type is to have a Trusted Authority (TA) to verify the shares of participants. The second type is to have each participant to verify the shares of other participants. In this section we present attacks and improvement on four existent cheat-preventing methods.

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

The first cheat-preventing method of Yang and Laih [30] needs a TA to hold the special verification share for detecting fake shares. It generates $n+1$ shares $VS, S_1, S_2, \ldots, S_n$, where VS is the verification share. If $VS + S_i$ shows the verification image that is known to all participants, the share S_i is genuine. Let S^0 and S^1 be the basis matrices of a (Γ, m)-VCS. They assign pixels to shares by four sets $C^{0,0}, C^{0,1}, C^{1,0}, C^{1,1}$ which are the sets of all $(n+1) \times (m+2)$-matrices obtained by permuting the columns of

$$S^{00} = \left[\begin{array}{c|ccc} 10 & 0 & \ldots & 0 \\ \hline 10 & & & \\ \vdots & & S^0 & \\ 10 & & & \end{array}\right], S^{01} = \left[\begin{array}{c|ccc} 10 & 0 & \ldots & 0 \\ \hline 10 & & & \\ \vdots & & S^1 & \\ 10 & & & \end{array}\right],$$

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

1. For a white pixel in the fake image, assign the corresponding pixel of S_1 to FS.

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

3. For the rest of unassigned subpixels in FS, copy those from S_1.

Share verification phase: P_1 and P_2 submit S_1 and FS to TA. TA checks the validity of S_1 and FS.
Image reconstruction phase (the fake image): For $Y = \{1, 2, i_1, i_2, \ldots, i_q) \notin Q$, $S_1 + FS + S_{i_1} + S_{i_2} + \cdots + S_{i_q}$ reveals the fake image.

Figure 4.9: Cheat against Yang and Laih's cheat-preventing method.

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

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

Our attack, depicted in Figure 4.9, involves two malicious participants. Without loss of generality, we assume that they are P_1 and P_2. P_1 and P_2 together constructs a fake share FS such that $FS + VS$ reveals the verification image and FS cheats other participants.

We see how the attack works.

1. $FS + VS$ reveals the verification image. The reason is that the first two

subpixels (before permutation) of FS and S_1 are the same. The first two subpixels of $FS+VS$ are the same as those of S_1+VS. Thus, the verification image appears on $FS+VS$. The details are as follows.

For the white pixel of the verification image, the first two pairs of subpixels in S_1 and S_2 are $(1,1)$ and $(0,0)$ by S^{00} and S^{01}, the corresponding subpixels in FS are the same as those in S_1 by Step 2 in the fake share construction phase. Thus, the pixel of $FS+VS$ is white since S_1+VS shows whiteness in the pixel. For the black pixel of the verification image, the first two pairs of subpixels in S_1 and S_2 are $(0,0)$ and $(1,1)$ by S^{10} and S^{11}, the corresponding subpixels in FS are the same as those in S_1. Thus, the pixel of $FS+VS$ is black since S_1+VS shows blackness in the pixel.

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

$$[\cdots 1\cdots 0\cdots]+[\cdots 0\cdots 0\cdots]=[\cdots 1\cdots 0\cdots],$$

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

$$[\cdots 1\cdots 0\cdots]+[\cdots 0\cdots 1\cdots]=[\cdots 1\cdots 1\cdots],$$

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

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

Example 4.3.1. *Figure 4.10 shows the results of cheating a* $(3,3)$*-VCS of Yang and Laih. We see that all shares including the fake share* FS *pass verification by revealing the correct verification image* $\boldsymbol{V}$*. Since* S_1+FS+S_3 *reveals a fake image* ***FI****,* P_3 *is cheated.*

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

In the first cheat-preventing method of Horng et al. [12], each P_i has a verification share V_i. The shares S_i's are generated as usual. Each V_i is divided into $n-1$

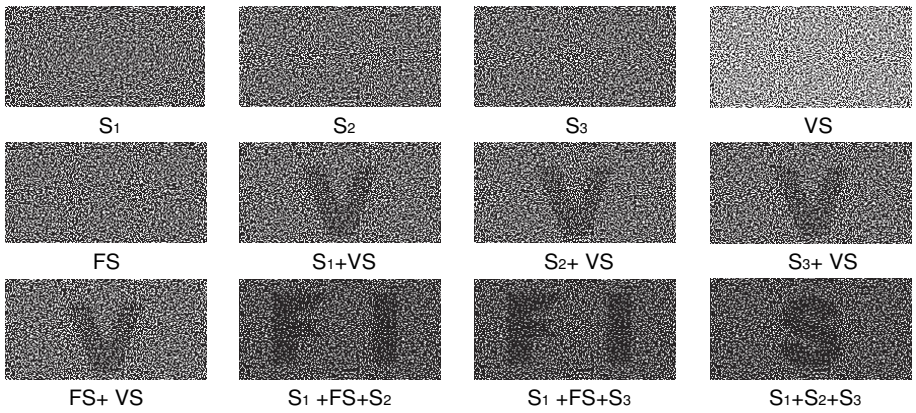


Figure 4.10: An example of cheating the cheat-preventing $(3, 3)$-VCS of Yang and Laih.

regions $R_{i,j}, 1 \leq j \leq n, j \neq i$. Each region $R_{i,j}$ of V_i is designated for verifying share S_j. The region $R_{i,j}$ of $V_i + S_j$ shall reveal the verification image for P_i verifying the share S_j of P_j. The verification image in $R_{i,j}$ is constructed by a $(2, 2)$-VCS. Although the method requires that the verification image be confidential, we show that it is still possible to cheat.

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

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

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

Horng et al.'s second cheat-preventing method uses the approach of redundancy [12]. It uses a $(2, n + l)$-VCS to implement a $(2, n)$-VCS cheat-preventing

scheme. The scheme needs no on-line TA for verifying shares. The scheme generates $n+l$ shares by the $(2, n+l)$-VCS for some integer $l > 0$, but distributes only n shares to the participants. The rest of shares are destroyed. They reason that since the cheater does not know the exact basis matrices even with all shares, the cheater cannot succeed. However, our three cheating methods do not need to use the basis matrices. Any of our cheating methods can cheat this cheat-preventing approach.

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

The second cheat-preventing method of Yang and Laih [30] is a transformation of a (Γ, m)-VCS (but not a $(2, n)$-VCS) to another cheat-preventing $(\Gamma, m + n(n-1))$-VCS. The stacking of any two shares reveals the verification image. This is how share verification is done.

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

$$S^{00} = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 0 & \dots & 1 & 1 & \\ 1 & 0 & 1 & 1 & \dots & 1 & 1 & \\ 1 & 1 & 1 & 0 & \dots & 1 & 1 & S^0 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & \\ \vdots & & \vdots & & \dots & 1 & 0 & \\ 1 & 1 & 1 & 1 & \dots & 1 & 0 & \end{array}\right], \; S^{01} = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 0 & \dots & 1 & 1 & \\ 1 & 0 & 1 & 1 & \dots & 1 & 1 & \\ 1 & 1 & 1 & 0 & \dots & 1 & 1 & S^1 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & \\ \vdots & & \vdots & & \dots & 1 & 0 & \\ 1 & 1 & 1 & 1 & \dots & 1 & 0 & \end{array}\right],$$

$$S^{10} = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 0 & \dots & 1 & 1 & \\ 0 & 1 & 1 & 1 & \dots & 1 & 1 & \\ 1 & 1 & 0 & 1 & \dots & 1 & 1 & S^0 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & \\ \vdots & & \vdots & & \dots & 1 & 0 & \\ 1 & 1 & 1 & 1 & \dots & 0 & 1 & \end{array}\right], \; S^{11} = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 0 & \dots & 1 & 1 & \\ 0 & 1 & 1 & 1 & \dots & 1 & 1 & \\ 1 & 1 & 0 & 1 & \dots & 1 & 1 & S^1 \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & \\ \vdots & & \vdots & & \dots & 1 & 0 & \\ 1 & 1 & 1 & 1 & \dots & 0 & 1 & \end{array}\right].$$

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

By the new definition, what the human eyes care about is contrast, no matter whether the image is darker or lighter than the background. Our improvements are applicable to Yang and Laih's cheat-preventing method. It reduces the pixel expansion to $m + n(n-1)/2$. Moreover, since the verification image can be made public to all participants, we can let the verification image appear on the shares. By this, we can further reduce the pixel expansion to $m + n(n-1)/4$.

Our improvement is based on the following three theorems, which are proved in Chapter 3.

