# EVCS-DAS: Evolving Visual Cryptography Schemes for Dynamic Access Structures

XIAOTIAN WU, College of Cyber Security, College of Information Science and Technology, and Guangdong Key Laboratory of Data Security and Privacy Preserving, Jinan University, China
XINJIE FENG, Department of Computer Science and Guangdong Key Laboratory of Data Security and Privacy Preserving, Jinan University, China
BING CHEN, School of Cyber Security, Guangdong Polytechnic Normal University, China
CHING-NUNG YANG, Computer Science and Information Engineering, National Dong Hwa University, Taiwan
QING-YU PENG, Department of Computer Science and Sino-French Joint Laboratory for Astrometry, Dynamics and Space Science, Jinan University, China
WEIQI YAN, Department of Computer Science, Auckland University of Technology, New Zealand

A systematic investigation of evolving visual cryptography scheme (EVCS) is carried out in this paper. The evolving scheme, denoted as $(k, \infty)$, differs from the $(k, n)$ threshold in that it permits an arbitrary and perhaps unlimited number of participants. More importantly, the access structure can be updated dynamically by adding new users. First of all, a preliminary implementation strategy for the $(2, \infty)$ EVCS is introduced. Then, by employing the $(2, 2)$ VCS recursively with the $(2, \infty)$ EVCS, a $(k, \infty)$ EVCS is created. In order to enhance the performance, an improved scheme is constructed based on the multi-secret VCS (MVCS) and a series of EVCS schemes with thresholds of $(1, \infty), \cdots, (k-1, \infty)$. Moreover, Boolean XOR operation is adopted for secret recovery to further improve the visual quality. To facilitate the XOR decryption, a novel access structure partition algorithm is presented. Additionally, the proposed partition method can successfully solve the security issue in existing multi-secret XOR-based VCS (MXVCS). By integrating the more secure MXVCS into the improved scheme, XOR decryption is provided. The two proposed methods are shown to be effective and advantageous through extensive experiments and comparisons.

CCS Concepts: • **Security and privacy** → **Security services**; **Cryptography**.

Additional Key Words and Phrases: Secret sharing, visual cryptography, evolving, dynamic access structure, infinite participants.

Authors' addresses: Xiaotian Wu (corresponding author), wxt.sysu@gmail.com, College of Cyber Security, College of Information Science and Technology, and Guangdong Key Laboratory of Data Security and Privacy Preserving, Jinan University, Guangzhou, China; Xinjie Feng, Department of Computer Science and Guangdong Key Laboratory of Data Security and Privacy Preserving, Jinan University, Guangzhou, China; Bing Chen, School of Cyber Security, Guangdong Polytechnic Normal University, Guangzhou, China; Ching-Nung Yang, Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan; Qing-Yu Peng, Department of Computer Science and Sino-French Joint Laboratory for Astrometry, Dynamics and Space Science, Jinan University, Guangzhou, China; WeiQi Yan (corresponding author), wyan@aut.ac.nz, Department of Computer Science, Auckland University of Technology, Auckland, New Zealand;


ACM 1551-6865/2024/12-ART
https://doi.org/10.1145/3708547

## 1 INTRODUCTION

Secret sharing is a popular technique for safeguarding a secret among a set of participants (or users). The well-known $(k, n)$ secret sharing scheme [16] encodes a secret into $n$ pieces of private information called shares or shadows. By gathering any $k$ shadows, the secret can be unlocked. Whereas, the set having less than $k$ shadows gives no clue about the secret. Secret sharing is now frequently utilized as a fundamental component in numerous security applications [5, 9, 12, 14, 15, 31, 32, 34, 37]. Visual cryptography scheme (VCS) [10, 11, 18, 24, 26], also called visual secret sharing, is a special category of secret sharing that can protect binary images based on human visual system. Differing from secret sharing, VCS recovers the secret in a visual manner by printing the shadows on transparencies and stacking them together. As a result, VCS enjoys the benefit of easy decoding.

With the fundamentals of VCS [13], various techniques with different functionalities were introduced. Some basic properties of VCS (i.e., pixel expansion, contrast, sharing capacity, and access structure) are usually discussed. In VCS, a secret pixel is encoded into $m \geq 2$ sub-pixels for each shadow where $m$ is referred to as pixel expansion. The created shadow is $m$ times of the secret. With pixel expansion, the overhead for storing and sending shadows grows as a result. Probabilistic VCS (PVCS) [35] and random grid-based VCS (RG-VCS) [2] were presented as solutions to the pixel expansion problem. On the other hand, integer linear programming (ILP) can be incorporated to achieve minimum pixel expansion [6, 22]. The visual performance of VCS is usually evaluated by the contrast of recovered image. A high contrast is anticipated to enable human eye to easily distinguish the reconstructed information. Consequently, approaches for enhancing the contrast [10, 28] were developed. When more shadows are stacked to recover the secret, the background of the overlay result will be noticeably darker. To provide brighter background for decrypted image, XOR-based VCS [8, 19, 27] and reversing-based VCS [25] were employed.

Multi-secret VCS (MVCS) is designed to enhance the sharing capacity by encoding numerous secrets into shadows. There exist two types of MVCS. In order to reveal distinct secrets, one category [20, 21] permits one or more extra operations (e.g., flipping or rotation) before the stacking operation. The other class [3, 23, 27] superimposes different numbers of shadows directly to reveal various secrets. Keep in mind that the former only encrypts a particular number of secrets, whereas the latter provides a general approach to deal with multiple secrets.

An access structure specifies the collection of subsets of participants which can decode the secret. One of the most commonly-used access structures is the $(k, n)$ threshold. To implement complicated sharing strategies, general access structure was investigated. However, current VCS cannot efficiently deal with the scenario that new users are frequently added. Usually, putting new participants into an already-existing access structure makes the current scheme unworkable. A *withdraw-and-rebuild* procedure would be employed if adding a new user is absolutely necessary. In this situation, the current shadows will be withdrew and then new shadows for the updated access structure will be created and distributed for each participant, including the new user. However, such a *withdraw-and-rebuild* process becomes inefficient when new users are frequently required.

In this research, we initiate a systematic investigation of evolving VCS (EVCS) which enables the efficient modification of access structure. The access structure of EVCS is denoted as $(k, \infty)$. There is no predetermined restriction on the number of users, and it might potentially be infinite. More importantly, the access structure can be changed dynamically upon the arrival of new participants. In this case, the *withdraw-and-rebuild* process is no longer applied. Instead, only new shadows are constructed and delivered to the new users. Main contribution of this paper is given below.

- A general method for implementing the $(k, \infty)$ EVCS is introduced. First of all, a basic $(2, \infty)$ EVCS is developed. By the recursive usage of the $(2, 2)$ VCS with the $(2, \infty)$ EVCS, a $(k, \infty)$ method is accomplished. Properties of the $(2, \infty)$ and $(k, \infty)$ schemes are theoretically analyzed.
- When $k$ increases, the number of shadows that the proposed $(k, \infty)$ scheme delivers to a new participant grows dramatically. The overhead for storing and transmitting the shadows burdens as a result. In order

to lower the number of shadows, an improved $(k, \infty)$ EVCS is investigated. Based on the concept of generation, the improved method is constituted by the multi-secret VCS (MVCS) and a series of EVCS schemes with thresholds of $(1, \infty), \cdots, (k-1, \infty)$. Theoretical verification is provided for the improved method as well.

- To enable XOR decryption for the improved method, a novel access structure partition algorithm is presented. The partition approach solves a security issue in existing multi-secret XOR-based VCS (MXVCS). So that a more secure MXVCS can be integrated into the improved scheme to offer XOR decoding.

## 2 PRELIMINARY

### 2.1 VCS and PVCS

Two $(n \times m)$ base matrices $B^0$ and $B^1$ constitute a $(k, n)$ VCS [13]. In this paper, digits 0 and 1 represent the white and black pixels, respectively. For any white (resp. black) secret pixel, a distribution matrix is randomly chosen from the collection $C^0$ (resp. $C^1$), and the $i$-th $(1 \le i \le n)$ row is given to the $i$-th shadow. Collections $C^0$ and $C^1$ are obtained by randomly permuting the columns of $B^0$ and $B^1$, respectively, in all possible ways. Definition 1 shows the properties of $B^0$ and $B^1$ [36]. Suppose $B^s$, $s \in \{0, 1\}$, is the base matrix for sharing a secret pixel $s$. Given a set of involved participants $X$, $(B^s|X)$ represents a $|X| \times m$ matrix which chooses the rows corresponding to the participants in $X$ from the matrix $B^s$. Additionally, the OR-ed result of all rows in $(B^s|X)$ is denoted as $OR(B^p|X)$, and the Hamming weight of $OR(B^s|X)$ is calculated as $H(OR(B^s|X))$. In Definition 1, the number of white pixels (i.e., brightness) in a recovered sub-pixel block correlated to a black (resp. white) secret pixel is represented as $l$ (resp. $h$). The first property in Definition 1 is called the correctness condition which ensures the secret is visually revealed from $k$ or more shadows. The second property is referred to as the security condition which guarantees any collection having $(k-1)$ or fewer shadows cannot disclose any clue about the secret. The visual performance of VCS is evaluated by contrast, as calculated by $\alpha = (h-l)/m$.

Definition 1. *A $(k, n)$ VCS can be built by two $(n \times m)$ base matrices $B^0$ and $B^1$, which satisfy the following conditions.*

*(Correctness): For any qualified subset $X$ (i.e., $|X| \ge k$), $H(OR(B^1|X)) \ge (m-l)$ and $H(OR(B^0|X)) \le (m-h)$, where $0 \le l < h \le m$.*

*(Security): For any forbidden subset $X$ (i.e., $|X| < k$), $H(OR(B^1|X)) = H(OR(B^0|X))$.*

Conventional VCS utilizes $m \ge 2$ sub-pixels to replace a secret pixel. The shadow is $m$ times of the secret. We refer to this as pixel expansion. To solve the drawback of pixel expansion, PVCS [35] was developed. Instead of expanding a secret pixel to $m$ sub-pixels, the primary concept of PVCS is to adopt the probability of a white pixel appearing in a secret pixel. When sharing a secret pixel, a column vector, randomly chosen from the base matrix of a conventional VCS, is employed to produce $n$ shared pixels. In this scenario, the shadow has the same size as the secret. Note that, the contrast in a PVCS is the same as that of a conventional VCS.

### 2.2 MVCS

A $(k, n, h)$ MVCS [3, 23] encrypts $h$ secrets into $n$ shadows in such a way that the stacked results by $k, k+1, \cdots, n$ shadows can gradually disclose the 1-st, 2-nd, $\cdots$, $h$-th secrets where $h = n-k+1$. Furthermore, a more general scheme, called $(k, n, h, R)$ MVCS, was also presented in [23]. Differing from the $(k, n, h)$ approach, a revealing list $R = \{r_k, r_{k+1}, \cdots, r_n\}$ with $k \le u \le n$ and $r_u \in \{0, 1, 2, \cdots, h\}$ controls the decoding of secrets. In addition, the number of secrets $h$ in $(k, n, h, R)$ MVCS can be any number between 2 and $n-k+1$ (i.e., $2 \le h \le n-k+1$). Two types of $r_u$s $(k \le u \le n)$ are employed in the revealing list: zero and non-zero items. When a zero item is used (i.e., $r_u = 0$), the superimposed result by any $u$ shadows cannot reveal any information about the secrets. On the other hand, with the usage of a non-zero item (i.e., $r_u \ne 0$), the correlated secret can be disclosed by

any collection having $u$ shadows. Note that, for the $h$ non-zero items $r_{j_1}, \cdots, r_{j_h}$ with $j_1 < \cdots < j_h$, they should satisfy $r_{j_1} < \cdots < r_{j_h}$. If $h = n - k + 1$ and $r_k = 1, r_{k+1} = 2, \cdots, r_n = n - k + 1$, the $(k, n, h, R)$ MVCS is reduced to a $(k, n, h)$ MVCS.

## 3 THE PROPOSED METHOD

Initially, analysis on existing methods and motivation of this paper are presented in this section. A scheme for $(2, \infty)$ threshold is then developed. By combining the $(2, \infty)$ method with the $(2, 2)$ VCS, a $(k, \infty)$ technique is established. Theoretical analysis on both schemes is presented at the end of this section.

### 3.1 Analysis and motivation

As the quantity of participants is pre-determined, current VCS is unable to handle dynamic access structure well. In this paper, we discuss the scenario that the access structure of VCS is modified frequently. One alternative solution for updating the access structure efficiently is to let the number of users be infinite. As new users arrive, we just simply derive and deliver new shadows for these incoming participants. The *withdraw-and-rebuild* process is no longer adopted. Nevertheless, developing a VCS with infinite participants is challenging due to the way of constructing shadows in existing VCS. Conventional techniques usually employ base matrices to generate shadows. The amount of users determines how many rows a base matrix has. It implies the number of users must be determined in advance so that the shadows can be generated. Even though the base matrices are no longer used in RG-based VCS, the number of users is still required for shadow generation. According to the above analysis, a novel EVCS with infinite users is preferred. The objective of having infinite participants in VCS is to update the access structure efficiently when new users are added.

### 3.2 The $(2, \infty)$ scheme

Given a binary pixel $s = S(x, y)$ from a $W \times H$ secret image $S$. The generation of shadow pixels for the proposed $(2, \infty)$ EVCS is described as follows. A shadow pixel $r_1$ for the 1-st participant $P_1$ is randomly generated, as depicted by $r_1 = R(\cdot)$ where procedure $R(\cdot)$ randomly outputs a black or white pixel. Then, record $r_1$ as the auxiliary information for this location (i.e., $A(x, y) = r_1$). Let the $t$-th ($t \geq 2$) participant $P_t$ be an incoming user. A corresponding shadow pixel for $P_t$ is achieved based on the secret pixel $s$ and the auxiliary information $A(x, y)$: if $s = 0$, $r_t = A(x, y)$; otherwise, $r_t = R(\cdot)$. When all secret pixels have been processed, the corresponding shared pixels form the shadow image.

### 3.3 The $(k, \infty)$ scheme

Based on the proposed $(2, \infty)$ approach, a general $(k, \infty)$ scheme is presented. Differing from the $(2, \infty)$ technique with each participant holding one shadow, a user in the $(k, \infty)$ approach might receive multiple shadows. The $(2, \infty)$ and $(2, 2)$ schemes are used as primitives to constitute the $(k, \infty)$ technique. We denote the shadow constructions of the $(2, \infty)$ and $(2, 2)$ schemes as $VCS^{SC}_{(2,\infty)}(s, I)$ and $VCS^{SC}_{(2,2)}(s, r)$, respectively, where $s$ is a secret pixel, $I$ is the additional information required for the $(2, \infty)$ scheme (e.g., the order of the current shadow and the auxiliary information which records the first shadow pixel), and $r$ is a shared pixel of the $(2, 2)$ case. Herein, the $(2, 2)$ method is implemented by PVCS [35] without pixel expansion. The following describes the shadow construction of the proposed $(k, \infty)$ EVCS. Given a secret pixel $s$, the shared pixels belonging to the $t$-th participant $P_t$ are generated based on the two cases of $t$: (i) $1 \leq t \leq k - 2$, and (ii) $t > k - 2$.

(i) $1 \le t \le k-2$. The following random pixels are generated and assigned to $P_t$.

$$\begin{cases} e_t^t, \\ e_t^{j_1 t}, \text{ for all } 0 < j_1 < t, \\ e_t^{j_1 j_2 t}, \text{ for all } 0 < j_1 < j_2 < t, \\ \quad \vdots \\ e_t^{j_1 \cdots j_{t-1} t}, \text{ for all } 0 < j_1 < \cdots < j_{t-1} < t. \end{cases} \tag{1}$$

In the above formula, the subscript of $j_{t-1}$ of the last random pixel $e_t^{j_1 \cdots j_{t-1} t}$ should satisfy $t-1 \ge 1$. These random pixels are the shadow pixels of $s$ for $P_t$.

