On the Design of Distributed Multi-User Secret Image Sharing for General Access Structures

Xiaotian Wu, *Member, IEEE*, Yuyang Xiong, Bing Chen, Ching-Nung Yang, *Senior Member, IEEE*, WeiQi Yan, *Senior Member, IEEE*, Qing-Yu Peng

***Abstract*—In this article, a secret image sharing (SIS) scheme for general access structures (GAS) is designed for distributed multi-user scenario. In the proposed distributed multi-user SIS (DM-SIS), multiple secret images are encoded into shadows which are then distributed to the corresponding storage nodes of a network. By collecting the shadows from nodes, each user is capable of decrypting the corresponding secret image. Fundamentally, we utilize an invertible target matrix, which is initially obtained from the GAS and a base matrix with Vandermonte coordinates, to construct shadows. To deal with the case of non-invertible target matrix, three matrix-adjusting procedures are further introduced. Theoretical analysis, numerical examples, and experiments are provided to verify the feasibility of the proposed technique. When compared to previous methods, the proposed approach can implement GAS sharing strategy in distributed multi-user environment. Meantime, significant improvements on storage overhead and sharing capacity are also achieved.**

***Index Terms*—Secret sharing, secret image sharing, distributed multi-user, general access structure.**

I. Introduction

With the exponential rise of multimedia infrastructures, bulk amounts of image data are captured, stored, and processed. Regarding data security, sensitive images need to be protected in these data-intensive ecosystems. To secure confidential images, techniques such as steganography [1], [2], reversible data hiding [3]–[5], and encryption [6] were usually adopted.

Secret image sharing (SIS) [7] and visual secret sharing (VSS) [8]–[11], both derived from secret sharing [12], [13], have also emerged as effective methodologies for visual data protection. The well-known (k, n)-SIS splits a secret image into n random-looking images, called shadows or shares, in such a way that any collection having k or more shadows can recover the secret. Whereas, any subset with $(k-1)$ or fewer shadows cannot attain any clue about the secret. With the pioneer work in [7], various SIS techniques with different functionalities, such as SIS with steganography and authentication [14], scalable SIS [15], essential SIS [16], [17], SIS with privileged sets [18], SIS for intelligent traffic management [19], SIS for image authentication [20], transform domain-based SIS [21], and SIS against fake and dishonest participants [22], were developed. Another essential component of secret sharing for safeguarding sensitive images is VSS. Differing from SIS, VSS encodes a binary secret into n shadows such that any k or more shadows are capable of visually recovering the secret without using computational devices. Easy-decoding property is provided. With the fundamentals of VSS proposed in [23], numerous VSS schemes were investigated. Most schemes focused on solving some critical problems, such like minimizing pixel expansion [24], improving image quality [25], and providing a flexible sharing strategy [11].

Most SIS methods consider the scenario for sharing one secret, where a dealer has direct and secure channels to all the users. Once the shadows are produced, they are readily delivered to the users by using these channels. However, in many real-world applications, the direct and secure channel from the dealer to each user might be not available. More suitably, the communication between dealer and users can be accomplished via several intermediate nodes. Based on this assumption, the distributed multi-user secret sharing protocol (DSSP) [26] was introduced. In DSSP scenario, multiple secrets are encrypted into shadows. The dealer is considered as a master node that transmits the shadows to n storage nodes of a network. Each user has independent access to a specific subset of k nodes so that he can collect k shadows from nodes to recover a specific secret. The three methods in [26] mainly depend on Shamir's secret sharing. By randomly predetermining the values of some shadows, the coefficients of polynomials are calculated and then reused to compute the remaining shadows. When the shadows are significantly overlapped, the storage overhead is reduced as a result. Improved DSSP was discussed in [27], where a circulant matrix is adopted for realizing a $(k, k+1)$ scheme. The $(k, k+1)$ method serves as a central building block to constitute the (k, n) scheme. Their construction is a step-by-step approach. Given any $t \geq k+1$, a $(k, t+1)$ scheme is obtained from the (k, t) approach. By repeating this procedure for $(n-k-1)$ times, the (k, n) scheme is built. However, previous DSSPs [26], [27] are not designed for images and suffer from the drawbacks, such as restricted

This work was partially supported by National Key R&D Program of China (Grant Nos. 2022YFB3103100 and 2022YFE0116800), National Natural Science Foundation of China (Grant No. 62102101), Guangdong Key Laboratory of Data Security and Privacy Preserving (Grant No. 2023B1212060036), the Doctoral Scientific Research Foundation of Guangdong Polytechnic Normal University (Grant No. 2025SDKYA004), and National Science and Technology Council (Grant No. 112-2221-E-259-007-MY2). (Corresponding author: Bing Chen)

X. Wu and Y. Xiong are with the College of Cyber Security and Guangdong Key Laboratory of Data Security and Privacy Preserving, Jinan University, Guangzhou, China. E-mail: wxt.sysu@gmail.com

B. Chen is with the School of Cyber Security, Guangdong Polytechnic Normal University, Guangzhou, China. E-mail: chenbing@gpnu.edu.cn

C.-N. Yang is with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. E-mail: cnyang@gms.ndhu.edu.tw

W. Q. Yan is with the Department of Computer Science, Auckland University of Technology, Auckland, New Zealand. E-mail: wyan@aut.ac.nz

Q.-Y. Peng is with the Department of Computer Science and Sino-French Joint Laboratory for Astrometry, Dynamics and Space Science, Jinan University, Guangzhou, China. E-mail: tpengqy@jnu.edu.cn

access structure, high storage overhead, unsatisfactory sharing capacity, and indirect shadow construction.

In this paper, we investigate the distributed multi-user SIS (DM-SIS) for general access structures (GAS). Sensitive images can be protected in distributed multi-user environment. Meanwhile, the weaknesses in previous methods are improved. Contribution of this paper is summarized below.

- A DM-SIS scheme suitable for GAS is presented. Distributed multi-user scenario is extended to image domain for safeguarding multiple secrets among different users. When generating shadows, a base matrix with Vandermonte coordinates is constructed. Based on the GAS, an initial target matrix is obtained by linearly combining the vectors of base matrix. Once an invertible target matrix is achieved, the shadows are constructed accordingly.
- Three matrix-adjusting procedures for constituting the proposed DM-SIS are introduced. These algorithms are utilized to alter the initial target matrix when it is not invertible. The main concept for the modification is to randomly append additional column vectors or/and row vectors to the initial target matrix.
- Experiments and comparisons are conducted to show the effectiveness and benefits of the proposed approach. When compared with existing DSSPs that are confined to (k, n) threshold, the proposed technique is suitable for GAS, which means complicated sharing strategies can be implemented. Improved storage overhead and enhanced sharing capacity are also provided. Furthermore, we can directly create shadows in an efficient manner.

The organization of this paper is given as follows. Section II provides traditional SIS and the definition of GAS. The proposed technique is formulated in Section III. Section IV offers theoretical analysis and numerical examples. Experimental results and comparisons are shown in Section V. Concluding remarks are given in Section VI.

II. Preliminaries

The technique of traditional SIS is described in this section, as well as the basic concept of GAS.

A. Traditional Secret Image Sharing

The (k, n) SIS utilizes Shamir's method [12] to conceal k secret pixels into the k coefficients of a $(k-1)$-degree polynomial $f(x) = (a_0 + \sum_{j=1}^{k-1} a_j x^j) \mod p$ where p is a prime. The k coefficients are replaced by k secret pixels. By choosing n non-zero $x_1, \cdots, x_n$, n shadow pixels $f(x_1), \cdots, f(x_n)$ are constructed and delivered to n participants. Any k shadow pixels can recover the $(k-1)$-degree polynomial via Lagrange interpolation. So that k secret pixels are decrypted. Usually, the secret image should be permuted [7] in a prior to enhance the security. Moreover, $p = 251$ is commonly adopted for calculations. As a result, the pixels within $[251, 255]$ must be truncated to 250, and this will lead to image distortion. To avoid distortion, finite field $GF(2^8)$ is usually employed [28].

B. General Access Structure

Suppose $\Omega = \{1, 2, \cdots, n\}$ is a set having n elements called participants and denote 2^Ω as the set of all subsets of Ω. Definition 1 lists the concept of GAS for conventional SIS schemes. Additionally, define Γ_0 as the collection consisting of all the minimal qualified subsets:

$$\Gamma_0 = \{B \in \Gamma^{Qual} : B' \notin \Gamma^{Qual}, \text{ for all } B' \subset B\}. \quad (1)$$

A participant $i \in \Omega$ is essential if there exists a set B' such that $B' \cup \{i\} \in \Gamma^{Qual}$ but $B' \notin \Gamma^{Qual}$. In the case where Γ^{Qual} is monotone increasing, Γ^{Forb} is monotone decreasing, and $\Gamma^{Qual} \cup \Gamma^{Forb} = 2^\Omega$, the GAS is said to be strong, and Γ_0 is termed a basis of the GAS.

Definition 1. *(GAS) Let $\Gamma^{Qual} \in 2^\Omega$ and $\Gamma^{Forb} \in 2^\Omega$, where $\Gamma^{Qual} \cap \Gamma^{Forb} = \emptyset$. Members of Γ^{Qual} are defined as qualified sets and members of Γ^{Forb} are classified as forbidden sets. The pair $(\Gamma^{Qual}, \Gamma^{Forb})$ is called the general access structure.*

In contrast to traditional SIS, our method is designed for distributed multi-user scenario. In our environment, there are multiple users and several storage nodes. Each shadow is distributed to a corresponding storage node, and every user recovers a specific secret from certain nodes. A qualified subset, which indicates a collection of nodes that can decode a secret, is correlated with a user. The access structure in our scheme is somewhat different from the conventional one [29]. The limitations, "the collections of qualified subsets Γ_{Qual} is monotone increasing" and "the collections of forbidden sets Γ_{Forb} is monotone decreasing", are removed. The access structure is not strong and the nodes are essential.

III. The Proposed Scheme

A. Analysis and Motivation

Most SIS methods focus on the scenario in which the dealer has direct and secure communication channels with all the users. Meantime, only one secret image is shared among all users. These schemes are not suitable for the distributed multi-user environment. When designing a distributed multi-user SIS, the following aspects should be carefully considered.

- **Access structure.** Previous DSSPs and most SISs are constrained to (k, n). Especially for the DSSPs in [26] [27], each secret is encoded into k shadows which are distributed to k storage nodes. Complex sharing strategy cannot be realized. A scheme for GAS is preferred.
- **Storage overhead.** The storage overhead for each node is expected to be as low as possible. However, a node in [26] [27] might store multiple shadows. Moreover, the number of shadows assigned to a node would be remarkably large for some cases.
- **Sharing capacity.** More secrets are expected to be encrypted within the shadows. Existing DSSPs [26] [27] encode only $m = \binom{n}{k}$ secrets for the (k, n) threshold. Sharing capacity is anticipated to be higher.
- **Way of shadow construction.** The shadows of the (k, n) scheme in [27] are generated in a step-by-step manner. A $(k, t+1)$ case is constituted from a smaller (k, t) method. Such a construction is repeated for $(n-k-1)$

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3603223

times to accomplish the (k, n) scheme. Their step-by-step construction [27] is inefficient. A technique that can directly produce all the shadows is required.

We are motivated to design a DM-SIS such that it provides solutions to the four mentioned aspects. For access structure, not only (k, n) threshold but also GAS policy can be realized. Complicated sharing strategy is provided. To improve the storage overhead, our approach delivers one shadow for each storage node. The number of shadows for a node is dramatically decreased. To enhance the sharing capacity, the maximum number of encoded secrets is increased to $\sum_{i=k}^{n} \binom{n}{i}$, which is larger than existing DSSPs. For the way of producing shadows, differing from the DSSP [27] with a step-by-step construction, the proposed technique outputs all the shadows directly.

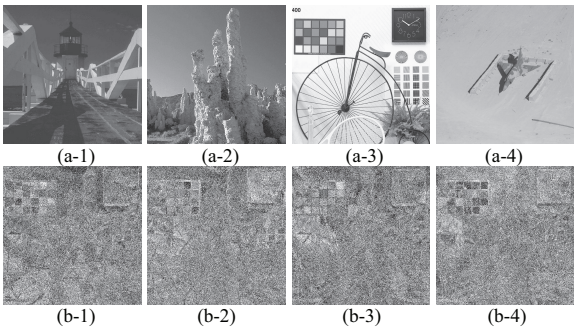

Fig. 1. A $(3, 4)$ scheme by using the method in [27]. (a) Four secret images, (b) four generated shadows.