Theorem 4.3.1. (*Composition property*) *Let* $\Gamma_1 = (P, Q_1, F_1)$ *and* $\Gamma_2 = (P, Q_2, F_2)$ *be two access structures. Assume that* $Q_1 \cap Q_2 = \emptyset$. *If there exist a* (Γ_1, m_1)-VCS_2 *and a* (Γ_2, m_2)-VCS_2, *there exist a* $(\Gamma, m_1 + m_2)$-VCS_2, *where* $\Gamma = (P, Q_1 \cup Q_2, F_1 \cap F_2)$. VCS_2 *is a visual cryptography scheme based on the new definition proposed.*

Theorem 4.3.2. (*Deletion property*) *Let* $\Gamma = (P, Q, F)$ *be an access structure. If* S^0 *and* S^1 *are basis matrices for a* (Γ, m)-VCS_2, $S^{0'}$ *and* $S^{1'}$ *are basis matrices for a* $(\Gamma, m-k)$-VCS_2, *where* $S^{0'}$ *and* $S^{1'}$ *are obtained from* S^0 *and* S^1 *by deleting the same* k *columns.*

Theorem 4.3.3. (*Inverse property*) *Let* $\Gamma = (P, Q, F)$ *be an access structure. If* S^0 *and* S^1 *are basis matrices for a* (Γ, m)-VCS_2, $S^{0'}$ *and* $S^{1'}$ *are basis matrices for a* (Γ, m)-VCS_2, *where* $S^{0'} = S^1$ *and* $S^{1'} = S^0$.

We denote the left appended matrices in $S^{b_1 b_2}$ as $n(n-1)/2$ sub-matrices $S_k^{b_1 b_2}$, where $1 \le k \le n(n-1)/2$, $b_1, b_2 \in \{0, 1\}$. Each sub-matrix $S_k^{b_1 b_2}$ consists of two columns counting from left to right. Based on Theorems 4.3.1-4.3.3, we can exchange the roles of S_k^{00} and S_k^{10}, and also S_k^{01} and S_k^{11}, and delete $n(n-1)/2$ common columns. Furthermore, we delete all columns having one "0" only for the case that the verification image may not appear on the shares. By these steps, the pixel expansion of the appended matrices is reduced to $n(n-1)/4$.

Let $P = \{1, 2, 3, 4\}$. The basis matrices for a cheat-preventing (Γ, m)-VCS using Yang and Laih's cheat-preventing method are as follows:

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

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

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

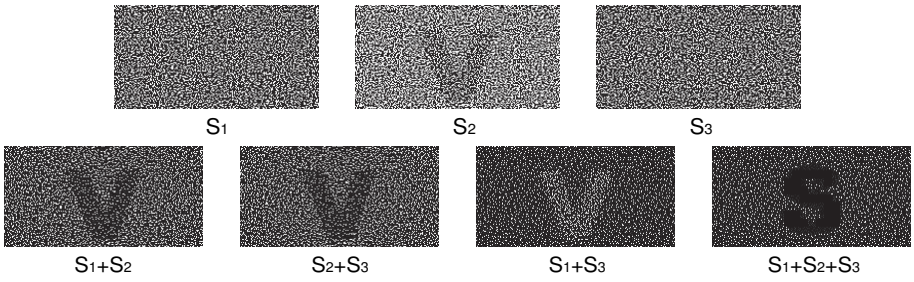


Figure 4.11: An improved $(3,3)$-VCS_2 for Yang and Laih's cheat-preventing method.

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

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

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

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

Example 4.3.2. *Figure 4.11 shows the results of the improved cheat-preventing* $(3,3)$*-VCS*$_2$*. We see that the stacking of any two shares reveals the verification image* $\mathbf{V}$*.* S_1+S_3 *reveals the reversed verification image and* S_2 *shows the verification image.*

4.3.4 A Generic Transformation for Cheating Prevention

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

1. It does not rely on the help of an on-line TA. Since VC emphasizes on easy decryption with human eyes only, we should not have a TA to verify validity of shares.

2. The increase to pixel expansion should be as small as possible.

3. Each participant verifies the shares of other participants. This is somewhat necessary because each participant is a potential cheater.

4. The verification image of each participant is different and confidential. It spreads over the whole region of the share. We have shown that this is necessary for avoiding the described attacks.

5. The contrast of the secret image in the stacking of shares is not reduced significantly in order to keep the quality of VC.

6. A cheat-preventing method should be applicable to any VCS.

We now present a generic transformation from a VCS to another cheat-preventing VCS. The resultant cheat-preventing VCS meets all the above requirements. The idea is similar to the first cheat-preventing method of Yang and Laih [30]. But, we let each participant hold a verification share. Our cheat-preventing scheme needs no help from an on-line TA. The verification image for each participant is different and known to the participant only.

Our transformation is quite efficient and almost optimal as it adds only two subpixels for each pixel of the original image. That is, if the pixel expansion of the VCS is m, the pixel expansion of the transformed VCS is $m+2$. The contrast is slightly reduced from $\alpha(m)$ to $\alpha(m') = (\alpha(m) \times m + 1)/(m+2)$. Our transformation is depicted in Figure 4.12. It generates two shares for each participant. One is the secret share and the other is the verification share. Let S^0 and S^1 be the $n \times m$ basis matrices of a (Γ, m)-VCS. At first, we create two $n \times (m+2)$-dimensional basis matrices T^0 and T^1. The transformed $(\Gamma, m+2)$-VCS uses T^0 and T^1 as the basis matrices to generate shares for the participants as usual. Then, for each P_i, it generates a verification share V_i for a chosen verification image. For each white pixel in the verification image, it puts the pixel of $(m+2)$-dimensional $[1\ 0\ 0\ \cdots\ 0]$ to V_i (after corresponding permutation as for the share S_i). For each black pixel in the verification image, it puts the pixel of $(m+2)$-dimensional $[0\ 1\ 0\ 0\ \cdots\ 0]$ to V_i (after corresponding permutation as for the share S_i). We see that the verification image is encoded into the first two subpixels. If P_i wants to verify the share S_j of P_j, he checks whether $V_i + S_j$ shows his verification image.

Example 4.3.3. *Figure 4.13 shows a transformed* $(\Gamma, m+2)$*-VCS with cheating prevention, where* $P = \{1, 2, 3\}$ *and* $Q = \{(1, 2), (2, 3), (1, 2, 3)\}$. *The verification*

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

1. Let

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

2. Use T^0 and T^1 as the basis matrices for generating shares S_i, $1 \leq i \leq n$, of $(\Gamma, m+2)$-VCS.

3. For each P_i, $1 \leq i \leq n$, choose a verification image and generate a verification share V_i as follows:

   (a) For each white pixel in the verification image, put the pixel of $(m+2)$-dimensional $[1\ 0\ 0\ \cdots\ 0]$ (subpixels) to V_i (after corresponding permutation as for the share S_i).

   (b) For each black pixel in the verification image, put the pixel of $(m+2)$-dimensional $[0\ 1\ 0\ 0\ \cdots\ 0]$ (subpixels) to V_i (after corresponding permutation as for the share S_i).

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

Figure 4.12: Our generic transformation for VCS with cheating prevention.

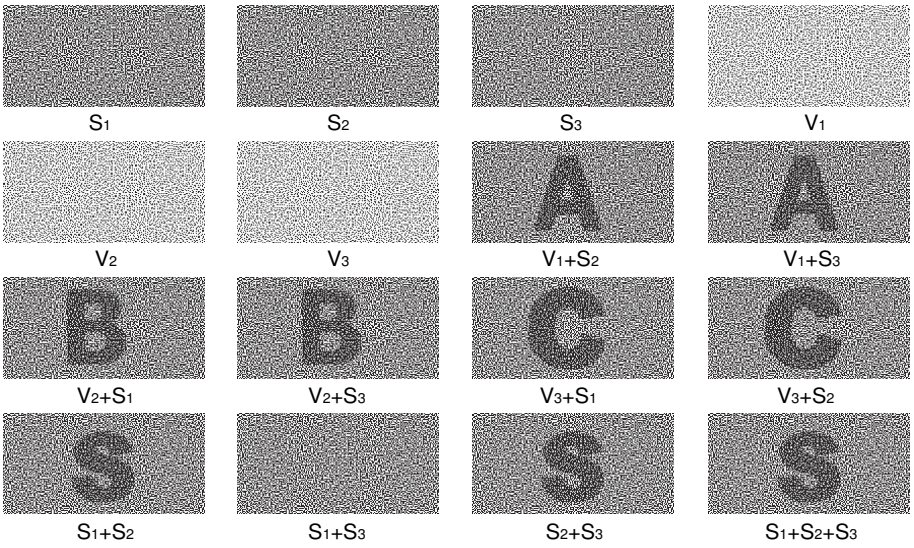


Figure 4.13: An example of a transformed VCS with cheating prevention.