(ii) $t > k-2$. Two types of shadow pixels are derived: (a) the random pixels and (b) the shared pixels constructed through the $(2,2)$ and $(2,\infty)$ schemes. For the first type, the random pixels are achieved by

$$\begin{cases} e_t^t, \\ e_t^{j_1 t}, \text{ for all } 0 < j_1 < t, \\ e_t^{j_1 j_2 t}, \text{ for all } 0 < j_1 < j_2 < t, \\ \quad \vdots \\ e_t^{j_1 \cdots j_{k-3} t}, \text{ for all } 0 < j_1 < \cdots < j_{k-3} < t. \end{cases} \tag{2}$$

Note: the subscript of $j_{k-3}$ of the last random pixel $e_t^{j_1 \cdots j_{k-3} t}$ should meet the condition $k-3 \ge 1$. Additionally, Formula (2) is different from Formula (1). The last random pixel in Formula (2) is $e_t^{j_1 \cdots j_{k-3} t}$ while the last random element in Formula (1) is $e_t^{j_1 \cdots j_{k-1} t}$. In the next step, construct the second type of shared pixels $f_t^{j_1 \cdots j_{k-2}}$ for all $0 < j_1 < \cdots < j_{k-2} < t$ by

$$f_t^{j_1 \cdots j_{k-2}} = VCS_{(2,\infty)}^{SC}(\tilde{e}^{j_1 \cdots j_{k-2}}, I_t^{j_1 \cdots j_{k-2}}) \tag{3}$$

where $I_t^{j_1 \cdots j_{k-2}}$ is the additional information required for the $(2,\infty)$ method and $\tilde{e}^{j_1 \cdots j_{k-2}}$ is achieved by using the $(2,2)$ scheme recursively: $\tilde{e}^{j_1 \cdots j_{k-2}} = VCS_{(2,2)}^{SC}(\tilde{e}^{j_1 \cdots j_{k-3}}, e_{j_{k-2}}^{j_1 \cdots j_{k-2}})$, $\tilde{e}^{j_1 \cdots j_{k-3}} = VCS_{(2,2)}^{SC}(\tilde{e}^{j_1 \cdots j_{k-4}}, e_{j_{k-3}}^{j_1 \cdots j_{k-3}})$, $\cdots$, $\tilde{e}^{j_1} = VCS_{(2,2)}^{SC}(s, e_{j_1}^{j_1})$.

Herein, $f_t^{j_1 \cdots j_{k-2}}$ is the shared pixel generated via the $(2,\infty)$ scheme. The superscript $j_1 \cdots j_{k-2}$ of $f_t^{j_1 \cdots j_{k-2}}$ confirms the corresponding secret $\tilde{e}^{j_1 \cdots j_{k-2}}$. Remember that, the subscript $t$ indicates that $f_t^{j_1 \cdots j_{k-2}}$ belongs to $P_t$. It does mean it is the $t$-th shared pixel in the $(2,\infty)$ scheme. When generating the shadow pixel, the order of $f_t^{j_1 \cdots j_{k-2}}$ in the $(2,\infty)$ method should be specified in the additional information $I_t^{j_1 \cdots j_{k-2}}$. When the same secret $\tilde{e}^{j_1 \cdots j_{k-2}}$ is utilized, different shared pixels are produced for different participants with the corresponding values of $t$. For example, with the same secret $\tilde{e}^{j_1 \cdots j_{k-2}}$, we can obtain $f_t^{j_1 \cdots j_{k-2}}$ and $f_{t+1}^{j_1 \cdots j_{k-2}}$ for $P_t$ and $P_{t+1}$ from the $(2,\infty)$ approach with $I_t^{j_1 \cdots j_{k-2}}$ and $I_{t+1}^{j_1 \cdots j_{k-2}}$, respectively.

In summary, the first type of pixels: $e_t^t$, $e_t^{j_1 t}$ for all $0 < j_1 < t$, $e_t^{j_1 j_2 t}$ for all $0 < j_1 < j_2 < t$, $\cdots$, $e_t^{j_1 \cdots j_{k-3} t}$ for all $0 < j_1 < \cdots < j_{k-3} < t$, and the second type of pixels: $f_t^{j_1 \cdots j_{k-2}}$ for all $0 < j_1 < \cdots < j_{k-2} < t$ are assigned to $P_t$. For any participant, the first type of pixels is collaborated with future users to decrypt the secret, while the second category of pixels is interacted with previous participants for secret recovery. Once a new user arrives, new shadows are generated by the above-mentioned steps.

### 3.4 Theoretical analysis

Properties of the proposed $(2,\infty)$ and $(k,\infty)$ approaches are analyzed theoretically. The number of shadows delivered to the $t$-th participant is provided in Theorem 1. Theorems 2 and 3 prove that both the techniques are valid constructions for dynamic access structures.

Theorem 1. *The number of shadows assigned to the $t$-th participant of the proposed $(2,\infty)$ and $(k,\infty)$ schemes is*

$$N_t = \begin{cases} 1, \ if\, k = 2, \\ 2^{t-1}, \ if\, k \geq 3, \ 1 \leq t \leq k-2, \\ \sum_{i=0}^{k-2} \binom{t-1}{i}, \ if\, k \geq 3, \ t > k-2. \end{cases} \tag{4}$$

Proof. According to the $(2,\infty)$ scheme, each participant receives only one shadow. The case of $k = 2$ holds. For the $(k,\infty)$ method with $k \geq 3$, two situations, (i) $1 \leq t \leq k-2$ and (ii) $t > k-2$, are considered. For $1 \leq t \leq k-2$, only the random shadows are given to the $t$-th user. We have $t$ different categories of random items $e_t^t$, $e_t^{j_1 t}$, $\cdots$, and $e_t^{j_1 \cdots j_{t-1} t}$ whose quantities are $\binom{t-1}{0}$, $\binom{t-1}{1}$, $\cdots$, and $\binom{t-1}{t-1}$, respectively. Thus, the total number of random shadows is calculated as $\binom{t-1}{0} + \binom{t-1}{1} + \cdots + \binom{t-1}{t-1} = 2^{t-1}$.

For $t > k-2$, two types of shadows, namely the random shadows and the shared images constructed from the $(2,2)$ and $(2,\infty)$ approaches, would be distributed to the $t$-th participant. For the random shadows, there are totally $(k-2)$ kinds of random items: $e_t^t$, $e_t^{j_1 t}$, $\cdots$, $e_t^{j_1 \cdots j_{k-3} t}$. Specifically, the quantities of $e_t^t$, $e_t^{j_1 t}$, $\cdots$, and $e_t^{j_1 \cdots j_{k-3} t}$ are $\binom{t-1}{0}$, $\binom{t-1}{1}$, $\cdots$, and $\binom{t-1}{k-3}$, respectively. As a result, the number of the first type of shadows is estimated as $\binom{t-1}{0} + \binom{t-1}{1} + \cdots + \binom{t-1}{k-3} = \sum_{i=0}^{k-3} \binom{t-1}{i}$. For the second type of shadows, the shadow pixels $f_t^{j_1 \cdots j_{k-2}}$ for all $0 < j_1 < \cdots < j_{k-2} < t$ are built for each secret pixel. The quantity of shared images from the $(2,2)$ and $(2,\infty)$ methods is $\binom{t-1}{k-2}$. Finally, the number of shadows assigned to the $t$-th participant is computed by $\binom{t-1}{k-2} + \sum_{i=0}^{k-3} \binom{t-1}{i} = \sum_{i=0}^{k-2} \binom{t-1}{i}$. According to the above analysis, the case of $k \geq 3$ holds. This theorem is proved. □

Theorem 2. *The proposed $(2,\infty)$ scheme is a valid VCS for the $(2,\infty)$ threshold.*

Proof. We demonstrate that the security and correctness requirements are satisfied. For the security condition, no single participant can learn any information about the secret. For the correctness requirement, any collection having 2 participants can decrypt the secret.

For any single shadow, when the secret pixel is white, the corresponding shadow pixel is black or white randomly. The average gray-level of the area correlated to the white secret pixels is $1 \times 0.5 + 0 \times 0.5 = 0.5$. Similarly, for a black secret pixel, the shared pixel is black or white randomly as well. The average gray-level of the area correlated to the black secret pixels is $1 \times 0.5 + 0 \times 0.5 = 0.5$. As a result, any single shadow gives no clue about the secret. The security condition is met.

When having 2 shadows, we calculate the average gray-levels of the stacked areas correlated to the white and black secret pixels. If the secret pixel is white, the two shadow pixels are the same. Both the two shadow pixels are black or white. Hence, the average gray-level is $1 \times 0.5 + 0 \times 0.5 = 0.5$. If the secret pixel is black, the two shadow pixels are randomly generated. Only when the two shadow pixels are all white, the stacked result is white. Thus, the average grayness is $1 - (0.5 \times 0.5) = 0.75$. Since $0.75 > 0.5$, the stacked area correlated to the black secret pixels is darker than the one correlated to the white secret pixels. The secret can be visually decrypted. The correctness condition is satisfied.

□

Theorem 3. *The proposed $(k,\infty)$ scheme with $k \geq 3$ is a valid VCS for the $(k,\infty)$ threshold.*

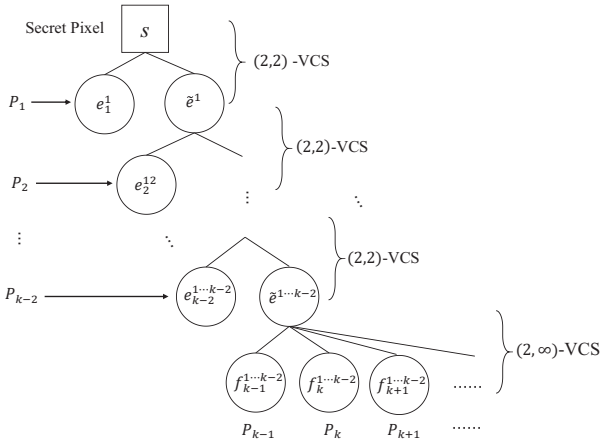



Fig. 1. A $(k, \infty)$ structure with shadow pixels distributed to a qualified subset $\{P_1, P_2, \cdots, P_{k-2}, P_i, P_j\}$ where $k-1 \leq i < j$.

Proof. We introduce the $(k, \infty)$ structure to describe the shadow pixels derived from a secret pixel $s$, as depicted in Fig. 1. The $(k, \infty)$ structure is a $(k-1)$-layer tree structure. From layers 1 to $(k-2)$, the nodes are obtained based on the $(2, 2)$ scheme. Whereas, the nodes in layer $(k-1)$ are constructed by the proposed $(2, \infty)$ technique. According to the proposed $(k, \infty)$ scheme, all shared pixels distributed to participants are the leaf nodes of the $(k, \infty)$ structure. To be more concrete, the random pixels are obtained from the leaf nodes of the first $(k-2)$ layers, while the shadow pixels of $(2, \infty)$ scheme are achieved from the $(k-1)$-th layer. As illustrated in Fig. 1, the leaf node of the 1-st layer is $e_1^1$ for $P_1$, the leaf node of the 2-nd layer is $e_2^{12}$ for $P_2$, and so forth. In the $(k-1)$-th layer, the first leaf node $f_{k-1}^{1\cdots k-2}$ obtained from the $(2, \infty)$ scheme is distributed to $P_{k-1}$, and the second one $f_k^{1\cdots k-2}$ is delivered to $P_k$, and so on. There exist various $(k, \infty)$ structures. The structure depicted in Fig. 1 distributes the leaf nodes of the 1-st, 2-nd, $\cdots$, $(k-2)$-th layers to $P_1$, $P_2$, $\cdots$, $P_{k-2}$, and delivers the 1-st, 2-nd, $\cdots$, leaf nodes of the $(k-1)$-th layer to $P_{k-1}$, $P_k$, $\cdots$. For simplicity, we use $\Omega_{\{i_1 i_2 i_3 \cdots\}}$ with $i_1 < i_2 < i_3 < \cdots$ to represent the pixel set generated from the $(k, \infty)$ structure, where the subscript $\{i_1 i_2 i_3 \cdots\}$ denotes the sequence of participants involving in this structure. For example, the pixel set given in Fig. 1 is denoted as $\Omega_{\{123\cdots\}}$.

Based on the $(k, \infty)$ structure, we prove the security condition that any less than $k$ participants give no clue about the secret. For any participant $P_t$, we have $e_t^t \in \Omega_{\{t\cdots\}}, e_t^{j_1 t} \in \Omega_{\{j_1 t\cdots\}}, \cdots, e_t^{j_1\cdots j_{k-3} t} \in \Omega_{\{j_1\cdots j_{k-3} t\cdots\}}, f_t^{j_1\cdots j_{k-3} j_{k-2}} \in \Omega_{\{j_1\cdots j_{k-3} j_{k-2} t\cdots\}}$. Since $\{t\cdots\}, \{j_1 t\cdots\}, \cdots, \{j_1\cdots j_{k-3} t\cdots\}$, and $\{j_1\cdots j_{k-3} j_{k-2} t\cdots\}$ are different from each other, the shadow pixels distributed to $P_t$ belong to different $(k, \infty)$ structures. When recovering the secret, according to the $(k, \infty)$ structure, at least $(k-2)$ leaf nodes of the first $(k-2)$ layers and 2 leaf nodes from the $(k-1)$-th layer should be collected. However, for any less than $k$ participants, at most $(k-1)$ leaf nodes are obtained. The secret cannot be decrypted. The security condition is met.

The correctness condition can be obtained via Fig. 2, where any $k$ participants $P_{i_1}, P_{i_2}, \cdots, P_{i_k}$ with $i_1 < i_2 < \cdots < i_k$ can recover the secret. In this case, $P_{i_1}$ has $e_{i_1}^{i_1}$, $P_{i_2}$ has $e_{i_2}^{i_1 i_2}$, $\cdots$, $P_{i_{k-2}}$ has $e_{i_{k-2}}^{i_1\cdots i_{k-2}}$, $P_{i_{k-1}}$ has $f_{i_{k-1}}^{i_1\cdots i_{k-2}}$, and $P_{i_k}$ has $f_{i_k}^{i_1\cdots i_{k-2}}$. $P_{i_{k-1}}$ and $P_{i_k}$ can reconstruct $\tilde{e}^{i_1\cdots i_{k-2}}$ via the $(2, \infty)$ scheme. Then, $P_{i_{k-2}}$ can combine $e_{i_{k-2}}^{i_1\cdots i_{k-2}}$ with the recovered $\tilde{e}^{i_1\cdots i_{k-2}}$ to decrypt $\tilde{e}^{i_1\cdots i_{k-3}}$ via the $(2, 2)$ approach. By recursively using the $(2, 2)$ scheme with $P_{i_{k-3}}, \cdots, P_1$, the secret pixel $s$ can be recovered finally. Any $k$ participants can decrypt the secret. The correctness condition is satisfied. Consequently, the proposed $(k, \infty)$ technique with $k \geq 3$ is proved to be a valid VCS.

□

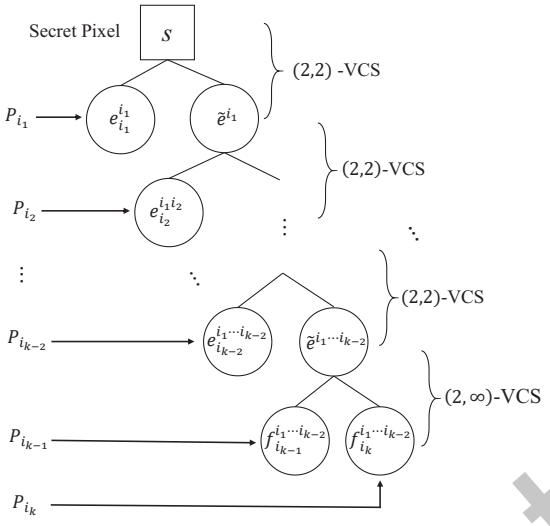



Fig. 2. Secret recovery of the proposed $(k, \infty)$ scheme by a qualified subset $\{P_{i_1}, P_{i_2}, \cdots, P_{i_k}\}$ where $i_1 < i_2 < \cdots < i_k$.

## 4 THE IMPROVED SCHEME

Theorem 1 states that as $k$ increases, there will be an increasing number of shadows delivering to new participants. Herein, we introduce an improved technique to reduce the quantity of shadows. In essence, the improved scheme is established by combining the concept of generation, MVCS with monotone property, and a series of EVCS methods with thresholds of $(1, \infty)$, $(2, \infty)$, $\cdots$, and $(k-1, \infty)$ together.