In addition, not all secret sharing methods can be directly applied to images. There exists a security concern when utilizing the DSSP in [27] for protecting images. A $(3, 4)$ scheme by [27] is realized to illustrate the weakness, as shown in Fig. 1. To process 8-bit images, $q = 251$ is adopted. The circulant matrix M^{-1} for this experiment is given as

$$M^{-1} = \begin{bmatrix} 83 & 84 & 84 & 84 \\ 84 & 83 & 84 & 84 \\ 84 & 84 & 83 & 84 \\ 84 & 84 & 84 & 83 \end{bmatrix}. \tag{2}$$

Four secret images and four generated shadows are demonstrated in Figs. 1 (a) and (b), respectively. The secret information is disclosed on shadows. Directly applying a DSSP [27] to images would result in secret leakage. Their approach is not suitable for images. On the other hand, the proposed approach can address this security concern by the skillful design.

B. Overview

Transmission of shadows via trustworthy and safe channels is necessary for secret sharing. However, both dealer and users are typically connected via a large public network (e.g., the Internet) in many real-life situations. Consequently, a direct and secure channel from dealer to user might be not available. Distributed secret sharing takes into account the application scenario that uses indirect secure channels between dealer and users. The dealer can directly distribute shadows to all storage nodes through some error-free, reliable and secure links. Every user can access to a specific subset of storage nodes, called a qualified subset, with secure channels. Delivering shares allows indirect secure connections. As compared with conventional techniques, the methods using indirect connections would require more secure links. But the overhead of dealer would be reduced since the dealer only distributes the shadows to the nodes and will not interact with the users. It is still practical. As secure connections between dealer/users and storage nodes are utilized, the shadows can be securely distributed to the user for recovering the respective secret image. On the other hand, it might be beneficial for many real-world applications. Consider the Internet of Things (IoT) scenario, where the dealer assumes the role of an IoT gateway which manages several servers (i.e., storage nodes). Every device, acting as a user, has the ability to access and download resources from a subset of controlled servers.

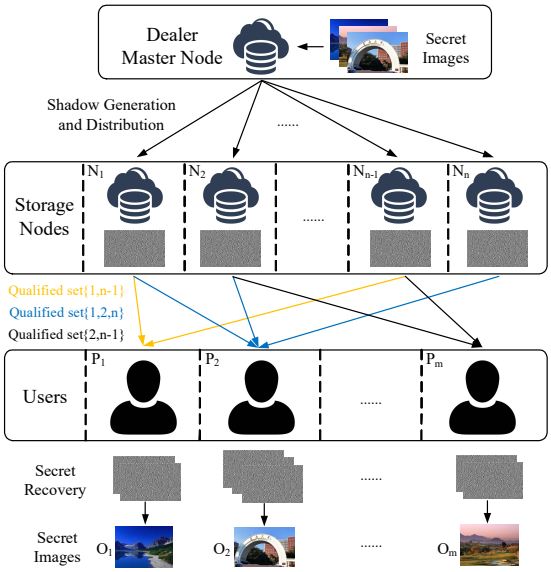

Fig. 2. Application scenario of the proposed DM-SIS.

TABLE I
NOTATIONS USED IN THIS PAPER.

Notation	Description
$P_1, \cdots, P_m$	m users
$O_1, \cdots, O_m$	m secret images
$\hat{O}_{m+1}, \cdots, \hat{O}_n$	$(n-m)$ random images
$N_1, \cdots, N_n$	n storage nodes of a network
$SH_1, \cdots, SH_n$	n shadows
$\sigma_1(\cdot), \cdots, \sigma_3(\cdot)$	three target-matrix-adjusting strategies
M_{nn}	a $(n \times n)$ base matrix whose row vectors have Vandermonde coordinates
$b_i^{(n)}$	a $(1 \times n)$ row vector with the i^{th} element being 1 and the remaining $(n-1)$ elements being 0
$f^{(n)}(x)$	a $(1 \times n)$ row vector comprised by $b_i^{(n)}$s
$e_i^{(n)}$	the i^{th} target vector having n elements
E	an initial target matrix comprised by target vectors
$\tilde{E}$	an invertible target matrix
$S_j^{(m)}$	the j^{th} secret matrix having m elements
H_j	the j^{th} temporary shared matrix
A_{mn}	a $(m \times n)$ coefficient matrix for generating E
C_{mn}	an indication matrix to record non-zero elements of A_{mn}
U	a collection having different row matrices
V	a collection having different column matrices
$R(\cdot)$	a function outputs a random integer within $[1, 255]$
$det(\cdot)$	the determinant of the input matrix

In the proposed system, there exists a trusty party called

dealer, n storage nodes of a network, m users, and m secret images. Let $\mathcal{P} = \{P_1, \cdots, P_m\}$ be the set of users and let $\mathcal{N} = \{N_1, \cdots, N_n\}$ be the collection of nodes. Our method is separated into two phases: shadow construction and image recovery. In shadow construction, m secret images $O_1, \cdots, O_m$ are encrypted into n shadows $SH_1, \cdots, SH_n$ which are distributed to n nodes $N_1, \cdots, N_n$ of a network. In image recovery, any user P_i, $1 \leq i \leq m$, can disclose a secret image O_i from the shadows belonging to a specific subset of storage nodes. Such a subset is described by a qualified subset $\Gamma_i = \{i_1, \cdots, i_t\}$. The application scenario of the proposed technique is depicted in Fig. 2. The dealer encodes m secret images into n shadows by using shadow construction procedure. Then the n shadows are transmitted to n storage nodes. Every user is capable of decoding a secret image by gathering certain shadows from nodes. For example, P_1 can download the 1^{st} and $(n-1)^{th}$ shadows to decode O_1.

All subsets of nodes that can reveal the secret images form the collection of qualified subsets Γ_{Qual}. The GAS $(\Gamma_{Qual}, \Gamma_{Forb})$, described in Section II-B, is used to specify the sharing policy. To guarantee the image data can be well-processed, finite field $GF(2^8)$ is utilized. Shadow construction, matrix-adjusting procedures, and image recovery, which constitute the proposed system, are described in the next three sub-sections. For better comprehension, Table I summarizes the notations used in this paper.

C. Shadow Construction

Based on m secret images $O_1, \cdots, O_m$, the dealer generates n shadows $SH_1, \cdots, SH_n$, which are then distributed to n storage nodes $N_1, \cdots, N_n$. The main steps are given below.

(1) Build a $(n \times n)$ base matrix M_{nn} whose row vectors have Vandermonde coordinates. Let $f^{(n)}(x)$ be a $(1 \times n)$ row vector defined by $f^{(n)}(x) = \sum_{i=1}^{n} x^{i-1} b_i^{(n)}$ where $b_i^{(n)} = [\underbrace{0 \cdots 0}_{i-1} 1 \underbrace{0 \cdots 0}_{n-i}]$ is a $(1 \times n)$ row vector with the i^{th} element being 1 and the remaining $(n-1)$ elements being 0. Given n distinct $x_1, \cdots, x_n \in [1, 255]$, a $(n \times n)$ base matrix M_{nn} is constituted by

$$M_{nn} = \begin{bmatrix} f^{(n)}(x_1) \\ \vdots \\ f^{(n)}(x_n) \end{bmatrix}. \tag{3}$$

Actually, M_{nn} is a matrix whose row vectors have Vandermonde coordinates.

(2) Derive an initial target matrix E based on the collection of qualified subsets Γ_{Qual} and the base matrix M_{nn}. Let $\Gamma_{Qual} = \{\Gamma_1, \cdots, \Gamma_m\}$. For each qualified subset $\Gamma_i = \{i_1, \cdots, i_t\}$, $1 \leq i \leq m$, calculate the corresponding target vector $e_i^{(n)}$ by $e_i^{(n)} = \sum_{j=1}^{n} a_{i,j} f^{(n)}(x_j)$ where the coefficient $a_{i,j}$, $1 \leq j \leq n$, is generated by

$$a_{i,j} = \begin{cases} R(\cdot), \text{ if } j \in \Gamma_i, \\ 0, \text{ if } j \notin \Gamma_i. \end{cases} \tag{4}$$

$R(\cdot)$ outputs a random integer within $[1, 255]$. Essentially, $e_i^{(n)}$ is the linear combination of rows $f^{(n)}(x_1), \cdots, f^{(n)}(x_n)$ of M_{nn}. When having m vectors $e_1^{(n)}, \cdots, e_m^{(n)}$, a $(m \times n)$ initial target matrix E is built by

$$E = \begin{bmatrix} e_1^{(n)} \\ \vdots \\ e_m^{(n)} \end{bmatrix}. \tag{5}$$

(3) Based on the initial target matrix E, construct an invertible target matrix $\tilde{E}$ (i.e., $det(\tilde{E}) \neq 0$ where $det(\tilde{E})$ denotes the determinant of $\tilde{E}$ over $GF(2^8)$). For the following three cases of m and n, three target-matrix-adjusting strategies $\sigma_1(\cdot)$, $\sigma_2(\cdot)$, $\sigma_3(\cdot)$, given in the next sub-section, would be employed accordingly to obtain $\tilde{E}$.

(i) $m = n$. The target matrix $\tilde{E}$ is obtained by

$$\tilde{E} = \begin{cases} E, \text{ if } det(E) \neq 0, \\ \sigma_1(E), \text{ if } det(E) = 0, \end{cases} \tag{6}$$

where $\sigma_1(E)$ turns E into an invertible matrix by randomly adding one row vector and one column vector into E. When $\sigma_1(\cdot)$ is adopted, the size of $\tilde{E}$ is altered to $(n+1) \times (n+1)$, and the base matrix is accordingly modified to

$$M_{(n+1)(n+1)} = \begin{bmatrix} f^{(n+1)}(x_1) \\ \vdots \\ f^{(n+1)}(x_n) \\ f^{(n+1)}(x_{n+1}) \end{bmatrix}. \tag{7}$$

(ii) $m > n$. The target matrix is accomplished by $\tilde{E} = \sigma_2(E)$ where $\sigma_2(E)$ randomly appends $(m-n)$ additional column vectors to E to form an $(m \times m)$ invertible target matrix $\tilde{E}$. Similar to case (i), the base matrix is altered as

$$M_{mm} = \begin{bmatrix} f^{(m)}(x_1) \\ \vdots \\ f^{(m)}(x_m) \end{bmatrix}. \tag{8}$$

(iii) $m < n$. $\tilde{E}$ is derived by $\tilde{E} = \sigma_3(E)$ where $\sigma_3(E)$ randomly adds $(n-m)$ additional row vectors to E to construct a $(n \times n)$ invertible target matrix $\tilde{E}$.

(4) Generate shadows based on the invertible target matrix $\tilde{E}$ and m secret images $O_1, \cdots, O_m$. Suppose each secret image O_i, $1 \leq i \leq m$, consists of $W \times H$ pixels. We denote the j^{th} $(1 \leq j \leq W \times H)$ pixel of the i^{th} secret image as $O_{i,j}$. Let $\tilde{E}^{-1}$ be the inverse of $\tilde{E}$. The following three situations are employed to create shadows.

(i) $m = n$. Two situations are considered for generating shadows: (a) $det(E) \neq 0$, and (b) $det(E) = 0$. For the first situation, since $det(E) \neq 0$, according to Eq. (6), the target matrix $\tilde{E}$ is the same as the initial one. For $1 \leq j \leq W \times H$, a secret matrix $S_j^{(m)}$ having m elements is constructed by

$$S_j^{(m)} = \begin{cases} \begin{bmatrix} O_{1,1} \\ O_{2,1} \\ \vdots \\ O_{m,1} \end{bmatrix}, \text{ if } j = 1, \\ \begin{bmatrix} O_{1,j} + r_1 O_{1,j-1} \\ O_{2,j} + r_2 O_{2,j-1} \\ \vdots \\ O_{m,j} + r_m O_{m,j-1} \end{bmatrix}, \text{ if } j \geq 2, \end{cases} \tag{9}$$

where $r_1, \cdots, r_m$ are m non-zero random integers. Then, a temporary shared matrix is built by $H_j = \tilde{E}^{-1} \times S_j^{(m)}$. The j^{th} pixel of the i^{th} shadow $SH_{i,j}$ is produced as $SH_{i,j} =$

$f^{(n)}(x_i) \times H_j$, $1 \le i \le n$. As all the secret pixels have been processed, n shadows $SH_1, \cdots, SH_n$ are obtained. These n shadows are transmitted to n storage nodes.