images for P_1, P_2 and P_3 are **A**, **B** *and* **C**, *respectively. Note that the simple verification images are for demonstration only. By our proposed principle in Section 4.3.2, we should use more complicated verification images.*

Theorem 4.3.4. *The algorithm in Figure 4.12 transforms any (Γ, m)-VCS to another (Γ, m')-VCS with cheating prevention, where $m' = m+2$ and $\alpha(m') = (\alpha(m) \times m+1)/m'$.*

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

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

Each participant has his own private verification image, which is not known to other participants. Since the first two subpixels $[1\ \ 0]$ (before permutation) of all shares are the same, a P_i even with all shares cannot know the positions of black pixels of the verification image of P_j, $j \neq i$. Therefore, P_i cannot produce a fake

share FS_i such that $FS_i + V_j$ shows the verification image of P_j. P_i cannot cheat P_j for $i \neq j$. Furthermore, we see that collaboration of some participants cannot succeed to cheat, either. □

Chapter 5

Improvements on VCSs with Reversing

In this chapter we propose three new ideal contrast VCS with reversing which is compatible and requires fewer stacking and reversing operations, compared to all previous schemes. One is based on VCS_2, the others is based on VCS_1. Each participant is required to store only two transparencies and obtain the ideal reconstruction image in only two runs.

5.1 Brief Review of Previous VCSs with Reversing

In this section, we review three existing VCSs with reversing. The first scheme is proposed by Viet and Kurosawa [27]. Their scheme generates c shares (for some c) for each participant by performing the original VCS c times independently.

Suppose that there exists a (k, n)-VCS of perfect reconstruction of black pixels. The "c-runs (k, n)-VCS with reversing of Viet and Kurosawa" is constructed as follows.

1. Let $(S_{1,i}, S_{2,i}, ..., S_{n,i})$ be the set of shares in the i-th run for $i = 1, ..., c$.
2. The transparencies of participants i_j are $S_{j,1}, S_{j,2}, ..., S_{j,c}$ for $j = 1, ..., n$.
3. Any k participants in Q reconstruct the secret image by:
   - superimposing their transparencies and obtain $T_i = S_{j_1,i} + S_{j_2,i} + ... + S_{j_k,i}$, where $i = 1, ..., c$.
   - computing $U = (T_1' + ... + T_c')'$
   - U, which is the reconstructed secret image.

We can see that a series of Boolean operations performed in this scheme is exactly equal to $c-1$ AND operations performed on the transparencies $T_1, ..., T_c$.

The First Scheme of S. Cimato et al. [8] encodes the secret image pixel by pixel. Each pixel is considered independently on the others. Their construction requires each participant to store m transparencies, each size are same as the original image. The scheme is constructed as follows.

1. Let (S^0, S^1) be the basis matrices constituting a VCS of perfect reconstruction of black pixels.

2. The dealer randomly chooses a matrix $C^0 = [c_{i,j}]$ from S^0 (C^1 from S^1, resp.).

3. For each participant i, consider the m bits $c_{i,1}, c_{i,2}, ..., c_{i,m}$ composing the i-th row of C^0 and C^1, for each $j = 1, ..., m$, put a white (black, resp.) pixel on the transparency $S_{i,j}$ if $c_{i,j} = 0$ ($c_{i,j} = 1$, resp.).

4. Any k participants in Q reconstruct the secret image by computing:

   - $T_j = OR(S_{i_1,j}, ..., S_{i_k,j})$, for $j = 1, ..., m$.
   - $U = (OR(T'_1 + ... + T'_m))'$
   - U, which is the reconstructed secret image.

The Second Scheme of S. Cimato et al. [8] reduce the number of transparencies by using as a building block a binary secret sharing scheme (BSS). A BSS consists of two collections $\mathcal{B}^0$ and $\mathcal{B}^1$ of distribution functions. A distribution function $f \in \mathcal{B}^0 \cup \mathcal{B}^1$ is a function associating each participant i to the share $f(i)$. The scheme is constructed as follows.

1. The dealer randomly chooses a distribution function $f \in \mathcal{B}^0$ ($f \in \beta_1$, resp.), where $\mathcal{B}^0$ and $\mathcal{B}^1$ are the collections of distribution functions realizing a BSS [7] for (P, Q, F).

2. For each participant i, consider the binary representation $c_{i,1}, ..., c_{i,r}$ of share $f(i)$ and, for each $j = 1, ..., r$, where r is the size of the shares distributed by the BSS [7], put a white (black, resp.) pixel on the transparency $S_{i,j}$ if $c_{i,j} = 0$ ($c_{i,j} = 1$, resp.).

3. Any k participants in Q reconstruct the secret image by the sequence of reversing and stacking operations on their transparencies in parallel $Rec(f(i_1), ..., f(i_k))$, and $Rec()$ is a reconstruction algorithm which on inputs the shares and outputs the secret.

S_i	S_j	S_i AND S_j
0	0	0
0	1	0
1	0	0
1	1	1

Table 5.1: The truth table of S_i AND S_j.

5.2 A Compatible Ideal Contrast $(2, 2)$-VCS with Reversing in Two Runs

The basic idea of Viet and Kurosawa's scheme is to perform AND operations on two transparencies. Performing an AND operation on two pixels reveals a black pixel only while two pixels are both black (see the truth table of the AND operation in Table 1). Because the reconstructed secret image in Viet and Kurosawa's VCS is of perfect reconstruction of black pixels, the black pixels will stay black no matter how many AND operations are performed. Viet and Kurosawa's scheme performs AND operations as many times as possible on the stacked transparencies generated by a perfect black VCS. As a result, the secret images (black pixels) stay black and the background (white pixels) will increasingly become whiter.

We show how to construct an ideal contrast $(2, 2)$-VCS in two runs by computing OR and AND operations only in Figure 5.1. Compared to the scheme of Viet and Kurosawa, ours chooses the complement transparencies S_i', $i \in \{1, 2\}$, to be the shares of the second run while theirs chooses other transparencies randomly. Our scheme achieves ideal contrast in two runs and requires each participant to store only one transparency. With same stacking operations we achieve ideal contrast GREY(white)=0 while their scheme achieves GREY(white)=$\frac{1}{4}$ in addition to GREY(black)=1.

Figure 5.2 shows an example of comparing the results Viet and Kurosawa's scheme and ours. We see that the reconstructed image of our scheme has better contrast.

Theorem 5.2.1. *The scheme in Figure 5.1 is a two runs ideal contrast* $(2, 2)$*-VCS with reversing.*

Proof. Step 3 in the reconstruction phase computes an AND operation on T and T', i.e. $((T)' + (T')')'$ is equal to T AND T'. Suppose that a pixel P is black (the secret image). Then the pixel P on T and T' is always black since Naor-Shamir $(2, 2)$-

Input.

1. A $(2, 2)$ access structure (P, Q, F).
2. Let C^0 and C^1 be the set of all matrices obtained by permuting columns of S^0 and S^1 matrices constituting a Naor-Shamir $(2, 2)$-VCS in all possible ways.

Distribution phase.

1. Let (S_1, S_2) denote the transparencies generated by the basis matrices S^0 and S^1 for participants 1 and 2 .

Reconstruction phase.

1. Two participants $1, 2$ obtain S_1' and S_2' by reversing S_1 and S_2 respectively.
2. They superimpose S_1, S_2 and S_1', S_2' separately, and obtain T and T', where $T = S_1 + S_2$ and $T' = S_1' + S_2'$.
3. Two participants reverse T, T' separately, superimpose them together and obtain $U = ((T)' + (T')')'$.

Output. U, which is the reconstructed secret image.

Figure 5.1: A construction for ideal contrast $(2, 2)$-VCS with reversing.

VCS and the reverse of Naor-Shamir $(2, 2)$-VCS are all perfect black reconstruction, namely GREY(black)= 1.

On the other hand, suppose that P is a white pixel (the background). Then the color of P corresponding to T and T' is exactly opposite to each other, and the return pixel on U is always white. So, this scheme reveals an ideal contrast image U, where GREY(white)=0 in addition to GREY(black)=1. □

Same as in Viet and Kurosawa's scheme, the bit operation of AND is used in this scheme. We conclude that a compatible VCS with reversing can obtain ideal contrast by computing an AND operation in two runs, only if the following requirements are satisfied.