### 4.1 MVCS with monotone property

In previous $(k, n, h)$ MVCS [3, 23], $h$ secrets are revealed by stacking exactly $k, \cdots, n$ shadows. $h$ must satisfy $h = n - k + 1$. When $h < n - k + 1$, such a scheme is implemented by a more general $(k, n, h, R)$ MVCS [23] with a revealing list $R = \{r_k, r_{k+1} \cdots, r_n\}$. As mentioned in the previous $(k, n, h, R)$ MVCS, we have two categories of items in $R$. For a non-zero item (i.e., $r_u \neq 0$, $u \in \{k, \cdots, n\}$), the $r_u$-th secret is revealed by overlaying $u$ shadows. For a zero item (i.e., $r_u = 0$), the stacked result by any $u$ shadows cannot learn any information about the secrets.

Herein, we consider the $(1, n, k)$ MVCS with $k < n$. Such a scheme is implemented by the $(1, n, k, R)$ MVCS with $R = \{r_1 = 1, \cdots, r_k = k, r_{k+1} = 0, \cdots, r_n = 0\}$. When stacking $k$ shadows, the last secret (i.e., the $k$-th secret) is reconstructed since $r_k = k$. On the other hand, as $r_{k+1} = 0, \cdots, r_n = 0$, the superimposed results by $k+1, \cdots, n$ shadows give no clue about the secrets. The last secret cannot be obtained by the stacked results with $k+1, \cdots, n$ shadows. The previous $(1, n, k)$ MVCS with $k < n$ is non-monotone.

We introduce a $(1, n, k)$ MVCS with monotone property, indicating that the last secret can still be revealed by stacking more than $k$ shadows. The base matrices of this monotone MVCS are produced via the non-monotone VCS (NVCS) [3, 27]. In a $(k, n)$ NVCS, only the result by exactly $k$ shadows can disclose the secret. The stacked result by more than or less than $k$ shadows gives no clue about the secret. Let $B_{(1,n,k)}^{s_1 \cdots s_k}$ be the base matrix for encrypting $k$ secret pixels $s_1, \cdots, s_k$ in a monotone $(1, n, k)$ MVCS. It is constructed by

$$B_{(1,n,k)}^{s_1 \cdots s_k} = \tilde{B}_{(1,n)}^{s_1} || \cdots || \tilde{B}_{(k-1,n)}^{s_{k-1}} || B_{(k,n)}^{s_k} \tag{5}$$

where || denotes the matrix concatenation, $\tilde{B}^{s_1}_{(1,n)}, \cdots, \tilde{B}^{s_{k-1}}_{(k-1,n)}$ are the base matrices of the $(1,n), \cdots, (k-1,n)$ NVCSs for secrets $s_1, \cdots, s_{k-1}$, respectively, and $B^{s_k}_{(k,n)}$ is the base matrix of $(k,n)$ VCS for $s_k$. Since the $(k,n)$ VCS is monotone, the proposed MVCS is monotone as well.

## 4.2 Shadow construction

The improved $(k,\infty)$ technique adopts the concept of generation for shadow construction. Similar to [7], a generation is defined to be a collection of several consecutive participants. Let $G_i$, $i \geq 1$, be the $i$-th generation having $n_i$ consecutive participants. All users in a $(k,\infty)$ scheme can be partitioned into the following infinite generations: $G_1 = \{P_1, P_2, \cdots, P_{n_1}\}$, $G_2 = \{P_{n_1+1}, P_{n_1+2}, \cdots, P_{n_1+n_2}\}$, $G_3 = \{P_{n_1+n_2+1}, P_{n_1+n_2+2}, \cdots, P_{n_1+n_2+n_3}\}$, $\cdots$. There is no additional restriction on $n_i$ except that $n_i \geq k$. The value of $n_i$ might be determined by taking into account the application scenario, shadow size, visual performance, and other factors. As an illustration for application scenario, when 4 new participants are required for the $(3,\infty)$ scheme, we can set $n_i = 4$ to accommodate these 4 new users in one group at one time. In order to minimize the shadow size, it is anticipated that $n_i$ will be as large as possible. One can refer to Table 3. The shadow size of $P_{12}$ is 17 for the $(4,\infty)$ when $n_1 = n_2 = n_3 = 4$. However, we can set $n_1 = 12$ to include all these 12 participants in $G_1$. At that time, the shadow size of $P_{12}$ is 1. Nevertheless, when taking into account the visual performance (i.e., contrast), we should have $n_i$ to be as small as possible. For the majority of VCS techniques, larger value of $n_i$ results in worse image quality. This relationship can be observed in Table 5. For the $(2,\infty)$ (resp. $(3,\infty)$) scheme, when $n_i$ grows from 2 to 4 (resp. from 3 to 4), the contrast reduces. As a result of this, we can set $n_i = k$ to obtain good visual performance. In conclusion, determining the value of $n_i$ might consider the above-mentioned factors.

In the improved scheme, the shared pixels are constructed from the following two cases of generations: (1) $G_1$, and (2) $G_i$ with $i \geq 2$.

(1) $G_1$. For the 1-st generation $G_1 = \{P_1, P_2, \cdots, P_{n_1}\}$, construct $(k-1)$ random pixels $d^1_1, d^1_2, \cdots, d^1_{k-1}$ and obtain $(k-1)$ intermediate shared pixels via the $(2,2)$ VCS: $\tilde{d}^1_1 = VCS^{SC}_{(2,2)}(s, d^1_1)$, $\tilde{d}^1_2 = VCS^{SC}_{(2,2)}(s, d^1_2)$, $\cdots$, $\tilde{d}^1_{k-1} = VCS^{SC}_{(2,2)}(s, d^1_{k-1})$. These $(k-1)$ intermediate information is responsible for producing shared pixels in future generations. Let $d^1_1, \cdots, d^1_{k-1}, s$ be the $k$-pixel input of $(1, n_1, k)$ MVCS. We derive $n_1$ shadow pixels $m^1_1, m^1_2, \cdots, m^1_{n_1}$ by

$$(m^1_1, m^1_2, \cdots, m^1_{n_1}) = MVCS^{SC}_{(1,n_1,k)}(d^1_1, \cdots, d^1_{k-1}, s). \tag{6}$$

These $n_1$ pixels $m^1_1, \cdots, m^1_{n_1}$ are distributed to $n_1$ participants $P_1, \cdots, P_{n_1}$, respectively. According to the $(1, n_1, k)$ MVCS, any 1 participant in $G_1$ can reveal the 1-st secret $d^1_1$, any 2 participants can decrypt the 2-nd secret $d^1_2, \cdots$, and any $k$ or more users can recover the $k$-th secret $s$.

(2) $G_i$ with $i \geq 2$. For the $i$-th generation $G_i = \{P_{n_1+\cdots+n_{i-1}+1}, \cdots, P_{n_1+\cdots+n_{i-1}+n_i}\}$ with $i \geq 2$, two types of shared pixels are produced by the following operations. First of all, randomly generate $(k-1)$ pixels $d^i_1, \cdots, d^i_{k-1}$. Then, obtain $(k-1)$ intermediate shared pixels $\tilde{d}^i_1, \cdots, \tilde{d}^i_{k-1}$ via the $(2,2)$ scheme, as represented by

$$\tilde{d}^i_1 = VCS^{SC}_{(2,2)}(s, d^i_1), \cdots, \tilde{d}^i_{k-1} = VCS^{SC}_{(2,2)}(s, d^i_{k-1}). \tag{7}$$

These $(k-1)$ shared pixels $\tilde{d}^i_1, \cdots, \tilde{d}^i_{k-1}$ take charge of deriving shadow pixels in subsequent generations. Remember that we have $(i-1)$ groups of intermediate pixels from the previous $(i-1)$ generations:

$$(\tilde{d}^1_1, \cdots, \tilde{d}^1_{k-1}), \cdots, (\tilde{d}^{i-1}_1, \cdots, \tilde{d}^{i-1}_{k-1}). \tag{8}$$

Generally, for the $j$-th participant of $G_i$, the shared pixel $\tilde{d}^x_y$ with $1 \leq x \leq i-1$, $1 \leq y \leq k-1$ from the previous generation $G_x$ is considered as the secret of a $(k-y,\infty)$ scheme. Consequently, the $(k-y,\infty)$ method outputs a

shadow $w^x_{y(n_2+\cdots+n_{i-1}+j)}$ for this participant, as denoted by

$$w^x_{y(n_2+\cdots+n_{i-1}+j)} = VCS^{SC}_{(k-y,\infty)}(\tilde{d}^x_y, \hat{I}^x_{y(n_2+\cdots+n_{i-1}+j)}) \tag{9}$$

where $\tilde{d}^x_y$ is the secret and $\hat{I}^x_{y(n_2+\cdots+n_{i-1}+j)}$ indicates the additional information (e.g., the order of the current shadow pixel and the random pixels of the previous participants) needed in the $(k-y,\infty)$ approach. For any $\tilde{d}^x_y$, the same $(k-y,\infty)$ scheme is utilized to produce shadows for the participants in different generations. For example, a $(k-1,\infty)$ method with $\tilde{d}^1_1$ is utilized to build the shadows $w^1_{11}, w^1_{1(n_2+1)}, w^1_{1(n_2+n_3+1)}, \cdots$ for the 1-st participants in $G_2, G_3, G_4, \cdots$, respectively. Meanwhile, the $(1, n_i, k)$ MVCS is adopted to construct $n_i$ shared pixels $m^i_1, \cdots, m^i_{n_i}$ by

$$(m^i_1, \cdots, m^i_{n_i}) = MVCS^{SC}_{(1,n_i,k)}(d^i_1, \cdots, d^i_{k-1}, s) \tag{10}$$

where $d^i_1, \cdots, d^i_{k-1}, s$ are the $k$ input pixels. These $n_i$ pixels $m^i_1, \cdots, m^i_{n_i}$ are then delivered to $P_{n_1+\cdots+n_{i-1}+1}, \cdots, P_{n_1+\cdots+n_{i-1}+n_i}$ of $G_i$, respectively.

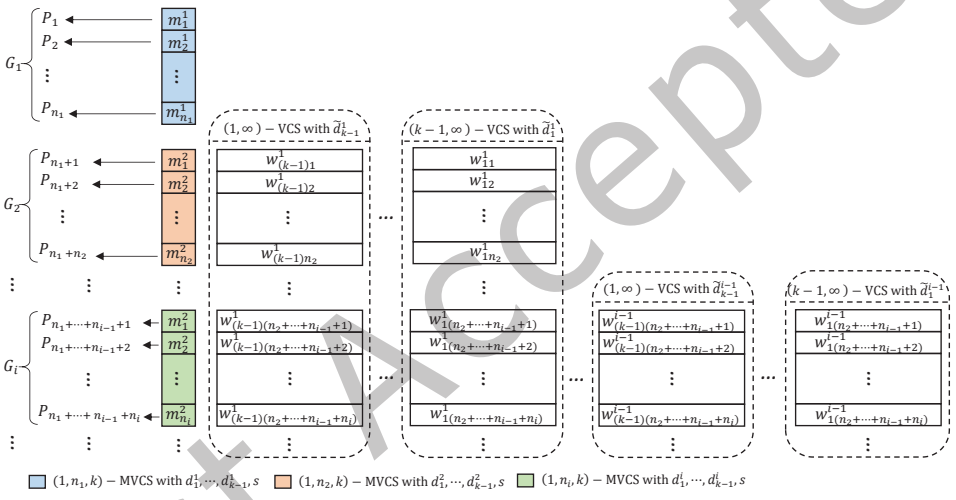



Fig. 3. Shadow pixels for participants in the improved $(k,\infty)$ scheme when sharing a secret pixel $s$.

In summary, when sharing one secret pixel $s$, the $j$-th participant $P_{n_1+\cdots+n_{i-1}+j}$ of the $i$-th generation $G_i$ receives the following shadow pixels.

$$\begin{cases} m^i_j, \\ w^1_{(k-1)(n_2+\cdots+n_{i-1}+j)}, \cdots, w^1_{1(n_2+\cdots+n_{i-1}+j)}, \\ w^2_{(k-1)(n_2+\cdots+n_{i-1}+j)}, \cdots, w^2_{1(n_2+\cdots+n_{i-1}+j)}, \\ \vdots \\ w^{i-1}_{(k-1)(n_2+\cdots+n_{i-1}+j)}, \cdots, w^{i-1}_{1(n_2+\cdots+n_{i-1}+j)}. \end{cases} \tag{11}$$

Fig. 3 illustrates the shadow pixels distributed to different participants in the improved scheme. Note that, the probabilistic method is utilized to implement all mentioned MVCS and VCS methods.

### 4.3 Theoretical analysis

The improved method is verified to be a valid $(k, \infty)$ method by Theorems 4 and 5, where the security and correctness requirements are proved to be satisfied. The number of shadows assigned to a participant by the improved technique is provided in Theorem 6.

Theorem 4. *The improved $(k, \infty)$ scheme is secure: any less than k participants cannot learn any information about the secret.*

Proof. According to Formula (11), each participant holds multiple shadows coming from different schemes. For the $j$-th participant $P_{n_1+\cdots+n_{i-1}+j}$ of $G_i$, $m_j^i$ belongs to the $(1, n_i, k)$ MVCS with secrets $d_1^i, \cdots, d_{k-1}^i, s$, while $w^1_{(k-1)(n_2+\cdots+n_{i-1}+j)}, \cdots, w^1_{1(n_2+\cdots+n_{i-1}+j)}$ come from the $(k-1, \infty), \cdots, (1, \infty)$ schemes with secrets $\tilde{d}_1^1, \cdots, \tilde{d}_{k-1}^1$, respectively. Generally, $w^{i-1}_{(k-1)(n_2+\cdots+n_{i-1}+j)}, \cdots, w^{i-1}_{1(n_2+\cdots+n_{i-1}+j)}$ belong to the $(k-1, \infty), \cdots, (1, \infty)$ schemes with secrets $\tilde{d}_1^{i-1}, \cdots, \tilde{d}_{k-1}^{i-1}$, respectively. Any participant does not have two or more shadows coming from one scheme.

Let $P_{i_1}, \cdots, P_{i_{k-1}}$ with $i_1 < \cdots < i_{k-1}$ be the $(k-1)$ participants in the improved $(k, \infty)$ scheme. A secret pixel $s$ can be recovered from the two situations: (1) using shadow pixels all from an MVCS, and (2) the combined use of some shadow pixels from an MVCS and the other ones from an EVCS. We prove the $(k-1)$ participants cannot recover the secret from these two situations.

For the first case, suppose all the $(k-1)$ participants belong to a generation $G_u$. These $(k-1)$ participants hold $(k-1)$ different shadows from the $(1, n_u, k)$ MVCS with secrets $d_1^u, \cdots, d_{k-1}^u, s$. Based on the $(1, n_u, k)$ scheme, these participants can decrypt $d_1^u, \cdots, d_{k-1}^u$ but not the secret $s$ since they do not have $k$ shadows. For the second case, suppose $(k-1)$ participants are involved. Let $c$ participants come from generation $G_u$ and the remaining $(k-1-c)$ participants arrive from the future generations $G_{u+1}, G_{u+2}, \cdots$, where $1 \le c < k-1$. Additionally, when $c = k-1$, it is the first case. For the $c$ participants of $G_u$, they can recover the random pixels $d_1^u, \cdots, d_c^u$ through the $(1, n_u, k)$ MVCS with secrets $d_1^u, \cdots, d_{k-1}^u, s$. If any one of the pixels $\tilde{d}_1^u, \cdots, \tilde{d}_c^u$ can be revealed from the remaining $(k-1-c)$ participants, secret $s$ can be disclosed via the $(2, 2)$ scheme by using the two shared pixels: $(d_1^u, \tilde{d}_1^u)$, or $\cdots$, or $(d_c^u, \tilde{d}_c^u)$. However, for the remaining $(k-1-c)$ participants of $G_{u+1}, G_{u+2}, \cdots$, they can utilize their EVCS shadows to decrypt pixels $\tilde{d}_{c+1}^u, \cdots, \tilde{d}_{k-1}^u$ via the $(k-1-c, \infty), \cdots, (1, \infty)$ schemes. $\tilde{d}_1^u, \cdots, \tilde{d}_c^u$ cannot be achieved. Secret $s$ is not able to be decoded as a result. In summary, we prove that any $(k-1)$ participants are not capable of recovering the secret. The same conclusion also holds when the numbers of participants are $(k-2), \cdots, 1$. The improved $(k, \infty)$ scheme is secure.