For the second case, the size of $\tilde{E}$ is altered as $(n+1)\times(n+1)$ by $\sigma_1(\cdot)$. Accordingly, $(n+1)$ secret images are required to create $(n+1)$ shadows. Note that, a $(W \times H)$ random image $\hat{O}_{n+1}$ is regarded as the $(n+1)^{th}$ secret image. Therefore, we achieve $(n+1)$ secret images $O_1, \cdots, O_n, \hat{O}_{n+1}$. The secret matrix $S_j^{(n+1)}$ with $(n+1)$ elements is obtained by

$$S_j^{(n+1)} = \begin{cases} \begin{bmatrix} O_{1,1} \\ \vdots \\ O_{n,1} \\ \hat{O}_{n+1,1} \end{bmatrix}, \text{ if } j = 1, \\ \begin{bmatrix} O_{1,j} + r_1 O_{1,j-1} \\ \vdots \\ O_{n,j} + r_n O_{n,j-1} \\ \hat{O}_{n+1,j} + r_{n+1}\hat{O}_{n+1,j-1} \end{bmatrix}, \text{ if } j \ge 2, \end{cases} \tag{10}$$

where $r_1, \cdots, r_n, r_{n+1}$ are $(n+1)$ non-zero random numbers. Similarly, we can obtain the shadow pixel $SH_{i,j}$ by $SH_{i,j} = f^{(n+1)}(x_i) \times H_j$, $1 \le i \le n+1$, where $H_j = \tilde{E}^{-1} \times S_j^{(n+1)}$. Finally, we have $(n+1)$ shadows $SH_1, \cdots, SH_n, SH_{n+1}$. The n shadows $SH_1, \cdots, SH_n$ are delivered to n nodes. The $(n+1)^{th}$ shadow SH_{n+1} is referred to as a virtual shadow which is made public. Herein, the concept of virtual shadow is employed. When the adjusting procedures are used, the column size of the target matrix might increase. Additional shadows might be produced as a result. These additional shadows are called virtual shadows. As virtual shadows would be required in image reconstruction, they are made public once these shadows are constructed.

(ii) $m > n$. According to $\sigma_2(\cdot)$, $\tilde{E}$ is a $(m \times m)$ matrix. Since $m > n$, $(m-n)$ random images $\hat{O}_{n+1}, \cdots, \hat{O}_m$ are required to form m secret images $O_1, \cdots, O_n, \hat{O}_{n+1}, \cdots, \hat{O}_m$. At this time, the secret matrix $S_j^{(m)}$ is derived by

$$S_j^{(m)} = \begin{cases} \begin{bmatrix} O_{1,1} \\ \vdots \\ O_{n,1} \\ \hat{O}_{n+1,1} \\ \vdots \\ \hat{O}_{m,1} \end{bmatrix}, \text{ if } j = 1, \\ \begin{bmatrix} O_{1,j} + r_1 O_{1,j-1} \\ \vdots \\ O_{n,j} + r_n O_{n,j-1} \\ \hat{O}_{n+1,j} + r_{n+1}\hat{O}_{n+1,j-1} \\ \vdots \\ \hat{O}_{m,j} + r_m \hat{O}_{m,j-1} \end{bmatrix}, \text{ if } j \ge 2. \end{cases} \tag{11}$$

By using the same approach, we constitute $SH_{i,j}$ by $SH_{i,j} = f^{(m)}(x_i) \times H_j$, $1 \le i \le m$, where $H_j = \tilde{E}^{-1} \times S_j^{(m)}$. Finally, m shadows SH_1, $\cdots$, SH_n, SH_{n+1}, $\cdots$, SH_m are achieved. Among them, the n shadows $SH_1, \cdots, SH_n$ are distributed to n nodes, and the remaining $(m-n)$ shadows $SH_{n+1}, \cdots, SH_m$ are made public.

(iii) $m < n$. When $m < n$, $(n-m)$ rows are randomly added to E to obtain a $(n \times n)$ invertible target matrix $\tilde{E}$ by procedure $\sigma_3(\cdot)$. In this case, we derive $(n-m)$ random images $\hat{O}_{m+1}, \cdots, \hat{O}_n$ to form n secret images $O_1, \cdots, O_m, \hat{O}_{m+1}, \cdots, \hat{O}_n$. The corresponding secret matrix $S_j^{(n)}$ with n elements is given by

$$S_j^{(n)} = \begin{cases} \begin{bmatrix} O_{1,1} \\ \vdots \\ O_{m,1} \\ \hat{O}_{m+1,1} \\ \vdots \\ \hat{O}_{n,1} \end{bmatrix}, \text{ if } j = 1, \\ \begin{bmatrix} O_{1,j} + r_1 O_{1,j-1} \\ \vdots \\ O_{m,j} + r_m O_{m,j-1} \\ \hat{O}_{m+1,j} + r_{m+1}\hat{O}_{m+1,j-1} \\ \vdots \\ \hat{O}_{n,j} + r_n \hat{O}_{n,j-1} \end{bmatrix}, \text{ if } j \ge 2, \end{cases} \tag{12}$$

Similarly, shadow pixel $SH_{i,j}$ is built by $SH_{i,j} = f^{(n)}(x_i) \times H_j$, $1 \le i \le n$, where $H_j = \tilde{E}^{-1} \times S_j^{(n)}$. Finally, we have n shadows $SH_1, \cdots, SH_n$ which are given to n nodes.

In summary, when all shadows are produced, the first n shadows are distributed to the n nodes and the virtual shadows are made public. Meanwhile, the random integers used for building the secret matrix are kept by the users.

D. Strategies for Adjusting Target Matrix

Three adjusting procedures are proposed to derive an invertible target matrix $\tilde{E}$ from the given matrix E. Prior to introducing the algorithms, the analysis of initial target matrix E is provided below.

Given any row vector $e_i^{(n)}$ $(1 \le i \le m)$ of E, $e_i^{(n)}$ is the linear combination of the row vectors $f^{(n)}(x_1), \cdots, f^{(n)}(x_n)$ of the base matrix M_{nn}, as represented by $e_i^{(n)} = a_{i,1} f^{(n)}(x_1) + \cdots + a_{i,n} f^{(n)}(x_n)$. Denote a $(m \times n)$ coefficient matrix for generating E as

$$A_{mn} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{13}$$

Then E is rewritten as $E = A_{mn} \times M_{nn}$. The inverse of E can be accomplished by $E^{-1} = M_{nn}^{-1} \times A_{mn}^{-1}$. M_{nn} is actually a $(n \times n)$ Vandermonte matrix. M_{nn}^{-1} always exists. Hence, if A_{mn} is invertible, we can calculate E^{-1} via the foregoing equation. It also implies if $det(A_{mn}) \neq 0$ then $det(E) \neq 0$. Based on the coefficient matrix A_{mn}, we introduce an indication matrix C_{mn} to record the non-zero items of A_{mn}, as represented by

$$C_{mn} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} \end{bmatrix} \tag{14}$$

where

$$c_{i,j} = \begin{cases} 1, \text{ if } a_{i,j} \neq 0, \\ 0, \text{ otherwise.} \end{cases} \tag{15}$$

According to the analysis, three target-matrix-adjusting procedures $\sigma_1(\cdot)$, $\sigma_2(\cdot)$, and $\sigma_3(\cdot)$ are given as follows.

*1) **Procedure** $\sigma_1(\cdot)$:* For the case of $m = n$ with $det(E) = 0$, $\sigma_1(\cdot)$ is utilized to add one row vector and one column vector to E for obtaining a $(n+1)\times(n+1)$ invertible matrix $\tilde{E}$. The following four steps describes $\sigma_1(\cdot)$.

(1) Let $u_i = [c_{i,1}c_{i,2}\cdots c_{i,n}]$, $1 \le i \le n$, be the i^{th} row of the indication matrix C_{nn}. Then C_{nn} is represented as $C_{nn} = [u_1 u_2 \cdots u_n]^T$. Generate a collection U having (2^n-1) different $(1 \times n)$ row matrices by

$$U = \left\{ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}^T, \cdots, \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}^T \right\}. \tag{16}$$

Note: the all-0 row matrix is not included in U. Construct a row matrix collection $\tilde{U}$ by $\tilde{U} = U \backslash \{u_1, \cdots, u_n\}$.

(2) Randomly select one row matrix $\tilde{u}$ from $\tilde{U}$. Append $\tilde{u}$ to C_{nn} to build a $(n+1)\times n$ matrix $C_{(n+1)n}$, as denoted by

$$C_{(n+1)n} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \\ \tilde{u} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \\ c_{n+1,1} & c_{n+1,2} & \cdots & c_{n+1,n} \end{bmatrix}. \tag{17}$$

(3) Suppose that $v_j = [c_{1,j}c_{2,j}\cdots c_{n+1,j}]^T$ is the j^{th} $(1 \le j \le n)$ column of $C_{(n+1)n}$. Then, $C_{(n+1)n}$ is represented as $C_{(n+1)n} = [v_1 v_2 \cdots v_n]$. Derive a collection V with $(2^n - 1)$ different $(n+1)\times 1$ column matrices by

$$V = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \cdots, \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right\}. \tag{18}$$

For each column vector in V, the $(n+1)^{th}$ element must be 1, and the remaining n elements cannot be all 0s. Then, achieve a collection $\tilde{V}$ by $\tilde{V} = V \backslash \{v_1, \cdots, v_n\}$. Randomly choose one column matrix $\tilde{v}$ from $\tilde{V}$. Build a $(n+1)\times(n+1)$ matrix $C_{(n+1)(n+1)}$ by concatenating $\tilde{v}$ to $C_{(n+1)n}$, as given by

$$\begin{cases} C_{(n+1)(n+1)} = [v_1 v_2 \cdots v_n \tilde{v}] \\ = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} & c_{1,n+1} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} & c_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} & c_{n,n+1} \\ c_{n+1,1} & c_{n+1,2} & \cdots & c_{n+1,n} & c_{n+1,n+1} \end{bmatrix}. \end{cases} \tag{19}$$

(4) According to Steps (2) and (3), $(2n+1)$ elements $c_{n+1,1}$, $\cdots$, $c_{n+1,n+1}$, $c_{1,n+1}$, $\cdots$, $c_{n,n+1}$ are added into C_{nn} to obtain $C_{(n+1)(n+1)}$. Then, we achieve $(2n+1)$ new elements $a_{n+1,1}$, $\cdots$, $a_{n+1,n+1}$, $a_{1,n+1}$, $\cdots$, $a_{n,n+1}$ via

$$a_{i,j} = \begin{cases} R(\cdot), \text{ if } c_{i,j} = 1, \\ 0, \text{ otherwise.} \end{cases} \tag{20}$$

By adding these $(2n+1)$ elements to A_{nn}, a $(n+1)\times(n+1)$ coefficient matrix $A_{(n+1)(n+1)}$ is provided by

$$A_{(n+1)(n+1)} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & a_{1,n+1} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & a_{2,n+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} & a_{n,n+1} \\ a_{n+1,1} & a_{n+1,2} & \cdots & a_{n+1,n} & a_{n+1,n+1} \end{bmatrix}. \tag{21}$$

(5) Calculate $det(A_{(n+1)(n+1)})$. If $det(A_{(n+1)(n+1)}) \neq 0$, accept this matrix. Otherwise, repeat Steps (2)-(4) until the condition $det(A_{(n+1)(n+1)}) \neq 0$ is met. At this time, an invertible coefficient matrix $A_{(n+1)(n+1)}$ is obtained. The invertible target matrix $\tilde{E}$ is achieved via $\tilde{E} = A_{(n+1)(n+1)} \times M_{(n+1)(n+1)}$. Since one additional column vector is appended to the indication matrix, the qualified subsets should be updated accordingly. For the i^{th} $(1 \le i \le m)$ row, add $(n+1)$ to Γ_i if $c_{i,n+1} \neq 0$.