1. The VCS should be perfect black reconstruction, since the black pixels should remain black after computing an AND operation.
2. The GREY(white)$\geq \frac{1}{2}$, since the white pixels should become GREY(white)=0 in two runs.

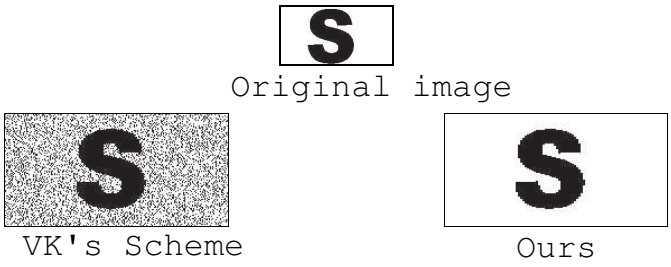


Figure 5.2: The images reconstructed in two runs by Viet and Kurosawa's scheme and ours.

3. The columns of elements of basis matrix S^0 should be either all 0's or 1's, since the white pixels in the reconstructed transparencies T and T' should be exactly opposite.

By the above requirements, our construction in this section is only applicable to $(2, 2)$-VCS. In the next section, we propose two construction for VCS with reversing with general access structures.

5.3 Two Constructions for Compatible Ideal Contrast VCSs with Reversing

In this section we describe two constructions of compatible VCSs with reversing which reveal an ideal contrast image for any access structure in only two runs. The first construction is based on the traditional definition of VC [20]. The second one is based on new definition we proposed.

5.3.1 An Ideal VCS with Reversing for General Access Structure

Before introducing our approach, we describe a VCS for general minimal access structure Γ_0, which was proposed in [1] and [9], that will be used in our approach. Then we show how to construct another basis matrix to generate Auxiliary Transparencies (AT) for each participant. These ATs are generated for our VCS with reversing to reconstruct an ideal contrast secret image.

5.3.2 A VCS for Minimal Access Structure Γ_0

This VCS employs Naor-Shamir (k, k)-VCS as a basis unit for constructing a VCS for minimal access structure Γ_0. Suppose $\Gamma_0 = \{Q_1, ..., Q_b\}$, by employing the optimal

(k,k)-VCS, the basis matrices L^0 and L^1 are constructed as follows.

Suppose that $Q_r = \{i_1, ..., i_{k_p}\}$ and $k_p = |Q_r|$. For $1 \leq r \leq b$, construct an $n \times 2^{k_p-1}$ matrix E_r^i, $i \in \{0,1\}$, with the following steps:

The p_i row of E_r^0 is the i-th row of the basis matrix S^0 of the optimal (k_p, k_p)-VCS. The elements of other rows of E_r^0 are all 1's. Then $L^0 = E_1^0 \parallel ... \parallel E_b^0$. The construction of E_r^1 is similar to E_r^0 except that we replace the p_i row of E_r^1 from the basis matrix S^1 of the optimal (k_p, k_p)-VCS instead of S^0. Then $L^1 = E_1^1 \parallel ... \parallel E_b^1$.

Lemma 5.3.1. *L^0 and L^1 are a pair of basis matrices of a perfect black VCS for Γ_0 such that the pixel expansion $m = 2^{|Q_1|-1} + ... + 2^{|Q_b|-1}$ and GREY(white) $= 1 - \frac{1}{m}$ [27].*

For $1 \leq r \leq b$, an $n \times 2^{k_p-1}$ matrix F_r is constructed as follows. The elements in p_i row of F_r are all 0's. The other rows of F_r are all 1's. Then an auxiliary basis matrix $A^0 = F_1|...|F_b$. In other words, A^0 is the same matrix as L^0 except that we replace all the elements of the (k_p, k_p)-VCS with "0". We regard all the pixels on transparencies constituted by A^0 as white pixels. Therefore, we only need a basis matrix to generate the transparencies.

For example, for $\Gamma_0 = \{(1,2),(2,3,4)\}$ and $P = \{1,2,3,4\}$, then

$$L^0 = E_1^0 \parallel E_2^0 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}, \quad L^1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix},$$

$$A^0 = F_1 \parallel F_2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

5.3.3 Our Construction

The construction is depicted in Figure 5.3. We encode the secret image into n transparencies. Instead of only encoding one secret image into n shares, we divide every share (transparency) into $|\Gamma_0|$ blocks and every block has one secret image. It implies that there are $|\Gamma_0|$ secret images in the reconstructed transparency and that each secret image can be reconstructed by one qualified set.

Lemma 5.3.2. *The optimal (k,k)-VCS proposed by Naor and Shamir [20] is a compatible ideal contrast (k,k)-VCS with reversing.*

Input.

1. A minimal access structure Γ_0 on a set P of n participants.
2. Let C_r^0, C_r^1, and C'^0_r be the collection of basis matrices E_r^0, E_r^1 and F_r resp., where $1 \leq r \leq |\Gamma_0|$ and .

Distribution phase.
The dealer encodes each transparency t_i as $|\Gamma_0|$ sub-transparencies $S_{i,r}$ and each sub-block consists of one secret image. For $1 \leq r \leq |\Gamma_0|$, each white (black pixel) on sub-block of transparency $S_{i,r}$ is encoded using a $n \times 2^{k_p-1}$ matrice E_r^0 (E_r^1 resp.). To share a white (black, resp.) pixel, the dealer,

1. randomly chooses a matrix $S_r^0 = [s_{i,j}]$ in C_r^0 (S_r^1 in C_r^1 resp.), and a matrix $A_r^0 = [a_{i,j}]$ in C'^0_r.
2. For each participant i, put a white (black, resp.) pixel on the sub-block of transparency $S_{i,r}$ if $s_{i,j} = 0$ ($s_{i,j} = 1$, resp.).
3. For each participant i, put a white (black, resp.) pixel on the sub-block of transparency $AS_{i,r}$ if $a_{i,j} = 0$ ($a_{i,j} = 1$, resp.).

Reconstruction phase.
Participants in Q_p reconstruct the secret image by,

1. XORing all the shares S_i and stacking all the transparencies AS_i for $i = 1, ..., k_p$ and obtain T and A respectively.
2. computing $U = (T + A) \oplus A$.

Output. U, which is the reconstructed secret image.

Figure 5.3: A construction for ideal contrast VCS with reversing.

Proof. We show that *Naor and Shamir's* (k, k)-VCS with reversing is compatible and ideal contrast by the following:

Compatible. This VCS has been proven optimal since in any (k, k)-VCS, the pixel expansion has to be at least 2^{k-1} and contrast can be at most $\frac{1}{2^{k-1}}$ [20].

Ideal contrast. Naor and Shamir's (k, k)*-VCS* is obtained by means of the construction of the basis matrices S^0, S^1. S^0 is the matrix whose columns are all the Boolean k-vectors having an even number of 1's; whereas, S^1 is the matrix whose columns are all the Boolean k-vectors having an odd number of 1's. In order to obtain the ideal contrast secret image, k participants compute $S_1 \oplus S_2 \oplus ... \oplus S_k$ on the k transparencies. It is easy to see that the white pixels are all white since S^0

has an even number of 1's; whereas the black pixels are all black since S^1 has an odd number of 1's. □

Theorem 5.3.1. *Let $\Gamma = (P, Q, F)$ be an access structure on a set P of n participants. The basis matrices S^0, S^1 and A^0 constitute a compatible ideal contrast VCS with reversing in two runs.*

Proof. Compatible. The basis matrices S^0 and S^1 have been proven constituting a VCS in [27], i.e. the secret image can be reconstructed by directly superimposing the transparencies of any qualified set. As for the property of security, it is obvious that a VCS is as secure as a VCS with reversing [1]. The basis matrix A^0 reveals no information about the secret image since no secret is encoded into the shares At_i.