□

Theorem 5. *The secret of the improved $(k, \infty)$ scheme can be decrypted from any k users.*

Proof. Let $P_{i_1}, \cdots, P_{i_k}$ with $i_1 < \cdots < i_k$ be the $k$ users for revealing the secret $s$. Two cases are examined: (1) the $k$ participants from the same generation $G_u$, and (2) the first $c$ users from $G_u$, while the remaining $(k-c)$ users from future generations $G_{u+1}, G_{u+2}, \cdots$, where $1 \le c < k$.

For the first case, the $k$ participants hold $k$ different shadows from the $(1, n_u, k)$ MVCS with secrets $d_1^u, \cdots, d_{k-1}^u$, $s$. According to MVCS, these $k$ shadows can directly recover the secret $s$. For the second case, the $c$ participants of $G_u$ hold $c$ different shadows belonging to the $(1, n_u, k)$ approach with secrets $d_1^u, \cdots, d_{k-1}^u, s$. They can decode $d_c^u$ from these $c$ shadows. On the other hand, for the remaining $(k-c)$ participants of $G_{u+1}, G_{u+2}, \cdots$, they hold $(k-c)$ different shadows of the $(k-c, \infty)$ EVCS with secret $\tilde{d}_c^u$. Thus, these $(k-c)$ users can reconstruct $\tilde{d}_c^u$. Since $d_c^u$ and $\tilde{d}_c^u$ are the two shadows of the $(2, 2)$ scheme, the secret $s$ can be decrypted. Based on the two mentioned situations, any $k$ users are capable of decoding the secret. This theorem is proved.

□

Theorem 6. *The number of shadows assigned to the $t$-th participant $P_t$ of the improved $(k, \infty)$ method is estimated as*

$$N_t = \begin{cases} 1, \ \text{if } 1 \le t \le n_1, \\ 1 + \sum_{j=1}^{i-1} \sum_{y=1}^{k-1} N^{(y,\infty)}_{t-\sum_{x=1}^{j-1} n_x}, \ \text{if } i \ge 2, n_{i-1} \le t \le n_i, \end{cases} \tag{12}$$

*where $N^{(y,\infty)}_{t-\sum_{x=1}^{j-1} n_x}$ is the quantity of shadows constructed from the proposed $(y, \infty)$ scheme for the $(t - \sum_{x=1}^{j-1} n_x)$-th user, as calculated by Theorem 1.*

Proof. The shadows assigned to $P_t$ of generation $G_i$ are separated into two categories: (1) the shadows from MVCS, and (2) the others from the $(1, \infty), \cdots, (k-1, \infty)$ schemes. We use $N_t^{(1)}$ and $N_t^{(2)}$ to record the number of shadows from the first and second categories. For the first category, only 1 shadow of MVCS is provided to $P_t$. We have $N_t^{(1)} = 1$. For the second category, when $1 \le t \le n_1$ (i.e., $P_t$ belongs to the 1-st generation $G_1$), the $(1, \infty)$, $\cdots$, $(k-1, \infty)$ methods are not used to produce shadows. The quantity of shadows from the second category is 0 when $1 \le t \le n_1$. That is $N_t^{(2)} = 0$ for $1 \le t \le n_1$. When $n_{i-1} \le t \le n_i$ (i.e., $P_t$ coming from the $i$-th generation $G_i$ where $i \ge 2$), the $(1, \infty)$, $\cdots$, $(k-1, \infty)$ schemes are employed for $(i-1)$ rounds to generate the corresponding shadows. For the $j$-th round $(1 \le j \le i-1)$, $N^{(1,\infty)}_{t-a_j}, \cdots, N^{(k-1,\infty)}_{t-a_j}$ shadows are constructed by the $(1, \infty)$, $\cdots$, $(k-1, \infty)$ schemes, respectively, where $a_j = \sum_{x=1}^{j-1} n_x$, $N^{(1,\infty)}_{t-a_j}$ is 1, and $N^{(y,\infty)}_{t-a_j}$ with $2 \le y \le k-1$ is evaluated by Theorem 1. Thus, we have $N_t^{(2)} = \sum_{j=1}^{i-1} \sum_{y=1}^{k-1} N^{(y,\infty)}_{t-\sum_{x=1}^{j-1} n_x}$ for $i \ge 2$, $n_{i-1} \le t \le n_i$. In summary, we can compute the number of shadows assigned to $P_t$ by

$$N_t = N_t^{(1)} + N_t^{(2)} = \begin{cases} 1, \ \text{if } 1 \le t \le n_1, \\ 1 + \sum_{j=1}^{i-1} \sum_{y=1}^{k-1} N^{(y,\infty)}_{t-\sum_{x=1}^{j-1} n_x}, \ \text{if } i \ge 2, n_{i-1} \le t \le n_i. \end{cases} \tag{13}$$

□

## 4.4 Extension for XOR decryption

We further enhance the recovered image quality by using Boolean XOR decryption. The proposed $(k, n)$ approach can decode the secret with XOR operation while the improved approach cannot since the MVCS [23] [3] is not an XOR-based scheme. If a multi-secret XOR-based VCS (MXVCS) is applied in the improved scheme, the XOR decryption can be provided. Thus, we develop a non-monotone XVCS (NXVCS) using a novel partition algorithm in order to establish the MXVCS.

*4.4.1* ***Analysis on existing NXVCS.*** We adopt Construction 2 of NXVCS in [27] for building an MXVCS. For a $(k, n)$ NXVCS, only the XOR-ed result by exactly $k$ shadows can reveal the secret. While other results by more than or less than $k$ shadows cannot attain any information about the secret (i.e., the security condition). Nevertheless, the approach in [27] might suffer from a security flaw in which some forbidden subsets with more than $k$ participants would reveal the secret. Note that, a qualified subset is the subset of participants that can recover the secret, while a forbidden subset is the one that does not leak out any information. Additionally, a qualified subset is called minimal if any proper subset of it is a forbidden subset. We first provide a brief overview of Construction 2 and then highlight the security flaw with an example.

In Construction 2, all minimal qualified subsets of the $(k, n)$ threshold are separated into several parts by Shen et al.'s access structure partition approach [17]. Participants in each part are classified into three categories: kernel, related, and non-related participants. A related participant $i$ is connected with a kernel participant $j$ (i.e., an equivalent relationship, as denoted by $i \leftrightarrow j$). Both of them are then assigned the same rows of the $(k, k)$

VCS base matrices. For a non-related user, a row having binary random numbers is given. By concatenating the corresponding matrices of each part, the base matrices of $(k, n)$ NXVCS are produced.

We discover that the usage of Shen et al.'s algorithm [17] in Construction 2 [27] might compromise the security condition. Consider the $(2, 5)$ NXVCS. The base matrices by Construction 2 using Shen et al.'s partition technique are achieved as

$$\tilde{B}^{X,0}_{(2,5)} = \begin{bmatrix} 1010** \\ 101010 \\ 1010** \\ 101010 \\ 1010** \end{bmatrix}, \tilde{B}^{X,0}_{(2,5)} = \begin{bmatrix} 1010** \\ 011010 \\ 011001 \\ 011001 \\ 1001** \end{bmatrix}. \tag{14}$$

where $*$ denotes a binary random number. When XOR-ing the first 4 rows of $\tilde{B}^{X,0}_{(2,5)}$ and $\tilde{B}^{X,1}_{(2,5)}$, the results are $(0000**)$ and $(1111**)$, respectively. The recovered block correlated to the black secret area is darker than the one corresponding to the white secret area. The secret is revealed by 4 participants which compromises the security condition of $(2, 5)$ NXVCS.

Shen et al.'s method [17] is not suitable for NXVCS since it cannot guarantee each user is equivalent to one participant at most in each partition. Once a participant is connected with two or more users (e.g., $2 \leftrightarrow 3 \leftrightarrow 4$), the XOR-ed result by more than $k$ shadows might be reduced to the one by $k$ shadows. Consequently, the security condition is damaged. Take the partitions of $(2, 5)$ threshold given in Table 1 for example, we have $2 \leftrightarrow 3 \leftrightarrow 4$ (i.e., 2, 3 and 4 are equivalent) for the first partition by Shen et al.'s method [17]. In this case, the XOR-ed result by participants 1, 2, 3, and 4 is reduced to the result by participants 1 and 2 for the first partition.

**Algorithm 1** $L = Rel(R, Q)$.

**Input:**Two sets $R = \{i_1, \cdots, i_k\}$ and $Q = \{j_1, \cdots, j_k\}$.
**Output:**Set of equivalent relationships $L$.
**(1)** Achieve the set of common participants between $R$ and $Q$: $C = R \cap Q = \{c_1, \cdots, c_d\}$ where $0 \le d \le k-1$.
**(2)** $R$ and $Q$ can be rewritten as $R = \{c_1, \cdots, c_d, i_{x_1}, \cdots, i_{x_{k-d}}\}$ and $Q = \{c_1, \cdots, c_d, j_{y_1}, \cdots, j_{y_{k-d}}\}$. The set of equivalent relationships $L$ is established as $L = \{l_1 : i_{x_1} \leftrightarrow j_{y_1}, \cdots, l_{k-d} : i_{x_{k-d}} \leftrightarrow j_{y_{k-d}}\}$. Output $L$.

*4.4.2* ***Novel partition algorithm for NXVCS.*** A novel access structure partition algorithm is introduced to provide suitable divisions for NXVCS. Three procedures, Algorithms 1-3, are adopted to constitute the partition method (Algorithm 4). Algorithm 1 shows the steps for establishing the equivalent relationships between two minimal qualified subsets $R$ and $Q$ of the $(k, n)$ threshold, as represented by $L = Rel(R, Q)$, where $L$ denotes the set of equivalent relationships.

**Algorithm 2** $(L, C) = OnePartition(\Gamma)$.

**Input:**A set of qualified subsets $\Gamma$.
**Output:**Set of equivalent relationships $L$ and set of common participants $C$.
**(1)** Let $\Omega = \{1, \cdots, n\}$, $G = \Gamma = \{Q_1, \cdots, Q_t\}$, $j = 1$ and $L = \emptyset$. Compute $u = \min(n - k, k)$.
**(2)** If $t = 1$, then suppose $Q_1 = \{i_1, \cdots, i_k\}$ and randomly select $u$ different elements $j_1, \cdots, j_u$ from $\Omega \setminus Q_1$. Build $u$ equivalent relationships by $l_1 : i_1 \leftrightarrow j_1, \cdots, l_u : i_u \leftrightarrow j_u$. Output $L = \{l_1, \cdots, l_u\}$ and $C = \{i_{u+1}, \cdots, i_k\}$. Algorithm ends.
**(3)** If $t \ge 2$, then let $R = Q_1$ be the reference set and repeat Steps (4)-(6).
**(4)** Compute $j = j + 1$. Select $Q_j \in G$ and calculate $\Psi = Rel(R, Q_j)$ (i.e., Algorithm 1). All participants in $\Psi$ forms a set $\Delta$.
**(5)** If $\Delta \subseteq \Omega$ and $|L| + |\Psi| \le u$, then $L = L \cup \Psi$ and $\Omega = \Omega \setminus \Delta$.
**(6)** If $|L| = u$ or $j = t$, output $L$ and $C = \Omega$ and algorithm ends. Otherwise, repeat Steps (4)-(6).

By utilizing Algorithm 2, we can extract one partition from a given collection of minimal qualified subsets $\Gamma$, as denoted by $(L, C) = OnePartition(\Gamma)$, where $L$ is the set of equivalent relationships and $C$ is the set of common participants. With $L$ and $C$, all related minimal qualified subsets can be achieved via Algorithm 3, as indicated by $\Gamma = GenSet(L, C)$. Finally, Algorithm 4 outlines the processes for separating all minimal qualified subsets of $(k, n)$ threshold into $t$ parts $\Gamma_0^{(1)}, \cdots, \Gamma_0^{(t)}$. The corresponding $t$ pairs of sets of equivalent relationships and sets of common participants $(L^{(1)}, C^{(1)}), \cdots, (L^{(t)}, C^{(t)})$ are provided as well. With the partition results by our algorithm, a more secure NXVCS is accomplished via Construction 2 of [27].

**Algorithm 3** $\Gamma = GenSet(L, C)$.

**Input:**Set of equivalent relationships $L$ and set of common participants $C$.
**Output:**A set of qualified subsets $\Gamma$.
**(1)** If $C = \emptyset$, then $L = \{l_1 : i_1 \leftrightarrow j_1, \cdots, l_k : i_k \leftrightarrow j_k\}$. Select every element from every equivalent relationship to form $2^k$ different $k$-tuple subsets: $\{i_1, i_2, \cdots, i_k\}$, $\{i_1, i_2, \cdots, j_k\}$, $\cdots$, $\{j_1, j_2, \cdots, j_k\}$. These $2^k$ subsets forms $\Gamma$.
**(2)** If $C \neq \emptyset$, suppose $C = \{c_1, \cdots, c_d\}$. $L = \{l_1 : i_1 \leftrightarrow j_1, \cdots, l_{k-d} : i_{k-d} \leftrightarrow j_{k-d}\}$. Select every element from every equivalent relationship to form $2^{k-d}$ different $(k-d)$-tuple subsets: $\{i_1, \cdots, i_{k-d}\}$, $\{i_1, \cdots, j_{k-d}\}$, $\cdots$, $\{j_1, \cdots, j_{k-d}\}$. Add the $d$ elements $c_1, \cdots, c_d$ of $C$ to every $(k-d)$-tuple subset to obtain $2^{k-d}$ different $k$-tuple subsets: $\{c_1, \cdots, c_d, i_1, \cdots, i_{k-d}\}$, $\{c_1, \cdots, c_d, i_1, \cdots, j_{k-d}\}$, $\cdots$, $\{c_1, \cdots, c_d, j_1, \cdots, j_{k-d}\}$. These $2^{k-d}$ subsets forms $\Gamma$.
**(3)** Output $\Gamma$.

**Algorithm 4** Partition the set of all minimal qualified subsets of $(k, n)$ threshold.

**Input:**Set of minimal qualified subsets $\Gamma_0$ of a $(k, n)$ threshold.
**Output:**$t$ partitions $\Gamma_0^{(1)}, \cdots, \Gamma_0^{(t)}$ and $t$ pairs of sets of equivalent relationships and sets of common participants $(L^{(1)}, C^{(1)}), \cdots, (L^{(t)}, C^{(t)})$.
**(1)** Let $i = 1$ and $u = \min(n-k, k)$.
**(2)** Compute $(L^{(i)}, C^{(i)}) = OnePartition(\Gamma_0)$ (i.e., Algorithm 2).
**(3)** Calculate $\Gamma_0^{(i)} = GenSet(L^{(i)}, C^{(i)})$ (i.e., Algorithm 3).
**(4)** Update $\Gamma_0$ by $\Gamma_0 = \Gamma_0 \setminus \Gamma_0^{(i)}$ and $i = i + 1$.
**(5)** When $\Gamma_0 \neq \emptyset$, repeat Steps (2)-(5). Otherwise, output $\Gamma_0^{(1)}, \cdots, \Gamma_0^{(t)}$ and $(L^{(1)}, C^{(1)}), \cdots, (L^{(t)}, C^{(t)})$.