*2) **Procedure** $\sigma_2(\cdot)$:* When $m > n$, $\sigma_2(\cdot)$ is adopted to modify the initial target matrix by adding $(m-n)$ column vectors, as accomplished by the following steps.

(1) Denote the indication matrix as $C_{mn} = [v_1 v_2 \cdots v_n]$ where $v_j = [c_{1,j}c_{2,j}\cdots c_{m,j}]^T$ is the j^{th} $(1 \le j \le n)$ column of matrix C_{mn}. We build a column matrix collection V by

$$V = \left\{ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right\}. \tag{22}$$

Collection V contains $(2^n - 1)$ different column matrices whose sizes are $(m \times 1)$. The all-0 column matrix is not included in V. Produce a collection $\tilde{V} = V \backslash \{v_1, \cdots, v_n\}$.

(2) Randomly select $(m-n)$ different column matrices $\tilde{v}_1, \cdots, \tilde{v}_{m-n}$ from $\tilde{V}$, and concatenate them to C_{mn}. The newly formed matrix C_{mm} is expressed by

$$\begin{cases} C_{mm} = [v_1 \cdots v_n \tilde{v}_1 \cdots \tilde{v}_{m-n}] \\ = \begin{bmatrix} c_{1,1} & \cdots & c_{1,n} & c_{1,n+1} & \cdots & c_{1,m} \\ c_{2,1} & \cdots & c_{2,n} & c_{2,n+1} & \cdots & c_{2,m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & \cdots & c_{m,n} & c_{m,n+1} & \cdots & c_{m,m} \end{bmatrix}. \end{cases} \tag{23}$$

(3) Essentially, $m \times (m-n)$ elements $c_{1,n+1}$, $\cdots$, $c_{m,n+1}$, $\cdots$ $c_{1,m}$, $\cdots$, $c_{m,m}$ are added for achieving C_{mm}. The corresponding $m \times (m-n)$ elements $a_{1,n+1}$, $\cdots$, $a_{m,n+1}$, $\cdots$ $a_{1,m}$, $\cdots$, $a_{m,m}$ are used to constitute A_{mm}, where the $m\times(m-n)$ elements are determined by Eq. (20). The $(m\times m)$ coefficient matrix A_{mm} is represented as

$$A_{mm} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} & a_{1,n+1} & \cdots & a_{1,m} \\ a_{2,1} & \cdots & a_{2,n} & a_{2,n+1} & \cdots & a_{2,m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} & a_{m,n+1} & \cdots & a_{m,m} \end{bmatrix}. \tag{24}$$

(4) Compute $det(A_{mm})$. If $det(A_{mm}) \neq 0$, accept this matrix. Otherwise, repeat Steps (2)-(3) until we have $det(A_{mm}) \neq 0$. Finally, we achieve an invertible coefficient matrix A_{mm}. The target matrix $\tilde{E}$ is then constructed by $\tilde{E} = A_{mm} \times M_{mm}$. Update the qualified subsets based on the new elements of C_{mm}. That is, for the i^{th} $(1 \le i \le m)$ row, put t to Γ_i if $c_{i,t} \neq 0$, where $n+1 \le t \le m$.

*3) **Procedure** $\sigma_3(\cdot)$:* When $m < n$, procedure $\sigma_3(\cdot)$ is employed to alter the initial matrix by appending $(n-m)$ row vectors. The alternation is given as follows.

(1) The row vector $u_i = [c_{i,1}c_{i,2}\cdots c_{i,n}]$, $1 \le i \le m$, is used to denote the indication matrix $C_{mn} = [u_1 u_2 \cdots u_m]^T$. Build a set U having $(2^n - 1)$ different row matrices by

$$U = \left\{ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}^T, \cdots, \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}^T \right\}. \tag{25}$$

The size of each row matrix in U is $(1 \times n)$, and the all-0 matrix is not included. Then, obtain a matrix collection $\tilde{U}$ by $\tilde{U} = U \backslash \{u_1, \cdots, u_m\}$.

(2) Randomly choose $(n-m)$ different row matrices $\tilde{u}_1$, $\cdots$, $\tilde{u}_{n-m}$ from $\tilde{U}$, and append them to C_{mn} for building a $(n \times n)$ new matrix C_{nn}, as represented by

$$C_{nn} = \begin{bmatrix} u_1 \\ \vdots \\ u_m \\ \tilde{u}_1 \\ \vdots \\ \tilde{u}_{n-m} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & \cdots & c_{m,n} \\ c_{m+1,1} & c_{m+1,2} & \cdots & c_{m+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix}. \quad (26)$$

(3) Based on the $n \times (n-m)$ new elements $c_{m+1,1}$, $\cdots$, $c_{m+1,n}$, $\cdots$ $c_{n,1}$, $\cdots$, $c_{n,n}$ in C_{nn}, we calculate the corresponding $n \times (n-m)$ coefficient elements $a_{m+1,1}$, $\cdots$, $a_{m+1,n}$, $\cdots$ $a_{n,1}$, $\cdots$, $a_{n,n}$ by Eq. (20). Thus, the $(n \times n)$ coefficient matrix A_{nn} is determined as

$$A_{nn} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \\ a_{m+1,1} & a_{m+1,2} & \cdots & a_{m+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}. \quad (27)$$

(4) Compute $det(A_{nn})$. If $det(A_{nn}) \neq 0$, accept this matrix. Otherwise, repeat Steps (2)-(3) until $det(A_{nn}) \neq 0$. As a result, an invertible coefficient matrix A_{nn} is obtained. The invertible target matrix $\tilde{E}$ is constructed via $\tilde{E} = A_{nn} \times M_{nn}$.

For the coefficient matrix and indication matrix, they only involve in calculating the target matrix. Once the target matrix is constructed, these two matrices are no longer utilized. The coefficient matrix and indication matrix will not be kept. On the other hand, the base matrix and target matrix are adopted for image recovery. To facilitate the process of image reconstruction, both matrices are made public. The user can access these two public matrices while decoding the secret image. Even though the two matrices are crucial, the security of secret depends on the shadows. According to the proof of Theorem 2, only when all the shadows correlated to the qualified subsets are collected, the secret image is revealed. Publishing both the base matrix and target matrix does not compromise the security.

E. Image Recovery

In image recovery, a secret image is revealed by a user from the corresponding nodes. Note that, one user decodes only one secret image. Any single user cannot recover multiple secret images. Each user P_i, $1 \leq i \leq m$, is correlated with a qualified subset $\Gamma_i = \{i_1, \cdots, i_t, \hat{i}_{t+1}, \cdots, \hat{i}_q\}$, where $SH_{i_1}, \cdots, SH_{i_t}$ are the t shadows from nodes and $SH_{\hat{i}_{t+1}}, \cdots, SH_{\hat{i}_q}$ are the $(q-t)$ virtual shadows that are publicly accessible. Secret image reconstruction for P_i, $1 \leq i \leq m$, is depicted below.

First of all, P_i retrieves t shadows $SH_{i_1}, \cdots, SH_{i_t}$ from t nodes of the network, and obtains $(q-t)$ public shadows $SH_{\hat{i}_{t+1}}, \cdots, SH_{\hat{i}_q}$. Then, achieve the i^{th} target vector $\tilde{e}_i$ from the target matrix $\tilde{E}$. Calculate the revealing vector w_i which satisfies the condition $\tilde{e}_i = w_i \times M_{\Gamma_i}$ where M_{Γ_i} is a matrix constituted by selecting the related rows $i_1, \cdots, i_t, \hat{i}_{t+1}, \cdots, \hat{i}_q$ of base matrix M. A temporary pixel $s_{i,j}$, $1 \leq i \leq m$, $1 \leq j \leq W \times H$, is calculated by $s_{i,j} = w_i \times SH_{\Gamma_i,j}$ where $SH_{\Gamma_i,j} = [SH_{i_1,j}, \cdots, SH_{i_t,j}, SH_{\hat{i}_{t+1},j}, \cdots, SH_{\hat{i}_q,j}]^T$. The secret pixel is achieved by

$$O_{i,j} = \begin{cases} s_{i,j}, \text{ if } j = 1, \\ s_{i,j} - r_i O_{i,j-1}, \text{ otherwise.} \end{cases} \quad (28)$$

When all secret pixels have been decrypted, secret image O_i is recovered.

IV. Theoretical Analysis and Numerical Example

A. Theoretical Analysis

The correctness and security of the proposed scheme are evaluated theoretically. Theorem 1 shows that the shadows correlated to a qualified subset can recover the corresponding secret image. The security of the proposed scheme is analyzed in Theorems 2 and 3.

Theorem 1. *The shadows correlated to a qualified subset $\Gamma_i \in \Gamma_{Qual}$ can reveal the corresponding secret image.*

Proof. Let O_i $(1 \leq i \leq m)$ be the secret image correlated to a qualified subset Γ_i. According to Eq. (28), the j^{th} secret pixel $O_{i,j}$ can be decrypted when the corresponding temporary pixel $s_{i,j}$ is correctly recovered by $s_{i,j} = w_i \times SH_{\Gamma_i,j}$.

Let $SH_{\Omega,j} = [SH_{1,j}\ SH_{2,j}\ \cdots\ SH_{n,j}]^T$ be a $(1 \times n)$ column vector having n shadow pixels, where $\Omega = \{1, 2, \cdots, n\}$. Since $SH_{i,j} = f^{(n)}(x_i) \times H_j$, $SH_{\Omega,j}$ can be derived by

$$\begin{aligned} SH_{\Omega,j} &= M_{nn} \times H_j = M_{nn} \times \tilde{E}^{-1} \times S_j^{(n)} \\ &= M_{nn} \times M_{nn}^{-1} \times A_{nn}^{-1} \times S_j^{(n)} = A_{nn}^{-1} \times S_j^{(n)} \end{aligned} \quad (29)$$

where $S_j^{(n)} = [s_{1,j}\ s_{2,j}\ \cdots\ s_{n,j}]^T$ is a column vector having n temporary pixels. For $SH_{\Omega,j} = A_{nn}^{-1} \times S_j^{(n)}$, by multiplying a matrix A_{nn} on both sides, we obtain $A_{nn} \times SH_{\Omega,j} = A_{nn} \times A_{nn}^{-1} \times S_j^{(n)}$. That is $S_j^{(n)} = A_{nn} \times SH_{\Omega,j}$. As a result, the i^{th} element of $S_j^{(n)}$ is obtained by $s_{i,j} = [a_{i,1} a_{i,2} \cdots a_{i,n}] \times SH_{\Omega,j}$ where $[a_{i,1}\ a_{i,2}\ \cdots\ a_{i,n}]$ is the i^{th} row of A_{nn}. For a qualified subset Γ_i, since A_{nn} is a coefficient matrix, we have $a_{i,j} \neq 0$ if $j \in \Gamma_i$ and $a_{i,j} = 0$ if $j \notin \Gamma_i$. Thus, we achieve $[a_{i,1} a_{i,2} \cdots a_{i,n}] \times SH_{\Omega,j} = w_i \times SH_{\Gamma_i,j}$ where w_i is the revealing vector satisfying $e_i = w_i \times M_{\Gamma_i}$ and $SH_{\Gamma_i,j}$ is the vector containing the j^{th} pixels of the shadows correlated to Γ_i. The above analysis confirms that the temporary pixel $s_{i,j}$ can be correctly revealed by $s_{i,j} = w_i \times SH_{\Gamma_i,j}$. Afterwards, the j^{th} secret pixel $O_{i,j}$ can be successfully decoded. One can decode secret image O_i by the shadows belonging to Γ_i. □

Theorem 2. *The shadows correlated to a subset $\Gamma \notin \Gamma_{Qual}$ cannot give any clue about the secret images.*

Proof. According to the proof of Theorem 1, a secret pixel $O_{i,j}$ is decoded only when the corresponding temporary pixel $s_{i,j}$ is successfully recovered. Given any qualified subset Γ_i, the corresponding temporary pixel is recovered by $s_{i,j} = w_i \times SH_{\Gamma_i,j}$. $SH_{\Gamma_i,j}$ is the vector having t_i pixels of the shadows correlated to Γ_i, where $t_i = |\Gamma_i|$. It implies the temporary pixel

can be reconstructed only when all the t_i shadows belonging to Γ_i are obtained. For each qualified subset Γ_i, it is actually a (t_i, t_i) secret sharing scheme. For any subset $\Gamma \notin \Gamma_{Qual}$ and $\Gamma_{Qual} = \{\Gamma_1, \Gamma_2, \cdots, \Gamma_m\}$, we have $\Gamma \neq \Gamma_1$, $\Gamma \neq \Gamma_2$, $\cdots$, and $\Gamma \neq \Gamma_m$. Consequently, the shadows correlated to Γ cannot give any clue about the secret images.