Ideal contrast. Let $L^0 = E_1^0 \parallel ... \parallel E_b^0$, $L^1 = E_1^1 \parallel ... \parallel E_b^1$ and $A^0 = F_1 \parallel ... \parallel F_b$ be the basis matrices for a VCS with reversing, constructed using the previously described technique. Without loss of generality, let $\Gamma_0 = \{Q_1, ..., Q_b\}$ and $X = Q_1$, X be a subset of qualified participants. Since the secret image is reconstructed by computing $(T + A) \oplus A$, we have to show that L^0, L^1 and A^0 are the basis matrices of a VCS with reversing for the general access structure $\Gamma = (P, Q, F)$ having ideal contrast, i.e. $w((E_1^0 + F_1) \oplus F_1) = 0$, $w((E_1^1 + F_1) \oplus F_1) = 2^{|Q_1|-1}$ and $w((E_r^i + F_r) \oplus F_r) = 0$, $r \in \{2, \ldots, |\Gamma_0|\}$ and $i \in \{0, 1\}$. It results that

$$\begin{aligned} & w((E_1^0 + F_1) \oplus F_1) \\ = \; & w((E_1^0 + 0) \oplus 0) \\ = \; & w(E_1^0 \oplus 0) \\ = \; & w(E_1^0) = 0 \qquad \text{(according to Lemma 5.3.2)} \end{aligned}$$

and

$$\begin{aligned} & w((E_1^1 + F_1) \oplus F_1) \\ = \; & w((E_1^1 + 0) \oplus 0) \\ = \; & w(E_1^1 \oplus 0) \\ = \; & w(E_1^1) = 2^{|Q_1|-1} \qquad \text{(according to Lemma 5.3.2)} \end{aligned}$$

whereas,

$$\begin{aligned} & w((E_r^i + F_r) \oplus F_r) \quad \text{for} \quad r \in \{2, \ldots, |\Gamma_0|\} \text{ and } i \in \{0, 1\} \\ = \; & w((E_r^i + 1) \oplus 1) \\ = \; & w(1 \oplus 1) \\ = \; & 0 \end{aligned}$$

□

Example 5.3.1. *Let $P = \{1, 2, 3, 4\}$ and $\Gamma_0 = \{(1, 2), (2, 3, 4)\}$. Then the basis matrices L^0, L^1 and A^0 are constructed as follows.*

T	A	$T+A$	$(T+A)\oplus A$
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	0

Table 5.2: The truth table of $(T+A)\oplus A$.

$$L_0=\begin{bmatrix}1&0&1&1&1&1\\1&0&0&1&1&0\\1&1&0&1&0&1\\1&1&0&0&1&1\end{bmatrix}\ L_1=\begin{bmatrix}1&0&1&1&1&1\\0&1&1&1&0&0\\1&1&1&0&1&0\\1&1&1&0&0&1\end{bmatrix}\ A_0=\begin{bmatrix}0&0&1&1&1&1\\0&0&0&0&0&0\\1&1&0&0&0&0\\1&1&0&0&0&0\end{bmatrix}$$

There are two secret images encoded into four shares, one is in block 1 for $\{1,2\}$ and the other is in block 2 for $\{2,3,4\}$. Let $Q_2=\{2,3,4\}$, then $T=$ XOR (XOR $(S_2,S_3),S_4$) and $A=$ OR (OR $(S_2,S_3),S_4$). From the truth table of $(T+A)\oplus A$ in Table 5.2, we see that the outcome of $U=(T+A)\oplus A$ is 1 only while $T=1$ and $A=0$. Therefore, all the black pixels will be reconstructed as the perfect black pixels.

Example 5.3.2. *The results of the construction for* $\Gamma_0=\{(1,4),(2,3,4)\}$*, where* $P=\{1,2,3,4\}$*, are depicted in Figure 5.4. Shares* At_1 *and* At_4 *are omitted since they are the transparencies with all white subpixels.*

5.3.4 A Compatible Ideal Contrast VCS_2 with Reversing for General Access Structure

As we mentioned before, what the human eye cares about is contrast, no matter whether the image is *darker* or *lighter* than the background. In this section, we show a construction in Figure 5.5 for VCS_2. It still recover the secret image with ideal contrast, and remains compatible.

Theorem 5.3.2. *The basis matrices* S'^0*,* S'^1 *and* A^0 *in Figure 5.5 constitute a compatible ideal contrast* VCS_2 *with reversing in two runs.*

Proof. Compatible. The basis matrices S'^0 and S'^1 have been proven constituting a VCS_2, in which the recovered image is either darker or lighter than the background. As for the property of security, no information about the secret image will be revealed since the basis matrix A^0 is unchanged.

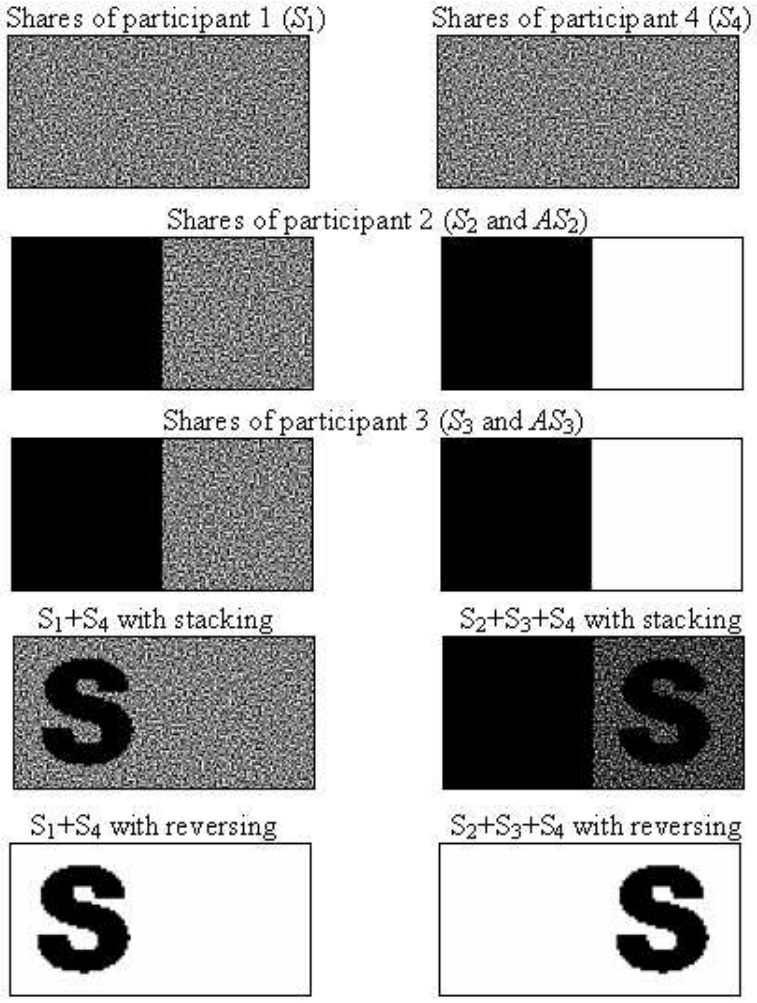


Figure 5.4: The results of construction one for VCS_1.

Ideal contrast. If the recovered secret image is darker than the background, then the proof is the same as that in Theorem 5.3.1. Suppose that the recovered secret image is reversed (the secret image is lighter than the background). Wlog, let $\Gamma_0 = \{Q_1, ..., Q_b\}$ and $X = Q_1$, X be a subset of qualified set. In order to prove the contrast of the reversed secret image is ideal, we have to show that L^0, L^1 and A^0 are the basis matrices of a VCS with reversing for $\Gamma = (P, Q, F)$, having ideal contrast, i.e. $w(E_1^0 + F_1) = m,\ w(E_1^1 + F_1)) = 0$ and $w(E_r^i + F_r) = m$ for $r \in \{2, \ldots, |\Gamma_0|\}$

Input.

1. A minimal access structure Γ_0 on a set P of n participants.

Distribution phase.

1. The dealer uses the approaches [a] to generate S'^0, S'^1 according to S^0, S^1, where the constructions of the basis matrices S^0, S^1 and A^0 remain the same as in Figure 5.3.

2. The transparencies S_i and AS_i, $i = 1, ..., n$, are constructed as in Figure 5.3 except that the basis matrices S^0, S^1 for S_i are replaced with S'^0, S'^1.

Reconstruction phase.
Let $Q_r = \{i_1, ..., i_{k_p}\}$ be the minimal qualified set in Γ_0, participants in Q_r reconstruct the secret image by,

1. XORing all the shares S_i and stacking all the shares AS_i for $i = 1, ..., k_p$ and obtain T and A respectively.

2. Computing $U = (T + A) \oplus A$, if the recovered image is darker than the background else $U = T + A$.

Output. The transparency U, which is the secret image with deal contrast.

[a] The relative approaches and proofs can be found in Chapter 3

Figure 5.5: A construction for ideal contrast VCS_2 with reversing.