According to Step (5) of Algorithm 2, the participants engaged in an equivalent relationship are selected from the set $\Omega$. Once a relationship is established, the involved users are deleted from $\Omega$. It guarantees that each user is equivalent to a maximum of one participant in a single partition. Consequently, the security issue is solved. For example, our method yields 3 partitions for the $(2, 5)$ threshold, as provided in Table 1. By using Construction 2 of [27] with our partition results, the base matrices of $(2, 5)$ NXVCS are achieved as

$$\tilde{B}_{(2,5)}^{X,0} = \begin{bmatrix} 1010** \\ 101010 \\ 10**10 \\ 101010 \\ **1010 \end{bmatrix}, \tilde{B}_{(2,5)}^{X,1} = \begin{bmatrix} 1010** \\ 011010 \\ 01**01 \\ 100101 \\ **0110 \end{bmatrix}. \quad (15)$$

By using the following steps, we can verify that the secret information cannot be revealed by the XOR-ed results of any 3, 4, or 5 rows of the base matrices. Remember that, the secret information cannot be decrypted if the

average gray-levels (i.e., Hamming weights) of the recovered white and black pixels are the same. In the above $(2, 5)$ NXVCS, the average Hamming weight of the XOR-ed result by any 3 rows of $\tilde{B}^{X,0}_{(2,5)}$ is the same as that of $\tilde{B}^{X,1}_{(2,5)}$. Take the first 3 rows of $\tilde{B}^{X,0}_{(2,5)}$ and $\tilde{B}^{X,1}_{(2,5)}$ for example. Both the XOR-ed results are $(10 * * * *)$. Note that, the XOR-ed result of a random element $*$ and a determined element is still a random element $*$. The corresponding average Hamming weights are the same. More results of different combinations of 3 rows of $\tilde{B}^{X,0}_{(2,5)}$ and $\tilde{B}^{X,1}_{(2,5)}$ can be verified by the readers. By the same way, we can examine that the average Hamming weight of the XOR-ed result by any 4 (or 5) rows of $\tilde{B}^{X,0}_{(2,5)}$ is the same as that of $\tilde{B}^{X,1}_{(2,5)}$. Since the recovered white and black pixels have the same average Hamming weights, the secret information cannot be revealed. We also require the MXVCS to be monotone. The base matrix of a monotone $(1, n, k)$ MXVCS is obtained by

$$B^{X,s_1\cdots s_k}_{(1,n,k)} = \tilde{B}^{X,s_1}_{(1,n)} || \cdots || \tilde{B}^{X,s_{k-1}}_{(k-1,n)} || B^{X,s_k}_{(k,n)} \tag{16}$$

where $\tilde{B}^{X,s_1}_{(1,n)}, \cdots, \tilde{B}^{X,s_{k-1}}_{(k-1,n)}$ are the base matrices of the $(1, n), \cdots, (k-1, n)$ NXVCS schemes, and $B^{X,s_k}_{(k,n)}$ is the base matrix of $(k, n)$ XVCS with monotone property [19, 30].

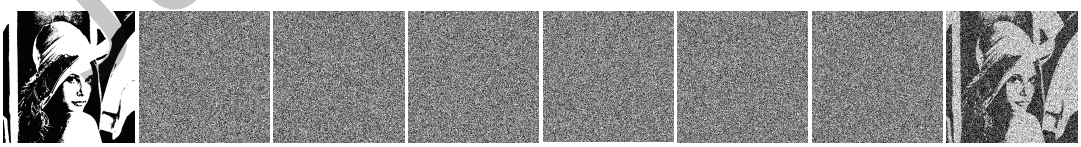
## 5 EXPERIMENTAL RESULT AND DISCUSSION

Experimental results of the two proposed methods are demonstrated. Meantime, discussions and comparisons on access structure partition, shadow size, contrast, and functionality are provided, illustrating the benefits of the proposed schemes.

### 5.1 Visual example

Two $(3, \infty)$ experiments by the proposed scheme with stacking decryption (i.e., OR recovery) and the improved method with XOR decryption are depicted in Examples 1 and 2, respectively, where 3 participants are involved. Detailed information is illustrated as below.

Example 1. *Consider a $(3, \infty)$ EVCS by the proposed technique with stacking (OR) decryption. When sharing a secret pixel $s$, for the 1-st participant, only a random pixel $e^1_1$ is produced. For the $t$-th ($t > 1$) participant, we generate random pixels and shared pixels from the $(2, \infty)$ and $(2, 2)$ methods. Thus, for $P_2$, a random pixel $e^2_2$ is provided, and meantime a shadow pixel $f^1_2$ is obtained by the $(2, \infty)$ method: $f^1_2 = VCS^{SC}_{(2,\infty)}(\tilde{e}^1, I^1_2)$ where $\tilde{e}^1$ is the shared pixel of a $(2, 2)$ VCS derived by $\tilde{e}^1 = VCS^{SC}_{(2,2)}(s, e^1_1)$. For $P_3$, we produce a random pixel $e^3_3$ and 2 shadow pixels $f^1_3$, $f^2_3$ constructed by $\tilde{e}^2 = VCS^{SC}_{(2,2)}(s, e^2_2)$, $f^1_3 = VCS^{SC}_{(2,\infty)}(\tilde{e}^1, I^1_3)$, $f^2_3 = VCS^{SC}_{(2,\infty)}(\tilde{e}^2, I^2_3)$. When all secret pixels have been processed, the corresponding shared pixels form the shadows. Fig. 4 shows this $(3, \infty)$ experiment, where the secret image is given in Fig. 4 (a) and the shadows distributed to $P_1$, $P_2$, $P_3$ are described in Figs. 4 (b)-(d). The secret recovery is listed below. For $P_1$, $P_2$, and $P_3$, the secret can be reconstructed by using stacking (OR) decryption based on $E^1_1$ of $P_1$, $F^1_2$ of $P_2$, and $F^1_3$ of $P_3$, as depicted in Fig. 4 (e).*



(a) (b) (c-1) (c-2) (d-1) (d-2) (d-3) (e)

Fig. 4. A $(3, \infty)$ experiment by the proposed technique with stacking (OR) decryption. (a) Secret image, (b) $E^1_1$ ,(c) $E^2_2$ and $F^1_2$, (d) $E^3_3$, $F^1_3$, and $F^2_3$, (e) stacking (OR) decryption by $E^1_1$, $F^1_2$, and $F^1_3$.

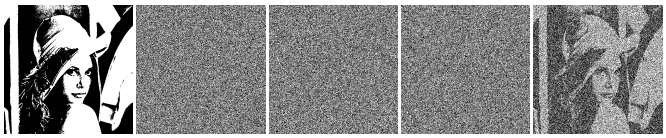



Fig. 5. A $(3,\infty)$ experiment by the improved technique with XOR decryption. (a) Secret image, (b) $M_1^1$, (c) $M_2^1$, (d) $M_3^1$, (e) XOR decryption by $M_1^1$, $M_2^1$, and $M_3^1$.

Example 2. *Consider a $(3,\infty)$ EVCS by the improved technique with XOR decryption. To constitute a $(3,\infty)$ scheme, the $(1,\infty)$ and $(2,\infty)$ methods are utilized. Let the number of participants in generation $G_1$ be $n_1 = 3$. For $G_1$, the $(1,3,3)$ MXVCS is adopted and the base matrix is built by*

$$B_{(1,3,3)}^{X,s_1s_2s_3} = \tilde{B}_{(1,3)}^{X,s_1} || \tilde{B}_{(2,3)}^{X,s_2} || B_{(3,3)}^{X,s_3} \tag{17}$$

*where $\tilde{B}_{(1,3)}^{X,s_1}$ and $\tilde{B}_{(2,3)}^{X,s_2}$ are the base matrices of $(1,3)$ and $(2,3)$ NXVCS schemes (note: the $(2,3)$ scheme is accomplished by [27] with our partition algorithm), and $B_{(3,3)}^{X,s_3}$ is the base matrix of $(3,3)$-XVCS. All base matrices are demonstrated in Appendix A.*

*Given a secret pixel $s$, for $G_1$, we generate 2 random pixels $d_1^1$ and $d_2^1$ and construct 2 intermediate shared pixels $\tilde{d}_1^1$ and $\tilde{d}_2^1$ by $\tilde{d}_1^1 = XVCS_{(2,2)}^{SC}(s, d_1^1)$ and $\tilde{d}_2^1 = XVCS_{(2,2)}^{SC}(s, d_2^1)$. With the $(1,3,3)$ MXVCS, 3 shadow pixels $m_1^1$, $m_2^1$, $m_3^1$ are derived by $(m_1^1, m_2^1, m_3^1) = MXVCS_{(1,3,3)}^{SC}(d_1^1, d_2^1, s)$ and then delivered to $P_1$, $P_2$, $P_3$ of $G_1$, respectively. When all secret pixels have been processed, the corresponding shared pixels form the shadows. The secret image is depicted in Fig. 5 (a), and the shadows belonging to $P_1$, $P_2$, $P_3$ are shown in Figs. 5 (b)-(d). They can use their MXVCS shadows $M_1^1, M_2^1, M_3^1$ to reveal the secret, as decoded in Fig. 5 (e).*

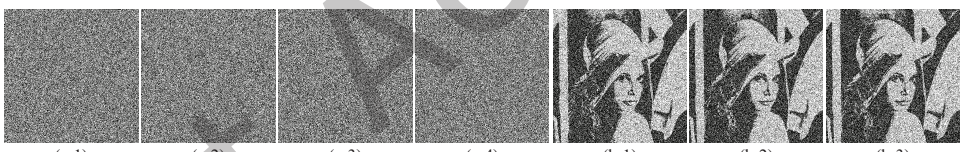



Fig. 6. Adding a new participant $P_4$ to an existing $(3,\infty)$ scheme by the proposed technique with stacking (OR) decryption. (a) $E_4^4$, $F_4^1$, $F_4^2$, $F_4^3$ for $P_4$, (b) stacking (OR) decryption by $\{P_1, P_2, P_4\}$, $\{P_1, P_3, P_4\}$, and $\{P_2, P_3, P_4\}$.

## 5.2 Dynamic access structure

When a new participant $P_4$ is required in the $(3,\infty)$ experiment of Example 1, we can utilize the proposed technique with OR decryption to generate 4 shadows $E_4^4$, $F_4^1$, $F_4^2$, $F_4^3$ for $P_4$, as depicted in Fig. 6. Detailed information of constructing the 4 shadows is provided. When a new participant $P_4$ arrives, for each secret pixel $s$, we randomly generate a pixel $e_4^4$ and 3 shared pixels $f_4^1$, $f_4^2$, and $f_4^3$ via the $(2,2)$ and $(2,\infty)$ schemes, as given by $\tilde{e}^3 = VCS_{(2,2)}^{SC}(s, e_3^3)$, $f_4^1 = VCS_{(2,\infty)}^{SC}(\tilde{e}^1, I_4^1)$, $f_4^2 = VCS_{(2,\infty)}^{SC}(\tilde{e}^2, I_4^2)$, and $f_4^3 = VCS_{(2,\infty)}^{SC}(\tilde{e}^3, I_4^3)$. As all the secret pixels have been processed, we receive the 4 shadows $E_4^4$, $F_4^1$, $F_4^2$, $F_4^3$ for $P_4$. By using $F_4^1$ of $P_4$ with $F_2^1$ of $P_2$ and $E_1^1$ of $P_1$, the secret can be decrypted via OR operation, as shown in Fig. 6 (b-1). Similarly, $P_1$, $P_3$ and $P_4$ (resp. $P_2$, $P_3$ and $P_4$) can also decrypt the secret with OR operation by their shadows $E_1^1$, $F_3^1$ and $F_4^1$ (resp. $E_2^2$, $F_3^2$ and $F_4^2$), as shown in Fig. 6 (b-2) and (b-3).

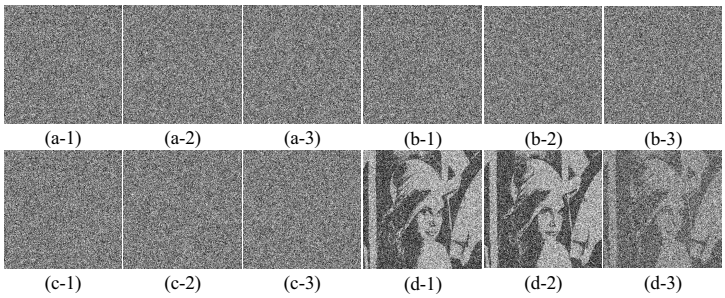



Fig. 7. Adding 3 new participants $P_4$, $P_5$, and $P_6$ to an existing $(3, \infty)$ scheme by the improved technique with XOR decryption. (a) $M_1^2$, $W_{21}^1$, $W_{11}^1$ for $P_4$, (b) $M_2^2$, $W_{22}^1$, $W_{12}^1$ for $P_5$, (c) $M_3^2$, $W_{23}^1$, $W_{13}^1$ for $P_6$, (d) XOR decryption by $\{P_4, P_5, P_6\}$, $\{P_1, P_3, P_6\}$, and $\{P_2, P_3, P_5\}$.

We consider adding 3 participants $P_4$, $P_5$ and $P_6$ of generation $G_2$ into the $(3, \infty)$ scheme of Example 2. In this case, the improved method is utilized to construct shadows for the 3 new users. Detailed information of generating shadows for the new participants $P_4$, $P_5$ and $P_6$ is illustrated. The base matrices for $G_2$ are the same as those given in Example 2. For each secret pixel $s$, we construct 2 random pixels $d_1^2$ and $d_2^2$ and generate 2 intermediate shadow pixels $\tilde{d}_1^2$ and $\tilde{d}_2^2$ via $\tilde{d}_1^2 = XVCS_{(2,2)}^{SC}(s, d_1^2)$ and $\tilde{d}_2^2 = XVCS_{(2,2)}^{SC}(s, d_2^2)$. By using the $(1, 3, 3)$ MXVCS, 3 shared pixels $m_1^2$, $m_2^2$, $m_3^2$ for $P_4$, $P_5$, $P_6$, respectively, are obtained by $(m_1^2, m_2^2, m_3^2) = MXVCS_{(1,3,3)}^{SC}(d_1^2, d_2^2, s)$. By taking the intermediate shared pixel $\tilde{d}_1^1$ of $G_1$ as secret, we utilize the $(2, \infty)$ scheme to derive 3 shadow pixels $w_{11}^1$, $w_{12}^1$, and $w_{13}^1$ for $P_4$, $P_5$, and $P_6$, respectively. Similarly, based on $\tilde{d}_2^1$ of $G_1$, the $(1, \infty)$ approach is adopted to achieve $w_{21}^1$, $w_{22}^1$, and $w_{23}^1$ for $P_4$, $P_5$, and $P_6$, respectively.

Finally, when all the secret pixels have been shared, $P_4$ receives $M_1^2$, $W_{21}^1$ and $W_{11}^1$, $P_5$ has $M_2^2$, $W_{22}^1$ and $W_{12}^1$, and $P_6$ obtains $M_3^2$, $W_{23}^1$ and $W_{13}^1$, as shown in Figs. 7 (a)-(c). $P_4$, $P_5$ and $P_6$ can decode the secret by using $M_1^2$, $M_2^2$ and $M_3^2$, as given in Fig. 7 (d-1). For other cases, such like the collection of $P_1$, $P_3$, $P_6$ (using $M_1^1$, $M_3^1$, $W_{23}^1$), or the collection of $P_2$, $P_3$, $P_5$ (using $M_2^1$, $M_3^1$, $W_{22}^1$), the XOR-ed results also disclose the secret, as illustrated in Fig. 7 (d-2) and (d-3).

## 5.3 Discussion and comparison

*5.3.1* ***Access structure partition.*** A novel access structure partition algorithm is introduced to deal with the security concern of NXVCS in [27]. The partition results by the proposed algorithm for some commonly-used thresholds (e.g., $2 \le k < n \le 5$) are provided in Table 1, as well as the ones derived from Shen et al.'s method [17]. According to Table 1, the $(2, 5)$ and $(3, 5)$ thresholds by Shen et al.'s method are not suitable for NXVCS [27] since one user is equivalent to two participants in one partition (e.g., $2 \leftrightarrow 3 \leftrightarrow 4$ in the first partition of $(2, 5)$, $3 \leftrightarrow 4 \leftrightarrow 5$ in the first partition of $(3, 5)$, $1 \leftrightarrow 2 \leftrightarrow 3$ in the third partition of $(3, 5)$). Whereas the proposed partition technique offers appropriate divisions for each threshold. We can adopt the proposed partition method to build a more secure MXVCS [27], enabling the XOR decryption for better visual quality.

*5.3.2* ***Shadow size.*** Shadow size indicates the amount of shared bits that are delivered to a participant. Shadow size is expected to be as small as possible so that the overhead for storing and transmitting the shared bits would be reduced. For the proposed $(2, \infty)$ scheme, the shadow size for each participant is 1 bit if a 1-bit secret is encoded. However, as more participants are added, the shadow size for both the proposed schemes grows as a result. The shared bits assigned to a newly-added participant by the two proposed scheme can be evaluated via Theorems 1 and 6.