□

Theorem 3. *The virtual shadows cannot reveal any information about the secret images.*

Proof. According to shadow construction, when $m = n$ with $det(E) = 0$ or $m > n$, there exist virtual shadows. The proof of this theorem is accomplished via the two situations.

For the situation of $m = n$ with $det(E) = 0$, $C_{(n+1)(n+1)}$ is achieved by adding one row matrix and one column matrix to C_{nn}. As a result of this, a virtual shadow SH_{n+1} is created and n qualified subsets might be updated. Actually, for $1 \leq i \leq n$, the i^{th} row of $C_{(n+1)(n+1)}$ represents a qualified subset Γ_i that can reveal the i^{th} secret. More concretely, if $c_{i,n+1} = 1$, $\Gamma_i = \{i_1, \cdots, i_t\}$ is updated as $\Gamma_i = \{i_1, \cdots, i_t, n+1\}$. For the $(n+1)^{th}$ row of $C_{(n+1)(n+1)}$, it also denotes a subset. However, the revealed image by this subset is a random image $\hat{O}_{n+1}$. For each subset correlated to a row of $C_{(n+1)(n+1)}$, we prove that the virtual shadow SH_{n+1} cannot reveal the corresponding secret image. (i) For the first n rows of $C_{(n+1)(n+1)}$, when $c_{i,n+1} = 1$ with $1 \leq i \leq n$, the virtual shadow pixel $SH_{n+1,j}$ involves in recovering the i^{th} secret pixel $O_{i,j}$. According to the proof of Theorem 2, the updated qualified subset $\Gamma_i = \{i_1, \cdots, i_t, n+1\}$ correlates to a $(t+1, t+1)$ secret sharing scheme containing $(t+1)$ shadows pixels $SH_{i_1,j}, \cdots, SH_{i_t,j}, SH_{n+1,j}$. Thus, $SH_{n+1,j}$ cannot disclose any clue about $O_{i,j}$. (ii) When $c_{i,n+1} = 0$ with $1 \leq i \leq n$, the qualified subset remains the same (i.e., $\Gamma_i = \{i_1, \cdots, i_t\}$). The virtual shadow pixel $SH_{n+1,j}$ is not responsible for decrypting the i^{th} secret pixel $O_{i,j}$. Hence, $SH_{n+1,j}$ gives no clue about $O_{i,j}$. (iii) Furthermore, for the $(n+1)^{th}$ row of $C_{(n+1)(n+1)}$, the revealed image is a random image $\hat{O}_{n+1}$. $SH_{n+1,j}$ cannot provide any information about the other secret pixels. According to cases (i)-(iii), the virtual shadow SH_{n+1} discloses no clue about the secrets.

For the situation of $m > n$, $(m-n)$ column matrices are appended to C_{mn} to obtain C_{mm}. Thus, $(m-n)$ virtual shadows SH_{n+1}, $\cdots$, SH_m are produced. Meantime, an updated qualified subset Γ_i corresponding to the i^{th} row of C_{mm} is used to recover the i^{th} secret where $1 \leq i \leq m$. For every qualified subset, we verify that the $(m-n)$ virtual shadows $SH_{n+1}, \cdots, SH_m$ cannot decode the corresponding secret image. Let $\Gamma_i = \{i_1, \cdots, i_t\}$ be the i^{th} initial qualified subset where $1 \leq i \leq m$. (i) When $1 \leq q-t \leq m-n$, $\hat{i}_{t+1} \neq \cdots \neq \hat{i}_q \in \{n+1, \cdots, m\}$, the subset Γ_i corresponding to the i^{th} row of C_{mm} is updated as $\Gamma_i = \{i_1, \cdots, i_t, \hat{i}_{t+1}, \cdots, \hat{i}_q\}$. Essentially, Γ_i correlates to a (q, q) secret sharing approach having q shadows pixels $SH_{i_1,j}, \cdots, SH_{i_t,j}, SH_{\hat{i}_{t+1},j}, \cdots, SH_{\hat{i}_q,j}$. Note that, $SH_{\hat{i}_{t+1},j}, \cdots, SH_{\hat{i}_q,j}$ are the $(q-t)$ virtual shadow pixels involved in decoding secret pixel $O_{i,j}$. Only when all the q shadow pixels are employed, $O_{i,j}$ can be decrypted. These $(q-t)$ virtual shadow pixels give no clue about $O_{i,j}$. Further, the remaining $(m-n-q+t)$ virtual shadow pixels are not responsible for reconstructing the secret pixel. Hence, all these $(m-n)$ virtual shadow pixels cannot reveal the secret pixel. (ii) When $q-t = 0$, Γ_i keeps unchanged (i.e., $\Gamma_i = \{i_1, \cdots, i_t\}$). All these $(m-n)$ virtual shadow pixels do not involve in recovering the secret pixel. They cannot disclose the secret pixel. Based on cases (i) and (ii), we conclude that the $(m-n)$ virtual shadows SH_{n+1}, $\cdots$, SH_m provide no clue about the secrets.

□

B. Numerical Example

Numerical examples are provided to understand the proposed algorithm. Example 1 depicts the shadow construction and image decryption of the proposed method. Examples 2-4 mainly show the three target-matrix-adjusting procedures.

Example 1. *Consider the shadow construction and image recovery of the proposed DM-SIS for* $\Gamma_{Qual} = \{\{1,2\},\{1,3\},\{1,4\},\{2,3,4\}\}$. *Four distinct variables are* $x_1 = 2$, $x_2 = 7$, $x_3 = 9$, *and* $x_4 = 11$. *The base matrix* M_{44} *and coefficient matrix* A_{44} *are given as*

$$M_{44} = \begin{bmatrix} 1 & 2 & 4 & 8 \\ 1 & 7 & 21 & 107 \\ 1 & 9 & 65 & 115 \\ 1 & 11 & 69 & 221 \end{bmatrix}, A_{44} = \begin{bmatrix} 66 & 13 & 0 & 0 \\ 244 & 0 & 87 & 0 \\ 114 & 0 & 0 & 6 \\ 0 & 39 & 24 & 103 \end{bmatrix}. \quad (30)$$

The initial target matrix E *is achieved by*

$$E = \begin{bmatrix} 79 & 167 & 252 & 143 \\ 163 & 32 & 196 & 69 \\ 116 & 222 & 86 & 67 \\ 88 & 155 & 185 & 227 \end{bmatrix}. \quad (31)$$

Since $det(E) = 91$ *over* $GF(2^8)$, $\tilde{E} = E$. *The inverse of* $\tilde{E}$ *is calculated as*

$$\tilde{E}^{-1} = \begin{bmatrix} 207 & 217 & 92 & 76 \\ 124 & 25 & 115 & 199 \\ 139 & 250 & 190 & 40 \\ 207 & 175 & 9 & 230 \end{bmatrix}. \quad (32)$$

Suppose the first and second pixels of four secret images are $O_{1,1} = 211$, $O_{2,1} = 29$, $O_{3,1} = 156$, $O_{4,1} = 8$ *and* $O_{1,2} = 199$, $O_{2,2} = 40$, $O_{3,2} = 177$, $O_{4,2} = 15$, *respectively. Let the four random numbers be* $r_1 = 143$, $r_2 = 5$, $r_3 = 39$, *and* $r_4 = 212$. *Two secret matrices* $S_1^{(4)}$ *and* $S_2^{(4)}$ *are obtained by*

$$S_1^{(4)} = \begin{bmatrix} 211 \\ 29 \\ 156 \\ 8 \end{bmatrix}, S_2^{(4)} = \begin{bmatrix} 199 + 143 \times 211 \\ 40 + 5 \times 29 \\ 177 + 39 \times 156 \\ 15 + 212 \times 8 \end{bmatrix} = \begin{bmatrix} 243 \\ 65 \\ 40 \\ 225 \end{bmatrix} \quad (33)$$

Two temporary shared matrices H_1 *and* H_2 *are generated as*

$$H_1 = \tilde{E}^{-1} \times S_1^{(4)} = \begin{bmatrix} 162 \\ 241 \\ 190 \\ 187 \end{bmatrix}, H_2 = \tilde{E}^{-1} \times S_2^{(4)} = \begin{bmatrix} 0 \\ 126 \\ 148 \\ 118 \end{bmatrix}. \quad (34)$$

As a result, the first shared pixel of the first shadow is calculated by $SH_{1,1} = f^{(4)}(2) \times H_1 = 46$. *By using the same method, the first shared pixels of the remaining three shadows are* $SH_{2,1} = 98$, $SH_{3,1} = 58$, *and* $SH_{4,1} = 42$. *Similarly, the second shared pixels of the four shadows are computed as* $SH_{1,2} = 1$, $SH_{2,2} = 139$, $SH_{3,2} = 208$, *and* $SH_{4,2} = 27$. *The shadow pixels are distributed to* 4 *nodes of the network.*

P_1 *is correlated with* $\Gamma_1 = \{1,2\}$. *By collecting the first and second shadows from the first and second nodes,* P_1 *can reveal the first secret image. Since* P_1 *is the first user, the first*

row of $\tilde{E}$ is selected as target vector, as denoted by $e_1 = [79\ 167\ 252\ 143]$. By choosing the first and second rows of M_{44}, the partial base matrix $M_{\{1,2\}}$ is obtained as

$$M_{\{1,2\}} = \begin{bmatrix} 1 & 2 & 4 & 8 \\ 1 & 7 & 21 & 107 \end{bmatrix}. \tag{35}$$

The revealing vector w_1 satisfying $e_1 = w_1 \times M_{\{1,2\}}$ is computed as $w_1 = [66\ 13]$. When $SH_{1,1} = 46$ and $SH_{2,1} = 98$, a temporary pixel s_1 is calculated as $s_1 = [66\ 13] \times [46\ 98]^T = 211$. Then we have $O_{1,1} = s_1 = 211$. When $SH_{1,2} = 1$ and $SH_{2,2} = 139$, we obtain $s_2 = [66\ 13] \times [1\ 139]^T = 243$. Then $O_{1,2} = s_2 - r_1 \times O_{1,1} = 243 - 143 \times 211 = 199$. The two secret pixels of O_1 are successfully reconstructed. By using the same approach, P_2, P_3, and P_4 are able to recover the secret pixels as well.