.

and $i \in \{0, 1\}$, where $m = 2^{|Q_1|-1}$. It results that:

$$\begin{aligned} & w(E_1^0 + F_1) \\ &= w(E_1^0 + 0) \\ &= w(E_1^0) = m \end{aligned}$$

and,

$$\begin{aligned} & w(E_1^1 + F_1) \\ &= w(E_1^1 + 0) \\ &= w(E_1^1) = 0 \end{aligned}$$

whereas,

$$\begin{aligned} & w(E_r^i + F_i) \quad \text{for} \quad r \in \{2, \ldots, |\Gamma_0|\} \quad \textit{and} \quad i \in \{0, 1\} \\ = \; & w(E_r^i + 1) \\ = \; & m \end{aligned}$$

□

Example 5.3.3. *Let* $P = \{1, 2, 3\}$ *and* $\Gamma_0 = \{(1, 2), (2, 3)\}$. *We depict the results of the images reconstructed by* VCS_2 *with reversing in Figure 5.6.*

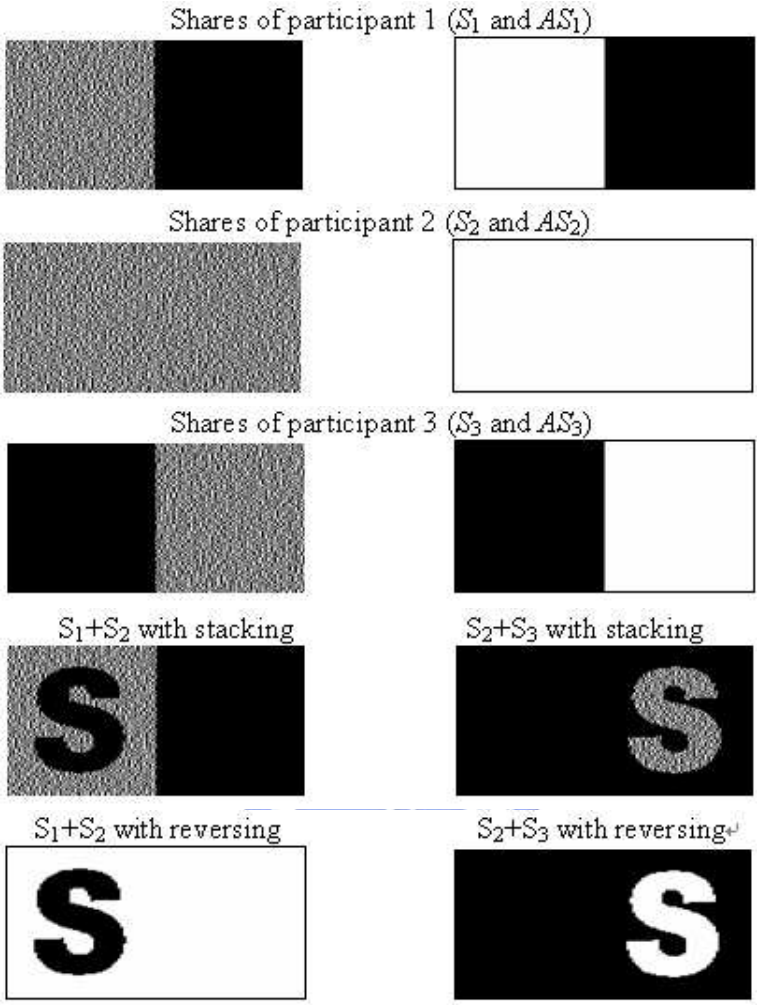


Figure 5.6: The results of construction two for VCS_2.

5.4 Discussions

5.4.1 Reducing Pixel Expansion And Improving Contrast

Every share in our schemes is divided into $b = |\Gamma_0|$ blocks. It implies that the pixel expansion is reduced by b times compared with Viet and Kurosawa's scheme. As a result, the contrast of the recovered secret images will also improve b times compared to Viet and Kurosawa's scheme while revealing the secret image only with human visual system.

Ito et al. [13] proposed a size invariant VCS to encode the secret image into the same size shares as the secret image, and the reconstructed image of the proposed scheme has the same contrast as in the conventional scheme. Compared to traditional VCSs, the contrast of their VCS is defined as $|p_0 - p_1|$ where p_0 and p_1 are the appearance probabilities of a black pixel on the background and the secret of the reconstructed image respectively [13]. In other words, contrast is increased when the probability of a black pixel appearing on the secret image becomes bigger, or the probability of a black pixel on the background of the reconstructed image becomes smaller.

Our VCSs with reversing can be applied to this method on each sub-block. It reduces the number of pixel expansion of our VCSs with reversing to b.

5.4.2 A Comparison of Properties Among the VCSs with Reversing in [27], [8] And Ours

Table 5.3 shows a comparison of properties between our scheme and previous VCSs with reversing. We measure the efficiency of VCS with reversing by the following seven factors:

- Compatibility
- Contrast of the reconstructed secret image with reversing
- Contrast of the reconstructed secret image with only stacking (to recover the secret image without using a copy machine)
- Number of stacking operations
- Number of reversing operations
- Shares held by each participant
- Pixel expansion

Some variables used in Table 5.3 are denoted as follows.

- k : the number of participants in the qualified set Q_i.
- c : the number of AND operations performed in Viet and Kurosawa's scheme.
- b : the number of the blocks divided in each transparency.

Properties	Viet and Kurosawa's	Cimato et al.'s (1)	Cimato et al.'s (2)	Ours
Compatible	$\surd$	$\times$	$\times$	$\surd$
Contrast with reversing	Almost ideal	Ideal	Ideal	Ideal
Contrast with only stacking	$\frac{1}{m}$	0	0	$\frac{b}{m}$
Number of stacking operations	ck	$k(m+1)$	$3r(k-1)$	$4k-1$
Number of reversing operations	$3(c-1)$	$m+1$	$4r(k-1)$	$4k$
Shares held by each participant	c	m	r	2
Pixel expansion	m	1	1	$\frac{m}{b}$

Table 5.3: A comparison of properties among the previous VCSs with reversing and ours.

- m : the pixel expansion of a VCS described in Section 5.3.1.
- r : the number of bits in the binary representation of the largest share.

Obviously, we hope that the scheme is compatible so that the secret image can still be obtained even when there is no available copy machine. It will be better to achieve ideal contrast in finite steps. Finally, we hope to minimize the numbers in the various factors. As we can see in Table 5.3, our scheme achieves both compatibility and ideal contrast. Compared to Viet and Kurosawa's scheme, our scheme is better on every property. To the first and second schemes of Cimato et al, we also have better properties except for pixel expansion. The pixel expansion m in both schemes of Viet and Kurosawa and ours is necessary in order to gain the property of compatibility.

Chapter 6

Improvements on Extended VCSs

In this chapter by the new definition, we show that EVCSs may have better contrast than those based on the conventional definition. We propose a (2, n)-EVCS scheme based on the new definition. Although the image of this construction is not "smooth", it has better contrast than previous results.

6.1 Optimal Contrast (k, k) Threshold EVCS

Theorem 6.1.1. *[2] In any (k, k)-threshold EVCS with pixel expansion m the relative differences $\alpha_F(m)$ and $\alpha_S(m)$ satisfy*

$$2^{k-1}\alpha_F(m) + \frac{k}{k-1}\alpha_S(m) \leq 1.$$

From the theorem above, we can calculate that a (2, 2) threshold EVCS cannot have a better contrast of more than 1/4. But, based on our new definition, we can improve the contrast to 1/3. Note that black and white pixel respectively in the conventional definition is represented in Figure 6.1. The pixel expansion is four.

Black and white pixel respectively in the new definition is represented in Figure 6.2. The pixel expansion is three, which is better than that based on the original definition.

Figure 6.1: Black and white pixel respectively in the conventional definition.

Figure 6.2: Black and white pixel respectively in the new definition.

6.2 (2, n)-EVCS Based on New Definition

In this section, we propose a construction that can solve a 2 out of n EVCS problems based on the new definition.

Example 6.2.1 shows a (2, 2) EVC construction based on the new definition.

Example 6.2.1. *Basis matrices of a (2, 2) EVCS base on the new definition*

$$S_w^{ww} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad and \quad S_b^{ww} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S_w^{wb} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad and \quad S_b^{wb} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$S_w^{bw} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad and \quad S_b^{bw} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$S_w^{bb} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad and \quad S_b^{bb} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Theorem 6.2.1. *The scheme in Figure 6.3 is a 2 out of n EVCS with parameters* $m = n + 1$, $\alpha_s(m) = \alpha_F(m) = 1/n + 1$.