Table 1. Comparison of partitions of various thresholds.

| (k, n) | Ref. [27] using Shen et al.'s method [17] | | | Our | | |
|---|---|---|---|---|---|---|
| | Partition | Equivalent | NXVCS | Partition | Equivalent | NXVCS |
| (2, 3) | $\{\{1,2\},\{1,3\}\},\{\{2,3\}\}$ | $\{2 \leftrightarrow 3\}$ | ✓ | $\{\{1,2\},\{1,3\}\}$, $\{\{1,3\},\{2,3\}\}$ | $\{2 \leftrightarrow 3\}$, $\{1 \leftrightarrow 2\}$ | ✓ |
| (2, 4) | $\{\{1,2\},\{1,3\}$, $\{2,4\},\{3,4\}\}$, $\{\{2,3\},\{1,4\}\}$ | $\{2 \leftrightarrow 3, 1 \leftrightarrow 4\}$, $\{1 \leftrightarrow 2, 3 \leftrightarrow 4\}$ | ✓ | $\{\{1,2\},\{1,3\}$, $\{2,4\},\{3,4\}\}$, $\{\{1,3\},\{1,4\}$, $\{2,3\},\{2,4\}\}$ | $\{2 \leftrightarrow 3, 1 \leftrightarrow 4\}$, $\{1 \leftrightarrow 2, 3 \leftrightarrow 4\}$, | ✓ |
| (2, 5) | $\{\{1,2\},\{1,3\},\{1,4\}$, $\{2,5\},\{3,5\},\{4,5\}\}$, $\{\{1,5\},\{2,3\},\{3,4\}\}$, $\{\{2,4\}\}$ | $\{1 \leftrightarrow 5$, $2 \leftrightarrow 3 \leftrightarrow 4\}$, $\{1 \leftrightarrow 2 \leftrightarrow 4$, $3 \leftrightarrow 5\}$ | ✗ | $\{\{1,2\},\{1,3\},\{2,4\}$, $\{3,4\}\},\{\{1,4\},\{1,5\}$, $\{2,4\},\{2,5\}\},\{\{2,3\}$, $\{2,4\},\{3,5\},\{4,5\}\}$ | $\{2 \leftrightarrow 3, 1 \leftrightarrow 4\}$, $\{4 \leftrightarrow 5, 1 \leftrightarrow 2\}$, $\{2 \leftrightarrow 5, 3 \leftrightarrow 4\}$ | ✓ |
| (3, 4) | $\{\{1,2,3\},\{1,2,4\}\}$, $\{\{1,3,4\},\{1,3,4\}\}$ | $\{2 \leftrightarrow 4\}$, $\{1 \leftrightarrow 2\}$ | ✓ | $\{\{1,2,3\},\{1,2,4\}$, $\{\{1,3,4\},\{2,3,4\}\}$ | $\{3 \leftrightarrow 4\}$, $\{1 \leftrightarrow 2\}$ | ✓ |
| (3, 5) | $\{\{1,2,3\},\{1,2,4\}$, $\{1,2,5\}\},\{\{1,3,4\}$, $\{2,3,4\},\{1,3,5\}$, $\{2,3,5\}\},\{\{2,4,5\}$, $\{3,4,5\},\{1,4,5\}\}$ | $\{3 \leftrightarrow 4 \leftrightarrow 5\}$, $\{1 \leftrightarrow 2,4 \leftrightarrow 5\}$, $\{1 \leftrightarrow 2 \leftrightarrow 3\}$ | ✗ | $\{\{1,2,3\},\{1,2,4\}$, $\{1,3,5\},\{1,4,5\}\}$, $\{\{1,2,5\},\{1,4,5\}$, $\{2,3,5\},\{3,4,5\}\}$, $\{\{1,3,4\},\{2,3,4\}$, $\{1,4,5\},\{2,4,5\}\}$ | $\{3 \leftrightarrow 4,2 \leftrightarrow 5\}$, $\{2 \leftrightarrow 4,1 \leftrightarrow 3\}$, $\{1 \leftrightarrow 2,3 \leftrightarrow 5\}$ | ✓ |
| (4, 5) | $\{\{1,2,3,4\},\{1,2,3,5\}\}$, $\{\{1,2,4,5\},\{1,3,4,5\}\}$, $\{\{2,3,4,5\}\}$ | $\{4 \leftrightarrow 5\}$, $\{2 \leftrightarrow 3\}$ | ✓ | $\{\{1,2,3,4\},\{1,2,3,5\}\}$, $\{\{1,2,4,5\},\{1,3,4,5\}\}$, $\{\{2,3,4,5\},\{1,3,4,5\}\}$ | $\{4 \leftrightarrow 5\}$, $\{2 \leftrightarrow 3\}$, $\{2 \leftrightarrow 1\}$ | ✓ |

Table 2. Comparison of shadow size (measured by bits) when sharing a 1-bit secret for the $(3, \infty)$ threshold.

| Participant | Ref. [13] | Ref. [33] | Ref. [10] | Ref. [29] | Ref. [4] | | Our | |
|---|---|---|---|---|---|---|---|---|
| | | | | | M1 | M2 | Proposed | Improved |
| $P_1$ | 22 | 1 | 1 | 1 | 1 | 6 | 1 | 1 |
| $P_2$ | 22 | 1 | 1 | 1 | 2 | 6 | 2 | 1 |
| $P_3$ | 22 | 1 | 1 | 1 | 4 | 7 | 3 | 1 |
| $P_4$ | 22 | 1 | 1 | 1 | 7 | 7 | 4 | 1 |
| $P_5$ | 22 | 1 | 1 | 1 | 11 | 14 | 5 | 3 |
| $P_6$ | 22 | 1 | 1 | 1 | 16 | 14 | 6 | 3 |
| $P_7$ | 22 | 1 | 1 | 1 | 22 | 15 | 7 | 3 |
| $P_8$ | 22 | 1 | 1 | 1 | 29 | 15 | 8 | 3 |
| $P_9$ | 22 | 1 | 1 | 1 | 37 | 23 | 9 | 5 |
| $P_{10}$ | 22 | 1 | 1 | 1 | 46 | 23 | 10 | 5 |
| $P_{11}$ | 22 | 1 | 1 | 1 | 56 | 24 | 11 | 5 |
| $P_{12}$ | 22 | 1 | 1 | 1 | 67 | 24 | 12 | 5 |

Table 3. Shadow size of the improved scheme for $(4, \infty)$ threshold when a 1-bit secret is encoded.

| Gen. | Participant | MVCS | $(k, \infty)$-VCS with $1 \leq k \leq 3$ for the $(4, \infty)$ improved scheme | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|
| | | $(1, 4, 4)$ | $(1, \infty)$ | $(2, \infty)$ | $(3, \infty)^1$ | $(1, \infty)$ | $(2, \infty)$ | $(3, \infty)^1$ | |
| $G_1$ | $P_1$ | 1 | - | - | - | - | - | - | 1 |
| | $P_2$ | 1 | - | - | - | - | - | - | 1 |
| | $P_3$ | 1 | - | - | - | - | - | - | 1 |
| | $P_4$ | 1 | - | - | - | - | - | - | 1 |
| $G_2$ | $P_5$ | 1 | 1 | 1 | 1, 1 | - | - | - | 4, 4 |
| | $P_6$ | 1 | 1 | 1 | 2, 1 | - | - | - | 5, 4 |
| | $P_7$ | 1 | 1 | 1 | 3, 1 | - | - | - | 6, 4 |
| | $P_8$ | 1 | 1 | 1 | 4, 1 | - | - | - | 7, 4 |
| $G_3$ | $P_9$ | 1 | 1 | 1 | 5, 3 | 1 | 1 | 1, 1 | 11, 9 |
| | $P_{10}$ | 1 | 1 | 1 | 6, 3 | 1 | 1 | 2, 1 | 13, 9 |
| | $P_{11}$ | 1 | 1 | 1 | 7, 3 | 1 | 1 | 3, 1 | 15, 9 |
| | $P_{12}$ | 1 | 1 | 1 | 8, 3 | 1 | 1 | 4, 1 | 17, 9 |

1: Left: using the proposed $(3, \infty)$ scheme. Right: using the improved $(3, \infty)$ scheme.

Table 4. Shadow size of the improved scheme for $(5, \infty)$ threshold when a 1-bit secret is encoded.

| Gen. | Participant | MVCS | $(k, \infty)$-VCS with $1 \le k \le 4$ for the $(5, \infty)$ improved scheme | | | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $(1, 5, 5)$ | $(1, \infty)$ | $(2, \infty)$ | $(3, \infty)^1$ | $(4, \infty)^1$ | $(1, \infty)$ | $(2, \infty)$ | $(3, \infty)^1$ | $(4, \infty)^1$ | |
| $G_1$ | $P_1$ | 1 | - | - | - | - | - | - | - | - | 1 |
| | $P_2$ | 1 | - | - | - | - | - | - | - | - | 1 |
| | $P_3$ | 1 | - | – | - | | - | - | - | - | 1 |
| | $P_4$ | 1 | - | – | - | | - | - | - | - | 1 |
| | $P_5$ | 1 | - | – | - | | - | - | - | - | 1 |
| $G_2$ | $P_6$ | 1 | 1 | 1 | 1, 1 | 1, 1 | - | - | - | - | 5, 5 |
| | $P_7$ | 1 | 1 | 1 | 2, 1 | 2, 1 | - | - | - | - | 7, 5 |
| | $P_8$ | 1 | 1 | 1 | 3, 1 | 4, 1 | - | - | - | - | 10, 5 |
| | $P_9$ | 1 | 1 | 1 | 4, 1 | 7, 1 | - | - | - | - | 14, 5 |
| | $P_{10}$ | 1 | 1 | 1 | 5, 1 | 11, 1 | - | - | - | - | 19, 5 |
| $G_3$ | $P_{11}$ | 1 | 1 | 1 | 6, 3 | 16, 4 | 1 | 1 | 1, 1 | 1, 1 | 29, 14 |
| | $P_{12}$ | 1 | 1 | 1 | 7, 3 | 22, 4 | 1 | 1 | 2, 1 | 2, 1 | 38, 14 |
| | $P_{13}$ | 1 | 1 | 1 | 8, 3 | 29, 4 | 1 | 1 | 3, 1 | 4, 1 | 49, 14 |
| | $P_{14}$ | 1 | 1 | 1 | 9, 3 | 37, 4 | 1 | 1 | 4, 1 | 7, 1 | 62, 14 |
| | $P_{15}$ | 1 | 1 | 1 | 10, 3 | 46, 4 | 1 | 1 | 5, 1 | 11, 1 | 77, 14 |

1: Left: using the proposed $(3, \infty)$ or $(4, \infty)$ scheme. Right: using the improved $(3, \infty)$ or $(4, \infty)$ scheme.

Comparison of shadow size among related methods is illustrated in Table 2, when a 1-bit secret is shared for the $(3, \infty)$ threshold. D'Arco et al.'s two $(3, \infty)$ approaches (M1 and M2) [4] and Wu et al.'s size invariant $(k, \infty)$ method [29] are included for comparison, as well as three traditional $(k, n)$ VCS schemes [10, 13, 33]. As the methods in [10, 29, 33] are size invariant, the shadow size is therefore 1. For Naor and Shamir's VCS [13], as 12 users are utilized, we evaluate the shadow size based on the $(3, 12)$ scheme. So that we achieve the shadow size as 22.

D'Arco et al.'s second method (M2) is a CRT-based technique that improves their first approach (M1). The concept of generation is also utilized in their second approach. For a fair comparison, we use the same numbers of participants in generations (e.g., $n_1 = n_2 = n_3 = 4$) for both our scheme and their second method [4]. In D'Arco et al.'s first approach, when one bit is encoded, the amount of shared bits assigned to the $t$-th participant is calculated as $\binom{t-1}{2} + t$ (bits). For their second method, the shared information for the $j$-th participant of the $g$-th generation $G_g$ is divided into 5 pieces: (1) the $g$-th shadow from the proposed $(3, \infty)$ scheme, (2) for each previous generation $G_i$ $(1 \le i \le g-1)$, a forward shadow from a $(3, N^{(i)})$ CRT-based scheme, totally $(g-1)$ shadows, (3) for the current generation $G_g$, the $(j+1)$-th shadow from a $(3, N^{(g)})$ CRT-based scheme, (4) a backward shadow from a $(3, N^{(g)})$ CRT-based scheme, which is later masked with a random number $RS^i$ from each previous generation $G_i$, totally $(g-1)$ masked shadows, and (5) a random number $RS^g$. Herein, the number of each generation is set to be 4 (i.e., $n_1 = n_2 = \cdots = 4$). Since $N^{(i)} = n_i + 2$ according to [4], we have $N^{(i)} = 6$ for $1 \le i \le g$. When encoding a 1-bit secret, the $(3, 6)$ CRT-based scheme [1] is constructed based on the following parameters: $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7$, $p_4 = 11$, $p_5 = 13$, $p_6 = 17$. With such a configuration, we can calculate the shadow size for each participant by D'Arco et al.'s second approach. For example, the 5 pieces of shared information delivered to $P_5$ (i.e., 1-st participant of $G_2$) are: (1) the 2-nd shadow of the proposed $(3, \infty)$ scheme (2 bits), (2) a forward shadow from a $(3, 6)$ CRT-based scheme of $G_1$ ($\lceil \log_2(17) \rceil = 5$ bits), (3) the 2-nd shadow from a $(3, 6)$ CRT-based scheme ($\lceil \log_2(5) \rceil = 3$ bits), (4) a backward shadow from a $(3, 6)$ CRT-based scheme of $G_2$, masked with $RS^1$ of $G_1$ ($\lceil \log_2(3) \rceil = 2$ bits), and (5) $RS^2$ of $G_2$ ($\lceil \log_2(3) \rceil = 2$ bits). As a result, the shadow size of $P_5$ is obtained as $2 + 5 + 3 + 2 + 2 = 14$ bits. By using the same way, more results on shadow size by D'Arco et al.'s second method are provided in Table 2.

Based on Table 2, our two methods obtain smaller shadow size for the $(3, \infty)$ threshold when compared with D'Arco et al.'s techniques [4]. When using our improved technique, improved shadow size is provided. Furthermore, the shadow sizes for the $(4, \infty)$ and $(5, \infty)$ thresholds by our improved approach are demonstrated

in Tables 3 and 4, respectively. Remember that, for the $(4, \infty)$ case, we can employ the proposed $(3, \infty)$ scheme or the improved $(3, \infty)$ scheme to build the desired threshold. Hence, we provide two different shadow sizes. The $(5, \infty)$ threshold is similar to the $(4, \infty)$ case. As observed from Tables 3 and 4, the improved method offers smaller shadow size.

Table 5. Comparison of contrast among related schemes.

| Case | Stacking | | | | | | XOR-ing | | | CRT | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | [13] | [33] | [10] | [29] | Proposed | Improved[1] | [29] | Proposed | Improved[1] | M1 [4] | M2 [4] |
| (2, 2) | 0.5 | 0.5 | 0.5 | 0.251 | 0.25 | $(0.25, 0.125)^2$, $(0.125, 0.0625)^4$ | 0.5 | 0.5 | $(0.5, 0.25)^2$, $(0.125, 0.0625)^4$ | N/A | N/A |
| (2, 50) | 0.02 | 0.2551 | 0.2502 | | | | | | | | |
| (2, ∞) | N/A | N/A | N/A | | | | | | | | |
| (3, 3) | 0.25 | 0.25 | 0.25 | 0.0646 | 0.1250 | (0.0769, 0.0385, $0.0096)^3$, $(0.0455, 0.0227, 0.0057)^4$ | 0.1264 | 0.5 | (0.3636, 0.3636 $(0.1818)^6$, $0.0455)^3$, (0.1667, 0.1667 $(0.0833)^6$, $0.0208)^4$ | 1 | 1 |
| (3, 50) | 0.0102 | 0.0590 | 0.0558 | | | | | | | | |
| (3, ∞) | N/A | N/A | N/A | | | | | | | | |
| (4, 4) | 0.125 | 0.125 | 0.125 | 0.0191 | 0.0625 | $(0.0250, 0.0125, 0.0031, 0.0016)^4$ | 0.0477 | 0.5 | (0.3333, 0.1667, $0.1667(0.0417)^6$, $0.0208)^4$ | N/A | N/A |
| (4, 50) | $2.3 \times 10^{-4}$ | 0.0132 | 0.0118 | | | | | | | | |
| (4, ∞) | N/A | N/A | N/A | | | | | | | | |
| (5, 5) | 0.0625 | 0.0625 | 0.0625 | 0.0093 | 0.0313 | (0.0083, 0.0041, 0.0010, $5.2 \times 10^{-4}$, $2.6 \times 10^{-4})^5$ | 0.0155 | 0.5 | (0.2540, 0.2540 $(0.1270)^6$, 0.0952 $(0.0317)^6$, 0.0476 $(0.0159)^6$, $0.008)^5$ | N/A | N/A |
| (5, 50) | $3.8 \times 10^{-5}$ | 0.0029 | 0.0025 | | | | | | | | |
| (5, ∞) | N/A | N/A | N/A | | | | | | | | |

1: From left to right: contrast by $k$ shadows from one generation, contrast by $(k-1)$ shadows from one generation and 1 shadow from another, $\cdots$, and contrast by $k$ shadows from $k$ different generations.
2: The number of participants in each generation is 2. 3: The number of participants in each generation is 3.
4: The number of participants in each generation is 4. 5: The number of participants in each generation is 5.
6: Two different contrasts in MXVCS result in two different contrasts in the improved scheme.