Example 2. *Consider the target-matrix-adjusting procedure $\sigma_1(\cdot)$ for the proposed DM-SIS with Γ_{Qual}=$\{\{1,2\}, \{3,4\}, \{1,2,3\}, \{1,2,3,4\}\}$. Four distinct variables are $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, and $x_4 = 4$. The base matrix M_{44} and coefficient matrix A_{44} are obtained as*

$$M_{44} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 5 & 15 \\ 1 & 4 & 16 & 64 \end{bmatrix}, A_{44} = \begin{bmatrix} 3 & 3 & 0 & 0 \\ 0 & 0 & 22 & 1 \\ 6 & 6 & 18 & 0 \\ 12 & 12 & 2 & 2 \end{bmatrix}. \tag{36}$$

As a result, the initial target matrix E is generated by

$$E = \begin{bmatrix} 0 & 5 & 15 & 27 \\ 23 & 62 & 94 & 146 \\ 18 & 60 & 68 & 216 \\ 0 & 26 & 22 & 242 \end{bmatrix}. \tag{37}$$

Since $m = n = 4$ and $det(E) = 0$ over $GF(2^8)$, procedure $\sigma_1(\cdot)$ is adopted to generate $\tilde{E}$. The indication matrix C_{44} is

$$C_{44} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{38}$$

We can derive the row matrix collection $\tilde{U}$. Then, randomly choose a row vector $\tilde{u} = [0\ 1\ 1\ 0]$ from $\tilde{U}$. By appending $\tilde{u}$ to C_{44}, we have C_{54} as below. Moreover, the column matrix collection $\tilde{V}$ is generated. We can randomly select a column vector $\tilde{v} = [1\ 0\ 0\ 0\ 1]^T$ from $\tilde{V}$ and add it to C_{54} to constitute C_{55}, as given below.

$$C_{54} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, C_{55} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \tag{39}$$

Based on the 9 added elements of C_{55} and A_{44}, the (5×5) coefficient matrix A_{55} can be determined as

$$A_{55} = \begin{bmatrix} 3 & 3 & 0 & 0 & 121 \\ 0 & 0 & 22 & 1 & 0 \\ 6 & 6 & 18 & 0 & 0 \\ 12 & 12 & 2 & 2 & 0 \\ 0 & 98 & 210 & 0 & 33 \end{bmatrix}. \tag{40}$$

At this time, $det(A_{55}) = 195 \neq 0$. We accept this coefficient matrix. Meanwhile, based on M_{44} and a random non-zero variable $x_5 = 5$, the base matrix M_{55} is achieved as

$$M_{55} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 5 & 15 & 17 \\ 1 & 4 & 16 & 64 & 29 \\ 1 & 5 & 17 & 85 & 28 \end{bmatrix}. \tag{41}$$

The target matrix $\tilde{E}$ is finally constructed by

$$\tilde{E} = A_{55} \times M_{55} = \begin{bmatrix} 121 & 133 & 181 & 115 & 230 \\ 23 & 62 & 94 & 146 & 118 \\ 18 & 60 & 68 & 216 & 73 \\ 0 & 26 & 22 & 242 & 212 \\ 145 & 10 & 35 & 202 & 166 \end{bmatrix}. \tag{42}$$

Additionally, the inverse of $\tilde{E}$ is

$$\tilde{E}^{-1} = \begin{bmatrix} 76 & 21 & 243 & 217 & 46 \\ 159 & 116 & 116 & 71 & 227 \\ 18 & 52 & 231 & 177 & 150 \\ 233 & 219 & 225 & 131 & 13 \\ 166 & 116 & 70 & 209 & 156 \end{bmatrix}. \tag{43}$$

The collection of the qualified subsets is updated as $\Gamma_{Qual} = \{\{1,2,5\}, \{3,4\}, \{1,2,3\}, \{1,2,3,4\}\}$, where the 5^{th} shadow in $\{1,2,5\}$ is a virtual shadow. Note that, a row $[0\ 1\ 1\ 0\ 1]$ is added to the indication matrix C_{55}, but the corresponding subset $\{2,3,5\}$ is not added to Γ_{Qual} since $\{2,3,5\}$ only reveal a random image.

Let the first and second pixels of five secret images are $(151, 237, 119, 19, 238)$ and $(72, 90, 160, 110, 64)$, respectively, where 238 and 64 are the pixels from random image. Five random numbers used in sharing phase are $(172, 57, 48, 94, 227)$. Two secret matrices $S_1^{(5)}$ and $S_2^{(5)}$ are calculated by

$$S_1^{(5)} = \begin{bmatrix} 151 \\ 237 \\ 119 \\ 19 \\ 238 \end{bmatrix}, S_2^{(5)} = \begin{bmatrix} 72 + 151 \times 172 \\ 90 + 237 \times 57 \\ 160 + 119 \times 48 \\ 110 + 19 \times 94 \\ 64 + 238 \times 227 \end{bmatrix} = \begin{bmatrix} 169 \\ 22 \\ 197 \\ 5 \\ 236 \end{bmatrix} \tag{44}$$

Two temporary shared matrices H_1 and H_2 are generated by

$$H_1 = \tilde{E}^{-1} \times S_1^{(5)} = \begin{bmatrix} 111 \\ 82 \\ 43 \\ 161 \\ 126 \end{bmatrix}, H_2 = \tilde{E}^{-1} \times S_2^{(5)} = \begin{bmatrix} 140 \\ 91 \\ 99 \\ 102 \\ 157 \end{bmatrix}. \tag{45}$$

As a result, the first and second shared pixels of five shadows are calculated as $(201, 181, 242, 96, 186)$ and $(79, 153, 19, 81, 57)$. For P_3 with $\Gamma_3 = \{1,2,3\}$, P_3 can reveal the third secret image by collecting the first, second, and third shadows from the corresponding nodes. The target vector for P_3 is $e_3 = [121\ 133\ 181\ 115\ 230]$. The partial base matrix $M_{\{1,2,3\}}$ is obtained by

$$M_{\{1,2,3\}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 5 & 15 & 17 \end{bmatrix}. \tag{46}$$

Then, the revealing vector w_3 satisfying $e_3 = w_3 \times M_{\{1,2,3\}}$ is computed as $w_3 = [6\ 6\ 18]$. When $SH_{1,1} = 201$, $SH_{2,1} = 181$ and $SH_{3,1} = 242$, a temporary pixel s_1 is calculated as $s_1 = [6\ 6\ 18] \times [201\ 181\ 242]^T = 119$. Then we have $O_{3,1} = s_1 = 119$. When $SH_{1,2} = 79$, $SH_{2,2} = 153$ and $SH_{3,2} = 19$, we obtain $s_2 = [6\ 6\ 18] \times [79\ 153\ 19]^T = 197$. Then $O_{3,2} = s_2 - r_3 \times O_{3,1} = 197 - 48 \times 119 = 160$. The two secret pixels of O_3 are successfully reconstructed. Similarly, other secret pixels can be revealed by the correlated users.

Example 3. *Consider the target-matrix-adjusting procedure $\sigma_2(\cdot)$ for the proposed DM-SIS with Γ_{Qual}=$\{\{1,2\}, \{1,3\}, \{2,3\}, \{3,4\}, \{1,2,3\}, \{2,3,4\}\}$. Herein, $m = 6$ and $n = 4$. Four distinct variables are $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, and $x_4 = 4$. The base matrix M_{44} over $GF(2^8)$ is obtained as*

$$M_{44} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 5 & 15 \\ 1 & 4 & 16 & 64 \end{bmatrix}. \tag{47}$$

Coefficient matrix A_{64} and indication matrix C_{64} are given by

$$A_{64}=\begin{bmatrix}20&148&0&0\\56&0&33&0\\0&59&144&0\\0&0&4&92\\65&172&8&0\\0&13&119&60\end{bmatrix},C_{64}=\begin{bmatrix}1&1&0&0\\1&0&1&0\\0&1&1&0\\0&0&1&1\\1&1&1&0\\0&1&1&1\end{bmatrix}. \quad (48)$$

By adding two randomly selected column vectors $\tilde{v_1}=[1\ 0\ 0\ 1\ 0\ 0]^T$ and $\tilde{v_2}=[0\ 1\ 0\ 0\ 0\ 1]^T$ to C_{64}, we have C_{66}, as given below.

$$C_{66}=\begin{bmatrix}1&1&0&0&1&0\\1&0&1&0&0&1\\0&1&1&0&0&0\\0&0&1&1&1&0\\1&1&1&0&0&0\\0&1&1&1&0&1\end{bmatrix}. \quad (49)$$

Based on C_{66} and A_{64}, we generate A_{66} as

$$A_{66}=\begin{bmatrix}20&148&0&0&86&0\\56&0&33&0&0&180\\0&59&144&0&0&0\\0&0&4&92&155&0\\65&172&8&0&0&0\\0&13&119&60&0&251\end{bmatrix}. \quad (50)$$

Since $det(A_{66})=145\neq 0$, we accept A_{66}. With $x_5=5$ and $x_6=6$, the base matrix M_{66} is generated as

$$M_{66}=\begin{bmatrix}1&1&1&1&1&1\\1&2&4&8&16&32\\1&3&5&15&17&51\\1&4&16&64&29&116\\1&5&17&85&28&108\\1&6&20&120&13&46\end{bmatrix}. \quad (51)$$

Finally, the target matrix $\tilde{E}$ is constituted by

$$\tilde{E}=A_{66}\times M_{66}=\begin{bmatrix}214&50&33&254&103&154\\173&196&248&137&164&112\\171&219&6&230&242&156\\195&172&99&35&185&160\\229&28&227&48&219&224\\189&83&165&174&233&131\end{bmatrix}. \quad (52)$$

Meanwhile, the inverse of $\tilde{E}$ is calculated by

$$\tilde{E}^{-1}=\begin{bmatrix}113&28&4&0&41&32\\164&46&86&2&18&217\\150&118&36&163&254&190\\123&189&1&131&14&70\\198&197&229&72&65&199\\215&131&126&206&179&38\end{bmatrix}. \quad (53)$$

The collection of qualified subsets is updated as $\Gamma_{Qual}=\{\{1,2,5\},\ \{1,3,6\},\ \{2,3\},\ \{3,4,5\},\ \{1,2,3\},\ \{2,3,4,6\}\}$. The 5^{th} and 6^{th} shadows are virtual shadows.

Let $(151, 237, 119, 19, 147, 2)$ and $(72, 90, 160, 110, 8, 38)$ be the first and second pixels of six secret images. $(164, 238, 181, 233, 100, 180)$ are the six random numbers. Two secret matrices $S_1^{(6)}$ and $S_2^{(6)}$ are obtained by

$$S_1^{(6)}=\begin{bmatrix}151\\237\\119\\19\\147\\2\end{bmatrix},S_2^{(6)}=\begin{bmatrix}72+151\times 164\\90+237\times 238\\160+119\times 181\\110+19\times 233\\8+147\times 100\\38+2\times 180\end{bmatrix}=\begin{bmatrix}101\\139\\118\\126\\218\\83\end{bmatrix} \quad (54)$$

Then, two temporary shared matrices are achieved as

$$H_1=\tilde{E}^{-1}\times S_1^{(6)}=\begin{bmatrix}156\\89\\221\\134\\109\\50\end{bmatrix},H_2=\tilde{E}^{-1}\times S_2^{(6)}=\begin{bmatrix}44\\69\\72\\170\\29\\100\end{bmatrix}. \quad (55)$$

The first and second shared pixels of six shadows are derived as $(193, 169, 164, 102, 41, 42)$ and $(242, 115, 176, 239, 188, 195)$, where the 5^{th} and 6^{th} shadows are virtual. The first four shadow pixels are distributed to four nodes. P_2 with $\Gamma_2=\{1,3,6\}$ is supposed to decrypt the second secret image by accessing the first, third and sixth shadows. As the target vector for P_2 is obtained by $e_2=[173\ 196\ 248\ 137\ 164\ 112]$, the partial base matrix $M_{\{1,3,6\}}$ is achieved as

$$M_{\{1,3,6\}}=\begin{bmatrix}1&1&1&1&1&1\\1&3&5&15&17&51\\1&6&20&120&13&46\end{bmatrix}. \quad (56)$$

The revealing vector is calculated as $w_2=[56\ 33\ 180]$. With $SH_{1,1}=193$, $SH_{3,1}=164$, and $SH_{6,1}=42$, a temporary pixel s_1 is calculated as $s_1=[56\ 33\ 180]\times[193\ 164\ 42]^T=237$. Then we have $O_{2,1}=s_1=237$. When $SH_{1,2}=242$, $SH_{3,2}=176$ and $SH_{6,2}=195$, $s_2=[56\ 33\ 180]\times[242\ 176\ 195]^T=139$. Then $O_{2,2}=s_2-r_2\times O_{2,1}=139-238\times 237=90$. The two secret pixels of O_2 are successfully reconstructed. The remaining users are capable of decoding the secret pixels based on the same technique.