Proof. Let $\{(C_w^{C_1 \ldots C_n}, C_b^{C_1 \ldots C_n})\}_{C_1, \ldots C_n}$ be a family of 2^n pairs of collections constituting a $(2, n)$-threshold EVCS. Without loss of generality, we assume that for any choices of $c_1, ..., c_n \in \{b, w\}$, the pair of collections $(C_w^{C_1 \ldots C_n}, C_b^{C_1 \ldots C_n})$ are obtained by permuting, in all possible ways, the columns of the pair of basis matrices $(S_w^{C_1 \ldots C_n}, S_b^{C_1 \ldots C_n})$. □

$Case1$: The Contrast of all shares:

According to the step1 and step 2 of "generation phase", we can know that for every row i in basis matrices $(S_w^{C_1 \ldots C_n}, S_b^{C_1 \ldots C_n})$ has only one "0" if $c_i = b$. In contrast, every row i in basis matrices $(S_w^{C_1 \ldots C_n}, S_b^{C_1 \ldots C_n})$ has exactly two "0" if $c_i = w$. Therefore, $\alpha_S(m) = n - (n-1)/n + 1 = 1/n + 1$.

$Case2$: Contrast property:

When $S_w^{C_1 \ldots C_n}$ and $S_b^{C_1 \ldots C_n}$ are restricted to i rows , $i > 1$, $S_w^{C_1 \ldots C_n}$ has exactly

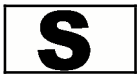
one column that contains only 0's and $S_b^{C_1 \cdots C_n}$ either has exactly two such columns or no any such columns, So for any qualified set $X \in \Gamma_{Qual}$ and for any matrix $M \in C_w^{C_1 \cdots C_n}$, we have that $w(M_X) = n - 1$ and $M' \in C_b^{C_1 \cdots C_n}$, we have that $w(M'_X) = n$ or $w(M'_X) = n - 2$. Therefore, $\alpha_F(m) = 1/n + 1$ and the contrast property hold.

*Case*3 : Security property:

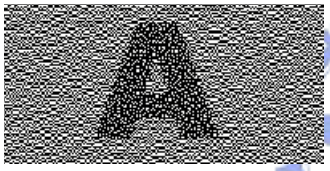
Any combination of shares of $(2, n)$-threshold EVCS must show the secret, so the security property is not required here.

Example 6.2.2. *Example 6.2.2 depicts an 2 out of 3 EVCS*$_2$*. The contrast of the example is 1/4 when the optimal contrast of original 2 out of 3 EVCS*$_1$ *is 1/6.*

Secret image



Share of participant 1



Share of participant 2

Share of participant 3

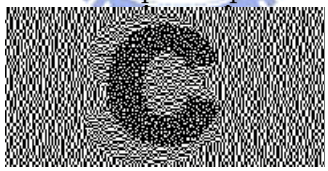


Image of participants 1 and 2

Image of participants 1 and 3

Image of participants 2 and 3

Image of participants 1, 2 and 3

Input:

1. An $(2, n)$ access structure $(\Gamma_{Qual}, \Gamma_{Forb})$ on a set P of n participants.
2. The colors $c_1, ..., c_n \in \{b, w\}$ of the pixels in the original n images.
3. The colors $c \in \{b, w\}$ of the pixel of the secret image the dealer wants to share.

Generation of the n shares:

1. Construct an $n \times n+1$ matrix W as follows:
   Set the first entry of all rows of W to 0.
   $j = 1$
   For $i = 1$ to n do
   If $c_i = b$ then set all entries of row i of W to 1 except first entry.
   else $j = j + 1$;
   set entry (i, j) of W to 0 and set all remaining entries of row i to 1;
   .
2. Construct an $n \times n+1$ matrix B as follows:
   $j = 1$
   For $i = 1$ to n do
   If $c_i = w$ then set all entries of row i of W to 1 except first entry and
   second entry to 0.
   else $j = (j + 1 \bmod (n + 1)) + 1$.
   set entry (i, j) of W to 0 and set all remaining entries of row i to 1;
3. The collection $C_c^{c_1 \cdots c_n}$ is constructed by considering the matrices obtained by permuting, in all possible ways, the columns of the matrix
$$S_c^{c_1 \cdots c_n} = \begin{cases} W & \text{if } c = w \\ B & \text{if } c = b \end{cases}$$
4. Let M be a matrix randomly chosen in $C_c^{c_1 \cdots c_n}$

Output: The matrix M

Figure 6.3: The protocol to generate the shares for EVCSs based on new definition.

Chapter 7

Conclusion and Future Work

We have proposed a new definition for visual cryptography, in which the revealed images may be lighter or darker than backgrounds. We have studied properties about our new definition. The results show that our VCS_2 indeed has better pixel expansion (contrast).

In chapter 4, we have proposed three cheating methods against VCS and EVCS. We examined previous cheat-preventing schemes and found that they are either not robust enough or still improvable. We finally proposed an efficient transformation of VCS for cheating prevention. It only added two subpixels for each pixel in the image.

In chapter 5, we have proposed three compatible VCSs with reversing, in which the contrast of the recovered image is ideal in only two runs. We also compared several properties of all the previous VCSs with reversing with ours. We also propose a method to construct a (2,n)-EVCS. Our $(2, n)$-EVCS$_2$ has smaller subpixels and better contrast than Droste's result.

After doing these researches, we think that there still are many achievable improvements on VC. The most important issue we think is constructing a practical VCS that is more efficient for every participant to recover the secret image. For example, it is desirable to design a VCS with reversing which does not need to divide the transparency into $|\Gamma_0|$ blocks and still has the same or better performance on every property than ours. Moreover, it will be a dramatic improvement if we can implement an efficient transformation of VCS for cheating prevention that only added two subpixels for each pixel and each participant just need to hold one share, instead of two. Besides, the new definition has been proposed. The more applications of the new definition on the extensions of VC may be an interesting topic to explore.

References

[1] G. Ateniese, C. Blundo, A. De Santis and D.R. Stinson, Visual Cryptography for General Access Structures, *Information and Computation* 129(2), pp.86-106, 1996.

[2] G. Ateniese, C. Blundo, A. De Santis and D.R. Stinson, Extended Capabilities for Visual Cryptography, *Theoretical Computer Science* 250(1-2), pp.143-161, 2001.

[3] I. Biehl and S. Wetzel, Traceable Visual Cryptography, *Proceedings of the 1st International Conference on Information and Communication Security* (*ICICS'97*), LNCS 1334, pp.61-71, 1997.

[4] C. Blundo and A. De Santis, Visual Cryptography Schemes with Perfect Reconstructions of Black Pixels, *Computers & Graphics* 22(4), pp.449-455, 1998.

[5] C. Blundo, A. De Santis and D.R. Stinson, On the Contrast in Visual Cryptography Schemes, *J. Cryptology* 12(4), pp.261-289, 1999.

[6] C. Blundo, P. D'Arco, A. De Santis and D.R. Stinson, Contrast Optimal Threshold Visual Cryptography Schemes, *SIAM J. Discrete Math.* 16(2), pp.224-261, 2003.

[7] A. De Bonis and A. De Santis, Randomness in Secret Sharing and Visual Cryptography Schemes, *Theoretical Computer Science* 314(3), pp.351-374, 2004.

[8] S. Cimato, A. De Santis, A.L. Ferrara and B. Masucci, Ideal Contrast Visual Cryptography Schemes with Reversing, *Information Processing Letters* 93(4), pp.199-206, 2005.

[9] S. Droste, New Results on Visual Cryptography, In *Proceedings of Advances in Cryptology-CRYPT'96*, LNCS 1109, pp.401-415, 1996.

[10] P. A. Eisen and D. R. Stinson, Threshold Visual Cryptography with Specified Whiteness Levels of Reconstructed Pixels, *Designs, Codes and Cryptography* 25(1), pp.15-61, 2002.

[11] T. Hofmeister, M. Krause and H-U. Simon, Contrast-Optimal k out of n Secret Sharing Schemes in Visual Cryptography, *Theoretical Computer Science* 240(2), pp.471-485, 2000.

[12] G.-B. Horng, T.-G. Chen and D.-S. Tsai, Cheating in Visual Cryptography, *Designs, Codes and Cryptography*, 38(2), pp.219-236, 2006.

[13] R. Ito, H. Kuwakado and H. Tanaka, Image Size Invariant Visual Cryptography, *IEICE Trans. Fundamentals* E 82-A (10), pp.2172-2176, 1999.

[14] T. Katoh and H. Imai, Some Visual Secret Sharing Schemes And Their Share Size, In *Proceedings of International Conferences on Cryptology and Information Security*, pp.41-47, 1996.