*5.3.3* ***Visual performance.*** Contrast comparison among related methods [4, 10, 13, 29, 33] is provided in Table 5. All contrasts are calculated theoretically. For VCS, the results are roughly divided into two segments: contrast by stacking decryption and contrast by XOR decryption. Note that, D'Arco et al.'s schemes are not VCS such that they cannot recover the secret by stacking. Their two approaches [4] are employed for comparison as well. As it turns out, their two techniques would make use of XOR decryption (the first method, M1) and arithmetic computation for CRT decryption (the second method, M2). The secret would be perfectly decrypted. As a result of this, the contrast is 1. D'Arco et al.'s schemes, however, are confined to $(3, \infty)$. The traditional VCS [13], Yan et al.'s method [33] and Liu et al.'s approach [10] are only applicable for $(k, n)$ threshold. The $(k, \infty)$ threshold cannot be implemented.

The base matrices used for constituting the improved scheme are provided in Appendix A. Theoretical estimation of contrast for the two proposed techniques is depicted in Appendix B. For the improved scheme, different numbers of participants in generations introduce different contrasts. For instance, we have $(0.25, 0.125)$ and $(0.125, 0.0625)$ for the $(2, \infty)$ threshold when the numbers of participants in each generation are 2 and 4, respectively. In addition, we also have diverse contrasts calculated from different combinations of shadows. For example, we obtain 3 different contrasts $(0.0769, 0.0385, 0.0096)$ for the $(3, \infty)$ case, which are calculated from the results by stacking 3 shadows from 1 generation, 2 shadows from 1 generation and 1 shadow from another, and 3 shadows from 3 different generations, respectively.

Referring to the results by stacking decryption in Table 5, the proposed technique offers higher contrast when compared with [29]. On the other hand, while comparing to the $(k, n)$ schemes [13] [33] [10], the proposed approach exhibits better visual performance when $n$ increases. XOR decryption is provided by both the proposed and improved schemes for enhancing the visual quality. According to the results by XOR decryption, the proposed method can achieve a contrast of 0.5 for all cases, which greatly outperforms existing VCS methods. For the

improved scheme, the best contrast for each threshold is offered by the result of XOR-ing $k$ shadows from a single generation. The best contrast of the improved scheme is also much better than the other VCS schemes.

Table 6. Feature comparison among related schemes.

| Scheme | Feature | | | | |
|---|---|---|---|---|---|
| | $(k, \infty)$ | Efficient generation | Lossless recovery | Easy decoding | Shadow size |
| Ref. [13] | ✗ | ✓ | ✗ | ✓(stacking) | Expanded |
| Ref. [33] | ✗ | ✓ | ✗ | ✓(stacking) | Non-expanded |
| Ref. [10] | ✗ | ✓ | ✗ | ✓(stacking) | Non-expanded |
| Ref. [29] | ✓ | ✗ | ✗ | ✓(stacking, XOR-ing) | Non-expanded |
| Ref. [4] | ✗, $(3, \infty)$ | ✓ | ✓ | ✗(XOR-ing, CRT decoding) | Expanded |
| Our | ✓ | ✓ | ✗(better than [10, 29, 33]) | ✓(stacking, XOR-ing) | Expanded(less than [4]) |

*5.3.4* ***Feature comparison.*** Feature comparison is illustrated in Table 6. When comparing to existing methods, our methods obtain the following merits.

**Dynamic access structure.** Our methods can implement the $(k, \infty)$ threshold. The number of participants is not determined in advance and might potentially be infinite. The access structure is allowed to be updated dynamically by adding new users. While the traditional techniques [10, 13, 33] are only applicable for $(k, n)$ case. They cannot efficiently deal with the scenario that new users are frequently added. Once a new user is required, the *withdraw-and-rebuild* process would be utilized. For example, when the $(n+1)$-th new user is needed in the $(2, n)$ scheme, the current $n$ shadows holding by $n$ participants should be discarded, and $(n+1)$ new shadows are generated via the $(2, n+1)$ technique and delivered to $(n+1)$ users. But for the proposed $(2, \infty)$ method, only one shadow should be constructed and then transmitted to this new participant. Furthermore, the method in [4] is a $(3, \infty)$ scheme, which is limited for applications.

**Improved contrast and reduced shadow size.** Our schemes can offer enhanced performance on contrast when compared with [10, 29, 33]. When XOR decryption is utilized, the improvement on contrast is more significant. For the aspect of shadow size, the participants would receive less shared information by the two proposed techniques when comparing to [4].

**Easy decoding by both stacking and XOR-ing.** Easy recovery of secret is provided by the proposed methods. Both stacking operation and Boolean XOR operation can be adopted to decode the secret. Usually, when computational devices are not available, the secret recovery can be accomplished by printing the shadows on transparencies and stacking them together. If a lightweight computational device is available, the secret can be reconstructed via Boolean XOR operation.

**Efficient shadow generation.** For the approach in [29], they model the $(k, \infty)$-VCS as a contrast-maximizing problem and develop a simulated annealing-based algorithm to solve this problem. The simulated annealing-based method is computationally expensive. However, we can implement the $(k, \infty)$ scheme easily by using the two proposed methods.

However, the proposed methods might be not competitive with the size invariant VCS methods [10, 29, 33], as the proposed techniques assign multiple shadows to a participant. The overhead for storing and delivering the shadows burdens as a result.

## 6 CONCLUSION

Two techniques of $(k, \infty)$-EVCS were presented in this paper. The first method implements the $(k, \infty)$ threshold by using the $(2, 2)$ and $(2, \infty)$ methods, while the second approach utilizes MVCS and a series of EVCSs with thresholds of $(1, \infty), \cdots, (k-1, \infty)$ to reduce the shadow size. Further, a more secure MXVCS is developed to

enhance the visual quality. Experiments and comparisons were demonstrated to show the effectiveness and advantages of the proposed techniques.

## APPENDICES

## A BASE MATRICES FOR THE IMPROVED SCHEME

Base matrices of related VCSs are given as follows.

$$B^0_{(2,2)} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}, B^1_{(2,2)} = \begin{bmatrix} 10 \\ 01 \end{bmatrix}, \quad B^0_{(3,3)} = \begin{bmatrix} 0110 \\ 0101 \\ 0011 \end{bmatrix}, B^1_{(3,3)} = \begin{bmatrix} 1100 \\ 1010 \\ 1001 \end{bmatrix},$$

$$B^0_{(2,4)} = \begin{bmatrix} 1000 \\ 1000 \\ 1000 \\ 1000 \end{bmatrix}, B^1_{(2,4)} = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix}, \quad B^0_{(3,4)} = \begin{bmatrix} 011100 \\ 101100 \\ 110100 \\ 111000 \end{bmatrix}, B^1_{(3,4)} = \begin{bmatrix} 100011 \\ 010011 \\ 001011 \\ 000111 \end{bmatrix},$$

$$B^0_{(4,4)} = \begin{bmatrix} 01110001 \\ 01001101 \\ 00101011 \\ 00010111 \end{bmatrix}, \quad B^1_{(4,4)} = \begin{bmatrix} 10001110 \\ 01001101 \\ 00101011 \\ 00010111 \end{bmatrix},$$

$$B^0_{(5,5)} = \begin{bmatrix} 0111100000011110 \\ 0100011100011101 \\ 0010010011011011 \\ 0001001010110111 \\ 0000100101101111 \end{bmatrix}, \quad B^1_{(5,5)} = \begin{bmatrix} 1000011111100001 \\ 1011100011100010 \\ 1101101100100100 \\ 1110110101001000 \\ 1111011010010000 \end{bmatrix}.$$

Base matrices of related XVCSs are demonstrated below.

$$B^{X,0}_{(2,2)} = B^0_{(2,2)}, B^{X,1}_{(2,2)} = B^1_{(2,2)}, \quad B^{X,0}_{(3,3)} = B^0_{(3,3)}, B^{X,1}_{(3,3)} = B^1_{(3,3)},$$
$$B^{X,0}_{(4,4)} = B^0_{(4,4)}, B^{X,1}_{(4,4)} = B^1_{(4,4)}, \quad B^{X,0}_{(5,5)} = B^0_{(5,5)}, B^{X,1}_{(5,5)} = B^1_{(5,5)},$$

$$B^{X,0}_{(2,4)} = \begin{bmatrix} 101010000000 \\ 100000101000 \\ 001000100010 \\ 000010001010 \end{bmatrix}, \quad B^{X,1}_{(2,4)} = \begin{bmatrix} 101010000000 \\ 010000101000 \\ 000100010010 \\ 000001000101 \end{bmatrix},$$

$$B^{X,0}_{(3,4)} = \begin{bmatrix} 0110011001100000 \\ 0101010100000110 \\ 0011000001010101 \\ 0000001100110011 \end{bmatrix}, \quad B^{X,1}_{(3,4)} = \begin{bmatrix} 1001100110010000 \\ 0101010100001001 \\ 0011000001010101 \\ 0000001100110011 \end{bmatrix}.$$

Base matrices of related NVCSs are provided as follows.

$$\tilde{B}^0_{(1,2)} = \begin{bmatrix} 10 \\ 01 \end{bmatrix}, \tilde{B}^1_{(1,2)} = \begin{bmatrix} 11 \\ 11 \end{bmatrix}, \qquad \tilde{B}^0_{(1,3)} = \begin{bmatrix} 011 \\ 101 \\ 110 \end{bmatrix}, \tilde{B}^1_{(1,3)} = \begin{bmatrix} 111 \\ 111 \\ 111 \end{bmatrix},$$

$$\tilde{B}^0_{(2,3)} = \begin{bmatrix} 101011 \\ 101110 \\ 111010 \end{bmatrix}, \tilde{B}^1_{(2,3)} = \begin{bmatrix} 101011 \\ 011110 \\ 110101 \end{bmatrix}, \quad \tilde{B}^0_{(1,4)} = \begin{bmatrix} 0111 \\ 1011 \\ 1101 \\ 1110 \end{bmatrix}, \tilde{B}^1_{(1,4)} = \begin{bmatrix} 1111 \\ 1111 \\ 1111 \\ 1111 \end{bmatrix},$$

$$\tilde{B}^0_{(2,4)} = \begin{bmatrix} 101010111111 \\ 101111101011 \\ 111011101110 \\ 111110111010 \end{bmatrix}, \qquad \tilde{B}^1_{(2,4)} = \begin{bmatrix} 101010111111 \\ 011111101011 \\ 110111011110 \\ 111101110101 \end{bmatrix},$$

$$\tilde{B}^0_{(3,4)} = \begin{bmatrix} 0110011001101111 \\ 0101010111110110 \\ 0011111101010101 \\ 1111001100110011 \end{bmatrix}, \qquad \tilde{B}^1_{(3,4)} = \begin{bmatrix} 1001100110011111 \\ 0101010111111001 \\ 0011111101010101 \\ 1111001100110011 \end{bmatrix},$$

$$\tilde{B}^0_{(1,5)} = \begin{bmatrix} 01111 \\ 10111 \\ 11011 \\ 11101 \\ 11110 \end{bmatrix}, \qquad \tilde{B}^1_{(1,5)} = \begin{bmatrix} 11111 \\ 11111 \\ 11111 \\ 11111 \\ 11111 \end{bmatrix},$$

$$\tilde{B}^0_{(2,5)} = \begin{bmatrix} 01010101111111111111 \\ 01111111010101111111 \\ 11011111011111010111 \\ 11110111110111011101 \\ 11111101111101110101 \end{bmatrix}, \tilde{B}^1_{(2,5)} = \begin{bmatrix} 01010101111111111111 \\ 10111111010101111111 \\ 11101111101111010111 \\ 11111011111011101101 \\ 11111110111110111010 \end{bmatrix},$$

$$\tilde{B}^0_{(3,5)} = \begin{bmatrix} 01100110011001100110011011111111111111111 \\ 01010101010111111111111110110011001101111 \\ 00111111111101010101111101010101111110110 \\ 11110011111100111111010100111111010101 \\ 11111111001111110011001111110011001100111 \end{bmatrix},$$

$$\tilde{B}^1_{(3,5)} = \begin{bmatrix} 11001100110011001100110011111111111111111 \\ 10101010101011111111111111001100110011111 \\ 10011111111110101010111111010101011111100 \\ 11111001111110011111101010011111110101010 \\ 11111111100111111001100111111100110011001 \end{bmatrix},$$

$$\tilde{B}^0_{(4,5)} = \begin{bmatrix} 01110001011100010111000101110001111111111 \\ 01001101010011010100110111111111101110001 \\ 00101011001010111111111101110000101001101 \\ 00010111111111110010101100101011001010111 \\ 11111111000101110001011100010111000100111 \end{bmatrix},$$

$$\tilde{B}^{1}_{(4,5)} = \begin{bmatrix} 10001110100011101000111010001110111111111 \\ 01001101010011010100110111111111110001110 \\ 00101011001010111111111101001101010011101 \\ 00010111111111111001010110010101100101011 \\ 11111111000101110001011100010111000010111 \end{bmatrix}.$$

Base matrices of related NXVCSs are illustrated below.

$$\tilde{B}^{X,0}_{(1,2)} = \begin{bmatrix} *0 \\ 0* \end{bmatrix}, \tilde{B}^{X,1}_{(1,2)} = \begin{bmatrix} 1* \\ *1 \end{bmatrix}, \quad \tilde{B}^{X,0}_{(1,4)} = \begin{bmatrix} 0*** \\ *0** \\ **0* \\ ***0 \end{bmatrix}, \tilde{B}^{1}_{(1,4)} = \begin{bmatrix} 1*** \\ *1** \\ **1* \\ ***1 \end{bmatrix},$$

$$\tilde{B}^{X,0}_{(2,3)} = \begin{bmatrix} 1010 \\ 1010 \\ 1010 \end{bmatrix}, \tilde{B}^{X,1}_{(2,3)} = \begin{bmatrix} 1010 \\ 0110 \\ 0101 \end{bmatrix}, \quad \tilde{B}^{X,0}_{(1,5)} = \begin{bmatrix} 0**** \\ *0*** \\ **0** \\ ***0* \\ ****0 \end{bmatrix}, \tilde{B}^{1}_{(1,5)} = \begin{bmatrix} 1**** \\ *1*** \\ **1** \\ ***1* \\ ****1 \end{bmatrix},$$

$$\tilde{B}^{X,0}_{(2,4)} = \begin{bmatrix} 0101 \\ 0101 \\ 0101 \\ 0101 \end{bmatrix}, \tilde{B}^{1}_{(2,4)} = \begin{bmatrix} 1010 \\ 0110 \\ 0101 \\ 1001 \end{bmatrix}, \quad \tilde{B}^{X,0}_{(2,5)} = \begin{bmatrix} 0101** \\ 010101 \\ 01**01 \\ 010101 \\ **0101 \end{bmatrix}, \tilde{B}^{X,1}_{(2,5)} = \begin{bmatrix} 0101** \\ 100101 \\ 10**10 \\ 011010 \\ **1001 \end{bmatrix},$$

$$\tilde{B}^{X,0}_{(1,3)} = \begin{bmatrix} 0** \\ *0* \\ **0 \end{bmatrix}, \tilde{B}^{X,1}_{(1,3)} = \begin{bmatrix} 1** \\ *1* \\ **1 \end{bmatrix}, \quad \tilde{B}^{X,0}_{(3,4)} = \begin{bmatrix} 01100110 \\ 01010110 \\ 00110101 \\ 00110011 \end{bmatrix}, \tilde{B}^{1}_{(3,4)} = \begin{bmatrix} 11001100 \\ 10101100 \\ 10011010 \\ 10011001 \end{bmatrix},$$

$$\tilde{B}^{X,0}_{(3,5)} = \begin{bmatrix} 011001100110 \\ 010101010110 \\ 001101100101 \\ 001101010011 \\ 010100110101 \end{bmatrix}, \quad \tilde{B}^{X,1}_{(3,5)} = \begin{bmatrix} 110011001100 \\ 101010101100 \\ 100111001010 \\ 100110101001 \\ 101010011010 \end{bmatrix},$$

$$\tilde{B}^{X,0}_{(4,5)} = \begin{bmatrix} 011100010111000101110001 \\ 010011010100110101110001 \\ 001010110100110101001101 \\ 000101110010101100101011 \\ 000101110001011100010111 \end{bmatrix}, \tilde{B}^{X,1}_{(4,5)} = \begin{bmatrix} 100011101000111010001110 \\ 010011010100110110001110 \\ 001010110100110101001101 \\ 000101110010101100101011 \\ 000101110001011100010111 \end{bmatrix}.$$

## B THEORETICAL ESTIMATION OF CONTRAST FOR THE TWO SCHEMES

Let $s$ and $s'$ be the secret pixel and recovered secret pixel. We denote $L^{s'}_{s=0}$ (resp. $L^{s'}_{s=1}$) as the average brightness (i.e., probability to be white) of the recovered result $s'$ when the secret pixel is $s = 0$ (i.e., $s = 1$). Hence, the contrast can be evaluated by $\alpha = L^{s'}_{s=0} - L^{s'}_{s=1}$.