Example 4. *Consider the target-matrix-adjusting procedure $\sigma_3(\cdot)$ for the proposed DM-SIS with $\Gamma_{Qual}=\{\{1,2\},\{3,4\},\{1,5,6\}\}$. Herein, $m=3$ and $n=6$. Six distinct variables are $x_1=1$, $x_2=2$, $x_3=3$, $x_4=4$, $x_5=5$, and $x_6=6$. The base matrix M_{66}, coefficient matrix A_{36}, and indication matrix C_{36} are given by*

$$\begin{cases}M_{66}=\begin{bmatrix}1&1&1&1&1&1\\1&2&4&8&16&32\\1&3&5&15&17&51\\1&4&16&64&29&116\\1&5&17&85&28&108\\1&6&20&120&13&46\end{bmatrix},\\ A_{36}=\begin{bmatrix}64&165&0&0&0&0\\0&0&83&218&0&0\\92&0&0&0&38&146\end{bmatrix},\\ C_{36}=\begin{bmatrix}1&1&0&0&0&0\\0&0&1&1&0&0\\1&0&0&0&1&1\end{bmatrix}.\end{cases} \quad (57)$$

We randomly choose 3 row vectors $\tilde{u_1}=[1\ 0\ 1\ 0\ 0\ 0]$, $\tilde{u_2}=[0\ 1\ 1\ 0\ 0\ 1]$ and $\tilde{u_3}=[0\ 1\ 1\ 1\ 0\ 1]$, and append them to C_{36} to obtain C_{66}, as represented by

$$C_{66}=\begin{bmatrix}1&1&0&0&0&0\\0&0&1&1&0&0\\1&0&0&0&1&1\\1&0&1&0&0&0\\0&1&1&0&0&1\\0&1&1&1&0&1\end{bmatrix}. \quad (58)$$

Based on C_{66} and A_{36}, A_{66} is constructed as

$$A_{66}=\begin{bmatrix}64&165&0&0&0&0\\0&0&83&218&0&0\\92&0&0&0&38&146\\31&0&9&0&0&0\\0&123&222&0&0&26\\0&3&88&191&0&77\end{bmatrix}. \quad (59)$$

Since $det(A_{66})=22\neq 0$, we accept A_{66} for building the target matrix $\tilde{E}$, which is calculated by

$$\tilde{E}=A_{66}\times M_{66}=\begin{bmatrix}229&23&238&1&194&89\\137&186&35&130&32&182\\232&169&135&56&183&48\\22&4&50&104&134&169\\191&213&165&165&254&236\\169&155&155&130&96&97\end{bmatrix}. \quad (60)$$

Finally, the inverse of $\tilde{E}$ is computed by

$$\tilde{E}^{-1} = \begin{bmatrix} 144 & 189 & 119 & 112 & 156 & 24 \\ 136 & 196 & 163 & 147 & 189 & 190 \\ 205 & 197 & 131 & 156 & 113 & 176 \\ 250 & 59 & 68 & 12 & 179 & 179 \\ 253 & 95 & 233 & 9 & 188 & 102 \\ 171 & 214 & 250 & 74 & 237 & 112 \end{bmatrix}. \quad (61)$$

The collection of qualified subsets are updated as $\Gamma_{Qual} = \{\{1,2\},\{3,4\},\{1,5,6\},\{1,3\},\{2,3,6\},\{2,3,4,6\}\}$, where the 4^{th}, 5^{th} and 6^{th} shadows are virtual ones.

Suppose the first and second pixels of five secret images are $(151, 237, 119, 154, 152, 132)$ and $(72, 90, 160, 171, 179, 166)$. With six random numbers $(175, 163, 14, 231, 29, 100)$, two secret matrices $S_1^{(6)}$ and $S_2^{(6)}$ are generated by

$$S_1^{(6)} = \begin{bmatrix} 151 \\ 237 \\ 119 \\ 154 \\ 152 \\ 132 \end{bmatrix}, S_2^{(6)} = \begin{bmatrix} 72 + 151 \times 175 \\ 90 + 237 \times 163 \\ 160 + 119 \times 14 \\ 171 + 154 \times 231 \\ 179 + 152 \times 29 \\ 166 + 132 \times 100 \end{bmatrix} = \begin{bmatrix} 13 \\ 11 \\ 16 \\ 59 \\ 176 \\ 91 \end{bmatrix} \quad (62)$$

Two temporary shared matrices are achieved as

$$H_1 = \tilde{E}^{-1} \times S_1^{(6)} = \begin{bmatrix} 126 \\ 62 \\ 204 \\ 215 \\ 227 \\ 175 \end{bmatrix}, H_2 = \tilde{E}^{-1} \times S_2^{(6)} = \begin{bmatrix} 117 \\ 97 \\ 113 \\ 137 \\ 60 \\ 99 \end{bmatrix}. \quad (63)$$

The first and second shared pixels of six shadows are calculated by $(23, 49, 241, 196, 187, 57)$ and $(179, 73, 38, 114, 240, 88)$. Since $\Gamma_1 = \{1,2\}$, the first secret image correlated with P_1 is decoded by collecting the first and second shadows. With the target vector $e_1 = [229\ 23\ 238\ 1\ 194\ 89]$, the partial base matrix $M_{\{1,2\}}$ is obtained by

$$M_{\{1,2\}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 & 32 \end{bmatrix}. \quad (64)$$

The revealing vector is calculated as $w_1 = [64\ 165]$. When $SH_{1,1} = 23$ and $SH_{2,1} = 49$, a temporary pixel s_1 is calculated as $s_1 = [64\ 165] \times [23\ 49]^T = 151$. Then we have $O_{1,1} = s_1 = 151$. When $SH_{1,2} = 179$ and $SH_{2,2} = 73$, we obtain $s_2 = [64\ 165] \times [179\ 73]^T = 13$. Then $O_{1,2} = s_2 - r_1 \times O_{1,1} = 13 - 175 \times 151 = 72$. The two secret pixels of O_1 are decoded. Other secret pixels can be disclosed in a similar manner.

V. Experimental Result and Discussion

A. Demonstration Example

To illustrate the effectiveness of the proposed technique, 49 test images with 512×512 pixels from USC-SIPI and CVG-UGR image databases are regarded as secret images for experiments, as illustrated in Fig. 3. The first experiment of the proposed DM-SIS with $\Gamma_{Qual} = \{\{1,2\},\{1,3\},\{1,4\},\{2,3,4\}\}$ is shown in Fig. 4, where $m = 4$ and $n = 4$. All parameters for this experiment are given in Example 1. The 4 test images with IDs $1-4$ are used as secret images. Based on the proposed method, 4 shadows are constructed and delivered to 4 storage nodes, as given in Figs. 4 (a)-(d). P_1 recovers the first secret image from nodes N_1 and N_2, as shown in Fig. 4 (e). Based on the qualified subsets $\{1,3\},\{1,4\},\{2,3,4\}$, P_2, P_3 and P_4 reconstruct their corresponding secret images as well, as demonstrated in Figs. 4 (f)-(h). The second experiment of the proposed DM-SIS with $\Gamma_{Qual} = \{\{1,2\},\{1,3\},\{2,3\},\{3,4\},\{1,2,3\},\{2,3,4\}\}$ is demonstrated in Fig. 5, where $m = 6$ and $n = 4$. Parameters for this experiment can refer to Example 3. The 6 secret images are chosen from the test images with IDs $5-10$. 6 shadows generated by the proposed scheme are illustrated in Figs. 5 (a)-(f), where the first 4 shadows are distributed to 4 nodes and the last 2 shadows are virtual shadows that are made public. The corresponding shadows are capable of decoding the secret images, as depicted in Figs. 5 (g)-(l).

Fig. 3. 49 test images (IDs from 1 to 49) from USC-SIPI and CVG-UGR image databases.

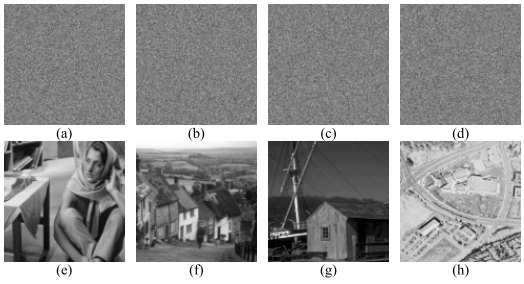

Fig. 4. Experiment of the proposed DM-SIS with Γ_{Qual} = $\{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3,4\}\}$. (a)-(d) 4 shadows for 4 nodes, (e)-(h) recovered secret images by the corresponding 4 qualified subsets.

More experiments by using the 49 test images are given in Table II. Various access structures are used, as described in Table III. Correlation coefficient (CC) and structural similarity (SSIM) are adopted to evaluate the security of the shadows, while the visual quality of recovered secret images is measured by PSNR. According to the experiments, the values of CC and SSIM are sufficiently small to guarantee that the shadows cannot disclose the secrets. Meanwhile, the PSNRs are ∞ which means all the secret images are losslessly reconstructed. Furthermore, an experiment using color secret images is shown in Fig. 6. The effectiveness of the proposed method for color images is verified as well.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3603223

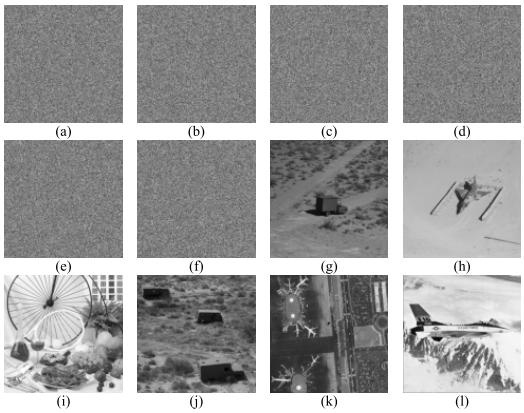

Fig. 5. Experiment of the proposed DM-SIS with $\Gamma_{Qual} = \{\{1,2\}, \{1,3\}, \{2,3\}, \{3,4\}, \{1,2,3\}, \{2,3,4\}\}$. (a)-(d) 4 shadows for 4 nodes, (e)-(f) 2 public shadows, (g)-(l) recovered secret images by the corresponding 6 qualified subsets.

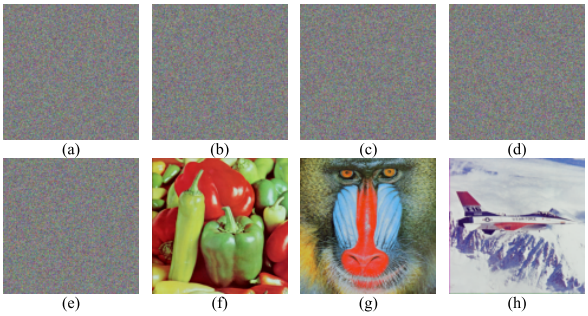

Fig. 6. Experiment of the proposed DM-SIS with $\Gamma_{Qual} = \{\{1,3\}, \{2,4\}, \{1,4,5\}\}$ for color images. (a)-(e) 5 shadows for 5 nodes, (f)-(h) recovered secret images by the corresponding 3 qualified subsets.

B. Applicability for Various Access Structures

Table III shows various access structures used in this paper. Some commonly used thresholds (IDs $1-6$) and GASs (IDs $7-20$) are adopted. For IDs $1-6$, the collection of qualified subsets is represented by (k,n). The quantity of qualified subsets of a (k,n) threshold is the number of combinations of any k nodes, as calculated by $\binom{n}{k}$. Therefore, the number of users m is $\binom{n}{k}$ as well. Take the $(2,4)$ threshold of ID 2 for example. The qualified subsets of $(2,4)$ threshold are $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, $\{2,3\}$, $\{2,4\}$, and $\{3,4\}$. The number of users m is computed as $\binom{4}{2} = 6$. We can pre-determine the number of qualified subsets of a GAS. Then, several storage nodes are selected, at random or according to the requirement, to create a qualified subset. Each qualified subset should stick to the GAS definition outlined in Section II-B. Apart from the fundamental definition, there is no additional restriction on GAS. The comparison of applicability among related schemes [26] [27] and our method is provided in Table IV. Existing approaches are confined to (k,n) threshold, while our method is applicable for GAS, where threshold schemes are the special cases of GAS. Complicated sharing policy can be implemented by our DM-SIS. The application scenario is further enriched.

TABLE II
EXPERIMENTS USING 49 TEST IMAGES FROM USC-SIPI AND CVG-UGR IMAGE DATABASES.