[15] K. Kim, J. Park and Y. Zheng, Human-Machine Identification Using Visual Cryptography, In *Proceedings of the 6th IEEE International Workshop on Intelligent Signal Processing and Communication Systems*, pp.178-182, 1998.

[16] K. Kobara and H. Imai, Limiting the Visible Space Visual Secret Sharing Schemes And Their Application to Human Identification, In *Proceedings of Advances in Cryptology - ASIACRYPT 96*, LNCS 1163, pp.185-195, 1996.

[17] M. Krause and H-U. Simon, Determining the Optimal Contrast for Secret Sharing Schemes in Visual Cryptography, *Combinatorics, Probability and Computing* 12(3), pp.285-299, 2003.

[18] D. Naccache, Colorful Cryptography - a Purely Physical Secret-Sharing Scheme Based on Chromatic Filters, In *Coding and Information Integrity*, French-Israeli Workshop, 1994.

[19] M. Naor and B. Pinkas, Visual Authentication and Identification, In *Proceedings of Advances in Cryptology - CRYPTO 97*, LNCS 1294, pp.322-336, 1997.

[20] M. Naor and A. Shamir, Visual Cryptography, In *Proceedings of Advances in Cryptology - EUROCRYPT'94*, LNCS 950, pp.1-12, 1994.

[21] M. Naor and A. Shamir, Visual cryptography II: Improving the Contrast via the Cover Base, *Cambridge Workshop on Cryptographic Protocols*, 1996. A full version is available at ftp://theory.lcs.mit.edu/pub/tcrypto/96-07.ps.

[22] V. Rijmen and B. Preneel, Efficient Colour Visual Encryption or Shared Colors of Benetton, presented at *EUROCRYPT 96 Rump Session*.

[23] A. Shamir, How to Share a Secret, *Commun. ACM* 22(11), pp.612-613, 1979.

[24] G. J. Simmons, W. Jackson, K. Martin, The Geometry of Shared Secret Schemes, *Bulletin of the ICA*, Vol 1, pp.71-88, 1991.

[25] P. Tuyls, H.D.L. Hollmann, J.H.v. Lint and L. Tolhuizen, A Polarisation based Visual Crypto System and its Secret Sharing Schemes, *http://eprint.iacr.org/2002/194*, 2002.

[26] E. R. Verheul and H. C. A. Van Tilborg, Constructions and Properties of k out of n Visual Secret Sharing Schemes, *Designs, Codes and Cryptography* 11(2), pp.179-196, 1997.

[27] D. Q. Viet and K. Kurosawa, Almost Ideal Contrast Visual Cryptography with Reversing. In *Proceedings of Topics in Cryptology - CT-RSA 2004*, LNCS 2964, pp.353-365, 2004.

[28] H. Yan, Z. Gan and K. Chen, A Cheater Detectable Visual Cryptography Scheme (in Chinese), *J. Shanghai Jiaotong University* 38(1), 2004.

[29] C.-N. Yang, New Visual Secret Sharing Schemes using Probabilistic Method, *Pattern Recognition Letters* 25(4), pp.481-494, 2004.

[30] C.-N. Yang and C.-S. Laih, Some New Types of Visual Secret Sharing Schemes, In *Proceedings of National Computer Symposium* (*NCS'99*), vol. 3, pp.260-268, 1999.

[31] C.-N Yang, C.-C Wang and T.-S Chen, Real Perfect Contrast Visual Secret Sharing Schemes with Reversing, *ACNS 06*, LNCS 3989, pp.433-447, 2006.

Appendix

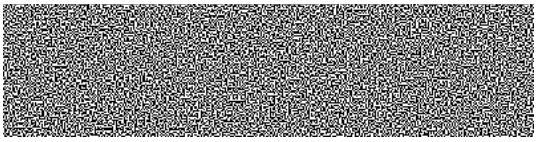
Let $\Gamma = (P, Q, F)$, where $P = \{1, 2, 3, 4\}$, $Q = \{(1, 2),\ (1, 4),\ (2, 3),\ (2, 4),\ (1, 3, 4),\ (1, 2, 3, 4)\}$ and $F = \{(1, 3),\ (3, 4),\ (1, 2, 3),\ (1, 2, 4),\ (2, 3, 4)\}$. Any (Γ, m)-VCS_1 has $m = 12$ at least. The basis matrices are:

$$S^0 = \begin{bmatrix} 101011110110 \\ 101110100000 \\ 111110110101 \\ 111011100011 \end{bmatrix} \quad \text{and} \quad S^1 = \begin{bmatrix} 101011111100 \\ 011110100000 \\ 111101111010 \\ 110111011001 \end{bmatrix}$$

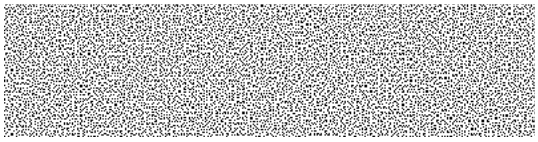
Our (Γ, m)-VCS_2 has $m = 4$ and $\alpha(m) = 1/4$. The basis matrices are

$$S^0 = \begin{bmatrix} 0011 \\ 0100 \\ 0100 \\ 0011 \end{bmatrix} \quad \text{and} \quad S^1 = \begin{bmatrix} 0011 \\ 0001 \\ 1000 \\ 0101 \end{bmatrix}.$$

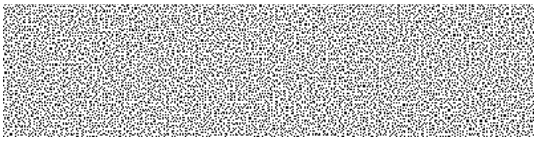
The following shows the shares of all participants and images of the stacked shares of participants of qualified and forbidden sets.



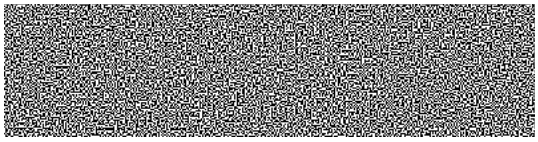
Share of participant 1



Share of participant 2



Share of participant 3



Share of participant 4

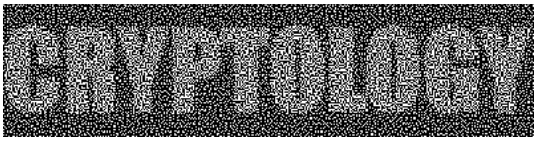


Image of participants 1 and 2

Image of participants 1 and 4

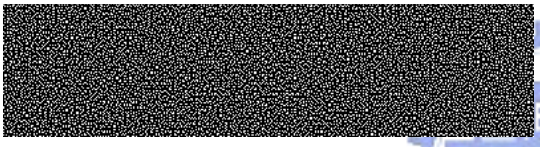


Image of participants 1 and 3

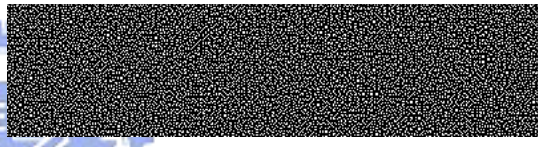


Image of participants 3 and 4

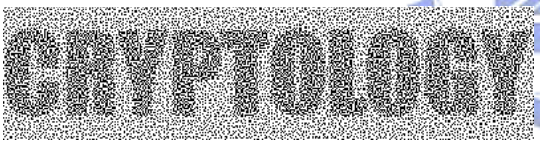


Image of participants 2 and 3

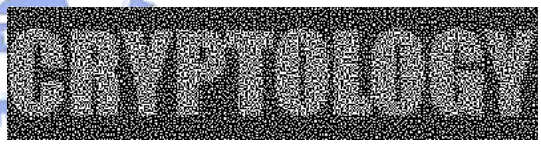


Image of participants 2 and 4

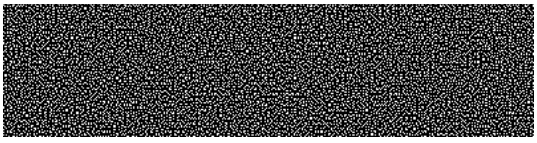


Image of participants 1 and 2 and 3

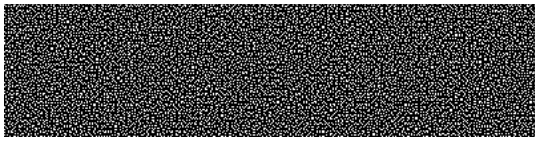


Image of participants 1 and 2 and 4

Image of participants 1 and 3 and 4

Image of participants 1 and 2 and 3 and 4