For the proposed $(2, \infty)$ scheme, when $s = 1$, the 2 shared pixels are randomly generated. The average brightness of the recovered result by stacking 2 random shared pixels is computed as $L^{s'}_{s=1} = (1/2) \times (1/2) = 1/4$. When $s = 0$, the 2 shadow pixels are the same. We achieve $L^{s'}_{s=0} = 1/2$. Thus, the contrast of the proposed $(2, \infty)$ scheme is calculated by $\alpha = 1/2 - 1/4 = 1/4$. Similarly, when employing the XOR decryption, we have $L^{s'}_{s=1} = 1/2$ (because

the 2 shadow pixels are random) and $L^{s'}_{s=0} = 1$ (since the 2 shared pixels are the same). Consequently, the contrast of the proposed $(2, \infty)$ scheme using XOR operation is achieved as $\alpha = 1 - 1/2 = 1/2$.

For the proposed $(k, \infty)$ method, the base matrices of $(2, 2)$ VCS [13] are used, as given in Appendix A. The secret recovery of the proposed $(k, \infty)$ scheme can refer to Fig. 2, where 2 shared pixels from the $(2, \infty)$ scheme and $(k-2)$ shadow pixels belonging to the $(2, 2)$ methods are involved. When $s = 1$ (resp. $s = 0$), the average brightness of the recovered result by stacking these $k$ shared pixels is calculated by $L^{s'}_{s=1} = (1/2)^k$ (resp. $L^{s'}_{s=0} = (1/2)^{k-1}$). As a result, the contrast is obtained by $\alpha = (1/2)^{k-1} - (1/2)^k = (1/2)^{k-1}$. For the proposed technique using XOR decryption, the same base matrices are used for the $(2, 2)$ method. If $s = 1$ (resp. $s = 0$), the average brightness of the recovered result by XOR-ing the $k$ shadow pixels is achieved as $L^{s'}_{s=1} = (1/2) \times 0 + (1/2) \times (1/2) = 1/4$ (resp. $L^{s'}_{s=0} = (1/2) \times 1 + (1/2) \times (1/2) = 3/4$). The contrast using XOR decoding is calculated by $\alpha = 3/4 - 1/4 = 1/2$.

For the improved method using stacking decryption, the base matrices of NVCS and VCS are achieved from Chen's approach [3] and Naor and Shamir's method [13], respectively. When using XOR decryption, the base matrices of NXVCS are derived based on Construction 2 of [27] using our proposed partition algorithm. While the base matrices of monotone XVCS are constructed from [19]. All base matrices are illustrated in Appendix A. As different base matrices would introduce different contrasts, the theoretical value is evaluated case by case. Generally, for any $k$ participants, suppose the first $c$ ($1 \le c \le k$) participants come from the same generation $G_u$ and the remaining $(k - c)$ participants belong to future generations $G_{u+1}$, $G_{u+2}$, $\cdots$. Theoretical estimation is provided as follows. When $c = k$, all the $k$ participants are from the same generation, the contrast of secret $s$ is directly evaluated based on MVCS. When $c < k$, secret $s$ is recovered based on the two shared pixels $d^u_c$ and $\tilde{d}^u_c$ from the $(2, 2)$ method, where $d^u_c$ is decrypted by the first $c$ participants via MVCS and $\tilde{d}^u_c$ is reconstructed by the remaining $(k - c)$ participants via the $(k - c, \infty)$ scheme. Once the average brightness values of $d^u_c$ and $\tilde{d}^u_c$ are obtained, we can derive the contrast of $s$. The contrast using XOR decryption can be estimated by the same way. Remember that, we might have different contrasts due to different values of $c$ are utilized.

We demonstrate the contrast estimation via the improved $(2, \infty)$ scheme using stacking operation. Suppose the number of participants in each generation is 2. The $(1, 2)$ NVCS and $(2, 2)$ VCS are adopted to build a $(1, 2, 2)$ MVCS. (1) When 2 participants come from the same generation, secret $s$ is directly decrypted from the $(1, 2, 2)$ MVCS. The average brightness of the recovered result is $L^{s'}_{s=1} = 0$ (resp. $L^{s'}_{s=0} = 1/4$) when $s = 1$ (resp. $s = 0$). Hence, the contrast of $s$ is computed as $\alpha = 1/4$. (2) When 1 participant comes from $G_u$ and 1 user belongs to future generation, we can obtain $d^u_1$ from the $(1, 2, 2)$ MVCS and $\tilde{d}^u_1$ from the $(1, \infty)$ scheme. For $d^u_1$, according to the $(1, 2, 2)$ MVCS, the average brightness of the reconstructed result $d^{u\prime}_1$ by one shared pixel is $L^{d^{u\prime}_1}_{d^u_1=1} = 1/4$ (resp. $L^{d^{u\prime}_1}_{d^u_1=0} = 1/2$) when $d^u_1 = 1$ (resp. $d^u_1 = 0$). For $\tilde{d}^u_1$, each shared pixel from the $(1, \infty)$ scheme is the same as $\tilde{d}^u_1$. We have $L^{\tilde{d}^{u\prime}_1}_{\tilde{d}^u_1=1} = 0$ (resp. $L^{\tilde{d}^{u\prime}_1}_{\tilde{d}^u_1=0} = 1$) when $\tilde{d}^u_1 = 1$ (resp. $\tilde{d}^u_1 = 0$). Moreover, $d^u_1$ and $\tilde{d}^u_1$ are derived from the $(2, 2)$ scheme based on $s$. When $s = 0$ (resp. $s = 1$), we achieve $\tilde{d}^u_1 = d^u_1 = 0$ or $\tilde{d}^u_1 = d^u_1 = 1$ (resp. $\tilde{d}^u_1 = 1$, $d^u_1 = 0$ or $\tilde{d}^u_1 = 0$, $d^u_1 = 1$). Therefore, when $s = 0$, the average brightness of the result by stacking the recovered $d^{u\prime}_1$ and $\tilde{d}^{u\prime}_1$ is $L^{s'}_{s=0} = (1/2) \times L^{\tilde{d}^{u\prime}_1}_{\tilde{d}^u_1=0} \times L^{d^{u\prime}_1}_{d^u_1=0} + (1/2) \times L^{\tilde{d}^{u\prime}_1}_{\tilde{d}^u_1=1} \times L^{d^{u\prime}_1}_{d^u_1=1} = 1/4$. When $s = 1$, we achieve $L^{s'}_{s=1} = (1/2) \times L^{\tilde{d}^{u\prime}_1}_{\tilde{d}^u_1=1} \times L^{d^{u\prime}_1}_{d^u_1=0} + (1/2) \times L^{\tilde{d}^{u\prime}_1}_{\tilde{d}^u_1=0} \times L^{d^{u\prime}_1}_{d^u_1=1} = 1/8$. Consequently, the contrast is computed as $\alpha = 1/4 - 1/8 = 1/8$.

## ACKNOWLEDGMENT

This work was partially supported by National Key Research and Development Program of China (Grant Nos. 2022YFB3103100 and 2022YFE0116800), National Natural Science Foundation of China (Grant Nos. 61972179 and

62102101), Guangdong Key Laboratory of Data Security and Privacy Preserving (Grant No. 2023B1212060036), Doctoral Scientific Research Foundation of Guangdong Polytechnic Normal University (Grant No. 2021SDKYA101), and National Science and Technology Council (Grant No. 112-2221-E-259-007-MY2).

## REFERENCES

[1] Charles Asmuth and John Bloom. 1983. A modular approach to key safeguarding. *IEEE Transactions on Information Theory* 29, 2 (1983), 208–210.
[2] Tzung-Her Chen and Kai-Hsiang Tsao. 2009. Visual secret sharing by random grids revisited. *Pattern recognition* 42, 9 (2009), 2203–2217.
[3] Yu-Chi Chen. 2017. Fully incrementing visual cryptography from a succinct non-monotonic structure. *IEEE Transactions on Information Forensics and Security* 12, 5 (2017), 1082–1091.
[4] Paolo D'Arco, Roberto De Prisco, and Alfredo De Santis. 2021. Secret sharing schemes for infinite sets of participants: A new design technique. *Theoretical Computer Science* 859 (2021), 149–161.
[5] Zhongyun Hua, Xingyu Liu, Yifeng Zheng, Shuang Yi, and Yushu Zhang. 2024. Reversible data hiding over encrypted images via preprocessing-free matrix secret sharing. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 3 (2024), 1799–1814.
[6] Xingxing Jia, Daoshun Wang, Daxin Nie, and Chaoyang Zhang. 2018. Collaborative visual cryptography schemes. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 5 (2018), 1056–1070.
[7] Ilan Komargodski, Moni Naor, and Eylon Yogev. 2017. How to share a secret, infinitely. *IEEE Transactions on Information Theory* 64, 6 (2017), 4179–4190.
[8] Peng Li, Liping Yin, Jianfeng Ma, and Hongtao Wang. 2022. XOR-based visual cryptography scheme with essential shadows. *Journal of Visual Communication and Image Representation* 85 (2022), 103513.
[9] Yanxiao Liu, Chingnung Yang, and Qindong Sun. 2021. Thresholds based image extraction schemes in big data environment in intelligent traffic management. *IEEE Transactions on Intelligent Transportation Systems* 22, 7 (2021), 3952–3960.
[10] Zuquan Liu, Guopu Zhu, Feng Ding, Xiangyang Luo, Sam Kwong, and Peng Li. 2022. Contrast-Enhanced Color Visual Cryptography for (k, n) Threshold Schemes. *ACM Transactions on Multimedia Computing, Communications and Applications* 18, 3s (2022), 1–16.
[11] Zuquan Liu, Guopu Zhu, Yuan-Gen Wang, Jianquan Yang, and Sam Kwong. 2020. A Novel (t, s, k, n)-Threshold Visual Secret Sharing Scheme Based on Access Structure Partition. *ACM Transactions on Multimedia Computing, Communications, and Applications* 16, 4 (2020), 1–21.
[12] Keju Meng, Fuyou Miao, Yan Xiong, and Chin-Chen Chang. 2021. A reversible extended secret image sharing scheme based on Chinese remainder theorem. *Signal Processing: Image Communication* 95 (2021), 116221.
[13] M. Naor and A. Shamir. 1995. Visual cryptography. *Lecture Notes in Computer Science* 950, 1 (1995), 1–12.
[14] Pauline Puteaux, Félix Yriarte, and William Puech. 2023. A Secret JPEG Image Sharing Method Over GF ($2^M$) Galois Fields. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 6 (2023), 3030 – 3042.
[15] Chuan Qin, Chanyu Jiang, Qun Mo, Heng Yao, and Chin-Chen Chang. 2021. Reversible Data Hiding in Encrypted Image via Secret Sharing Based on GF (p) and GF ($2^8$). *IEEE Transactions on Circuits and Systems for Video Technology* 32, 4 (2021), 1928–1941.
[16] A. Shamir. 1979. How to share a secret. *Commun. ACM* 22, 11 (1979), 612–613.
[17] Gang Shen, Feng Liu, Zhengxin Fu, and Bin Yu. 2017. Perfect contrast XOR-based visual cryptography schemes via linear algebra. *Designs, Codes and Cryptography* 85, 1 (2017), 15–37.
[18] Shivendra Shivani and Suneeta Agarwal. 2016. Progressive visual cryptography with unexpanded meaningful shares. *ACM Transactions on Multimedia Computing, Communications, and Applications* 12, 4 (2016), 1–24.
[19] Shyong Jian Shyu. 2018. XOR-based Visual Cryptographic Schemes with Monotonously Increasing and Flawless Reconstruction Properties. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 9 (2018), 2397–2401.
[20] Shyong Jian Shyu and Kun Chen. 2010. Visual multiple-secret sharing by circle random grids. *SIAM Journal on Imaging Sciences* 3, 4 (2010), 926–953.
[21] Shyong Jian Shyu and Kun Chen. 2011. Visual multiple secret sharing based upon turning and flipping. *Information Sciences* 181, 15 (2011), 3246–3266.
[22] Shyong Jian Shyu and Ming Chiang Chen. 2011. Optimum pixel expansions for threshold visual secret sharing schemes. *IEEE Transactions on Information Forensics and Security* 6, 3 (2011), 960–969.
[23] Shyong Jian Shyu and Hung-Wei Jiang. 2013. General constructions for threshold multiple-secret visual cryptographic schemes. *IEEE Transactions on Information Forensics and Security* 8, 5 (2013), 733–743.
[24] Sophie CC Sun, Yongkang Zhao, Fang-Wei Fu, and Yawei Ren. 2023. Improved random grid-based cheating prevention visual cryptography using Latin square. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 2s (2023), 1–21.
[25] Daoshun Wang, Tao Song, Lin Dong, and Ching-Nung Yang. 2013. Optimal Contrast Grayscale Visual Cryptography Schemes With Reversing. *IEEE Transactions on Information Forensics and Security* 8, 12 (2013), 2059–2072.

[26] Jonathan Weir, Weiqi Yan, and Mohan S Kankanhalli. 2012. Image hatching for visual cryptography. *ACM Transactions on Multimedia Computing, Communications, and Applications* 8, 2S (2012), 1–15.
[27] Xiaotian Wu, Na An, and Zishuo Xu. 2023. Sharing multiple secrets in XOR-based visual cryptography by non-monotonic threshold property. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 1 (2023), 88–103.
[28] Xiaotian Wu, Jia Fang, and Wei Qi Yan. 2023. Contrast optimization for size invariant visual cryptography scheme. *IEEE Transactions on Image Processing* 32 (2023), 2174–2189.
[29] Xiaotian Wu and Xinjie Feng. 2024. Size Invariant Visual Cryptography Schemes with Evolving Threshold Access Structures. *IEEE Transactions on Multimedia* 26 (2024), 1488–1503.
[30] Xiaotian Wu, Zishuo Xu, and WeiQi Yan. 2023. Sharing Visual Secrets among Multiple Groups with Enhanced Performance. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 11 (2023), 6503–6518.
[31] Lizhi Xiong, Xinwei Zhong, Ching-Nung Yang, and Xiao Han. 2021. Transform domain-based invertible and lossless secret image sharing with authentication. *IEEE Transactions on Information Forensics and Security* 16 (2021), 2912–2925.
[32] Xuehu Yan, Longlong Li, Lei Sun, Jia Chen, and Shudong Wang. 2023. Fake and dishonest participant immune secret image sharing. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 4 (2023), 1–26.
[33] Xuehu Yan, Xin Liu, and Ching-Nung Yang. 2018. An enhanced threshold visual secret sharing based on random grids. *Journal of Real-Time Image Processing* 14, 1 (2018), 61–73.
[34] Xuehu Yan, Yuliang Lu, Ching-nung Yang, Xinpeng Zhang, and Shudong Wang. 2021. A common method of share authentication in image secret sharing. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 7 (2021), 2896–2908.
[35] Ching-Nung Yang. 2004. New visual secret sharing schemes using probabilistic method. *Pattern Recognition Letters* 25, 4 (2004), 486–494.
[36] Ching-Nung Yang and Dao-Shun Wang. 2014. Property analysis of XOR-based visual cryptography. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 2 (2014), 189–197.
[37] Chunqiang Yu, Xianquan Zhang, Chuan Qin, and Zhenjun Tang. 2023. Reversible data hiding in encrypted images with secret sharing and hybrid coding. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 11 (2023), 6443–6458.