Access Structure ID	Secret ID	PSNR	Shadow No.	CC	SSIM
1	1	∞	1	-2.27×10^{-3}	8.66×10^{-3}
	2	∞	2	-6.42×10^{-4}	8.83×10^{-3}
	3	∞	3	-1.50×10^{-3}	7.88×10^{-3}
2	4	∞	1	8.67×10^{-4}	8.77×10^{-3}
	5	∞	2	8.46×10^{-4}	1.04×10^{-2}
	6	∞	3	-6.19×10^{-3}	9.44×10^{-3}
	7	∞	4	9.34×10^{-4}	8.81×10^{-3}
	8	∞	5	2.48×10^{-3}	9.73×10^{-3}
	9	∞	6	-2.14×10^{-5}	8.83×10^{-3}
8	10	∞	1	-5.94×10^{-4}	8.93×10^{-3}
	11	∞	2	-1.20×10^{-3}	6.52×10^{-3}
	12	∞	3	3.23×10^{-4}	6.99×10^{-3}
9	13	∞	1	-1.72×10^{-3}	9.04×10^{-3}
	14	∞	2	8.37×10^{-4}	8.79×10^{-3}
	15	∞	3	-1.72×10^{-3}	5.83×10^{-3}
11	16	∞	1	-6.75×10^{-4}	9.21×10^{-3}
	17	∞	2	-2.66×10^{-3}	7.82×10^{-3}
	18	∞	3	-1.98×10^{-3}	6.98×10^{-3}
	19	∞	4	-1.88×10^{-3}	6.98×10^{-3}
	20	∞	5	1.44×10^{-3}	9.71×10^{-3}
	21	∞	6	-1.57×10^{-3}	7.84×10^{-3}
	22	∞	7	-1.94×10^{-3}	8.61×10^{-3}
12	23	∞	1	5.66×10^{-4}	8.32×10^{-3}
	24	∞	2	-8.82×10^{-4}	8.78×10^{-3}
	25	∞	3	-5.12×10^{-4}	9.45×10^{-3}
13	26	∞	1	1.06×10^{-3}	9.30×10^{-3}
	27	∞	2	4.17×10^{-4}	7.43×10^{-3}
	28	∞	3	3.78×10^{-4}	9.97×10^{-3}
	29	∞	4	-4.41×10^{-3}	6.47×10^{-3}
	30	∞	5	-1.56×10^{-3}	7.35×10^{-3}
14	31	∞	1	-1.75×10^{-4}	8.24×10^{-3}
	32	∞	2	-2.79×10^{-3}	6.68×10^{-3}
	33	∞	3	-3.15×10^{-3}	7.49×10^{-3}
	34	∞	4	-5.11×10^{-4}	8.01×10^{-3}
	35	∞	5	2.29×10^{-3}	1.02×10^{-2}
	36	∞	6	1.27×10^{-3}	7.38×10^{-3}
	37	∞	7	1.71×10^{-3}	8.43×10^{-3}
	38	∞	8	3.46×10^{-3}	8.81×10^{-3}
16	39	∞	1	7.13×10^{-4}	8.92×10^{-3}
	40	∞	2	4.27×10^{-3}	1.09×10^{-2}
	41	∞	3	-7.69×10^{-4}	9.50×10^{-3}
	42	∞	4	1.28×10^{-3}	9.16×10^{-3}
	43	∞	5	-2.50×10^{-3}	7.10×10^{-3}
	44	∞	6	2.04×10^{-3}	9.94×10^{-3}
18	45	∞	1	-1.44×10^{-3}	9.03×10^{-3}
	46	∞	2	1.23×10^{-3}	8.07×10^{-3}
	47	∞	3	1.25×10^{-3}	9.80×10^{-3}
	48	∞	4	1.66×10^{-3}	9.51×10^{-3}
	49	∞	5	7.06×10^{-4}	9.62×10^{-3}
Average	-	∞	-	-2.70×10^{-4}	8.54×10^{-3}

C. Storage Overhead

Storage overhead of a node, indicating the amount of shadows assigned to a node, is expected to be as low as feasible. The size of a shadow is the same as the secret for the related methods [26] [27] and our scheme. For a fair comparison, we adopt the number of shadows delivered to a node to evaluate the storage overhead.

The comparison of storage overhead for each node is depicted in Table V. Three methods in [26] and the improved approach in [27] are included. For each item in Table V, there are five values. From left to right, we have the results produced by the first, second, and third methods in [26], the improved

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3603223

TABLE V
COMPARISON OF STORAGE OVERHEAD (FOR EACH ITEM, LEFT THREE: [26], MIDDLE TWO: [27] AND RIGHT: OUR).

Access structure	No. of shadows for each node							
	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8
(2,3)	1,1,1,1,1	2,1,1,1,1	3,1,1,1,1	-	-	-	-	-
(2,4)	1,1,2,1,1	2,1,2,1,1	3,2,1,2,1	4,2,1,3,1	-	-	-	-
(2,5)	1,1,2,1,1	2,1,2,1,1	3,2,2,1,1	4,3,2,3,1	5,3,2,4,1	-	-	-
(2,6)	1,1,3,1,1	2,1,3,1,1	3,2,3,1,1	4,3,2,3,1	5,4,2,4,1	6,4,2,5,1	-	-
(2,7)	1,1,3,1,1	2,1,3,1,1	3,2,3,1,1	4,3,3,3,1	5,4,3,4,1	6,5,3,5,1	7,5,3,6,1	-
(2,8)	1,1,4,1,1	2,1,4,1,1	3,2,4,1,1	4,3,4,3,1	5,4,3,4,1	6,5,3,5,1	7,6,3,6,1	8,6,3,7,1
(3,4)	1,1,1,1,1	1,1,1,1,1	2,1,1,1,1	4,1,1,1,1	-	-	-	-
(3,5)	1,1,2,1,1	1,1,2,1,1	2,1,2,1,1	4,3,2,1,1	7,4,2,6,1	-	-	-
(3,6)	1,1,4,1,1	1,1,4,1,1	2,1,3,1,1	4,3,3,1,1	7,6,3,6,1	11,8,3,10,1	-	-
(3,7)	1,1,5,1,1	1,1,5,1,1	2,1,5,1,1	4,3,5,1,1	7,6,5,6,1	11,10,5,10,1	16,13,5,15,1	-
(3,8)	1,1,7,1,1	1,1,7,1,1	2,1,7,1,1	4,3,7,1,1	7,6,7,6,1	11,10,7,10,1	16,15,7,15,1	22,19,7,21,1
(4,5)	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	2,1,1,1,1	5,1,1,1,1	-	-	-
(4,6)	1,1,3,1,1	1,1,3,1,1	1,1,3,1,1	2,1,2,1,1	5,4,2,1,1	11,7,2,10,1	-	-
(4,7)	1,1,5,1,1	1,1,5,1,1	1,1,5,1,1	2,1,5,1,1	5,4,5,1,1	11,10,5,10,1	21,17,5,20,1	-
(4,8)	1,1,9,1,1	1,1,9,1,1	1,1,9,1,1	2,1,9,1,1	5,4,9,1,1	11,10,9,10,1	21,20,8,20,1	36,32,8,35,1
(5,6)	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	2,1,1,1,1	6,1,1,1,1	-	-
(5,7)	1,1,3,1,1	1,1,3,1,1	1,1,3,1,1	1,1,3,1,1	2,1,3,1,1	6,5,3,1,1	16,11,3,15,1	-
(5,8)	1,1,7,1,1	1,1,7,1,1	1,1,7,1,1	1,1,7,1,1	2,1,7,1,1	6,5,7,1,1	16,15,7,15,1	36,31,7,35,1
(6,7)	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	2,1,1,1,1	7,1,1,1,1	-
(6,8)	1,1,4,1,1	1,1,4,1,1	1,1,4,1,1	1,1,4,1,1	1,1,3,1,1	2,1,3,1,1	7,6,3,1,1	22,16,3,21,1
(7,8)	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	1,1,1,1,1	2,1,1,1,1	8,1,1,1,1

TABLE III
THE ACCESS STRUCTURES AND THE PARAMETERS USED IN THIS PAPER.

ID	m	n	Collection of qualified subsets
1	3	3	$(2,3)$[1]
2	6	4	$(2,4)$[1]
3	10	5	$(3,5)$[1]
4	15	6	$(4,6)$[1]
5	35	7	$(4,7)$[1]
6	21	7	$(5,7)$[1]
7	2	3	$\{1,2\}, \{2,3\}$
8	3	3	$\{1,2\}, \{1,3\}, \{1,2,3\}$
9	3	4	$\{1,2\}, \{3,4\}, \{1,2,4\}$
10	4	4	$\{1,3\}, \{1,4\}, \{1,3,4\}, \{2,3,4\}$
11	7	4	$\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{1,2,3\}, \{1,2,4\}, \{1,2,3,4\}$
12	3	5	$\{1,2\}, \{1,3\}, \{2,4,5\}$
13	5	5	$\{1,2\}, \{2,3\}, \{2,4\}, \{1,2,3\}, \{3,4,5\}$
14	8	5	$\{1,3\}, \{1,4\}, \{2,3\}, \{1,3,4\}, \{2,3,5\}, \{1,2,3,5\}, \{1,3,4,5\}, \{2,3,4,5\}$
15	4	6	$\{1,5\}, \{1,3,4\}, \{2,4,5\}, \{3,5,6\}$
16	6	6	$\{1,2,3\}, \{2,4,5\}, \{3,5,6\}, \{1,2,3,4\}, \{1,3,4,6\}, \{2,3,5,6\}$
17	9	6	$\{1,3\}, \{1,6\}, \{1,3,5\}, \{2,4,6\}, \{4,5,6\}, \{2,3,4,5\}, \{3,4,5,6\}, \{1,2,4,5,6\}, \{2,3,4,5,6\}$
18	5	7	$\{1,3\}, \{2,4,5\}, \{3,4,7\}, \{1,2,5,7\}, \{2,3,5,6,7\}$
19	7	7	$\{3,6\}, \{1,2,3\}, \{2,4,7\}, \{1,5,6,7\}, \{2,3,4,5\}, \{1,2,4,5,7\}, \{2,3,4,5,6,7\}$
20	10	7	$\{1,3,4\}, \{1,4,5\}, \{2,3,6\}, \{3,4,7\}, \{1,2,3,4\}, \{3,4,6,7\}, \{1,3,5,6,7\}, \{2,3,4,5,6\}, \{1,2,3,4,5,6\}, \{1,2,4,5,6,7\}$

1: Use (k,n) threshold to represent the collection of qualified subsets. The number of users m is therefore $\binom{n}{k}$.

TABLE IV
COMPARISON OF APPLICABILITY FOR VARIOUS ACCESS STRUCTURES.

Access Structure ID	Scheme		
	Ref. [26]	Ref. [27]	Our
1	✓	✓	✓
2	✓	✓	✓
3	✓	✓	✓
4	✓	✓	✓
5	✓	✓	✓
6	✓	✓	✓
7	✗	✗	✓
8	✗	✗	✓
9	✗	✗	✓
10	✗	✗	✓
11	✗	✗	✓
12	✗	✗	✓
13	✗	✗	✓
14	✗	✗	✓
15	✗	✗	✓
16	✗	✗	✓
17	✗	✗	✓
18	✗	✗	✓
19	✗	✗	✓
20	✗	✗	✓

DSSP in [27], and our DM-SIS. Consider the item of node N_8 for the $(3,8)$ case. The three schemes in [26] would store 22, 19, and 7 shadows in this node, and the improved method in [27] has 21 shadows. Nevertheless, our method delivers 1 shadow to node N_8. According to Table V, our scheme obtains improved storage overhead for each node, where the number of shadows for each node is always 1. Whereas, multiple shadows might be transmitted to a node by the techniques in [26] and [27]. For some special cases, such as $(4,8)$ and $(5,8)$, the number of received shadows for a node (e.g., N_8) would dramatically increase.

D. Computation Complexity

For the first approach with nearly optimal storage overhead in [26], the sharing phase consists of preprocessing and encoding, where the preprocessing includes the generations of D^{-1} and a^t. Since D is $(k-1)\times(k-1)$, the computation complexity of D^{-1} using Gaussian elimination method is $O((k-1)^3)$. With $a^t = [\gamma_k\ \gamma_k^2\ \cdots\ \gamma_k^{k-1}]D^{-1}$, the complexity of calculating a^t is $O((k-1)^2)$. Therefore, the complexity of preprocessing is $O((k-1)^3 + (k-1)^2) = O(k^3)$. The encoding complexity for generating $m = \binom{n}{k}$ shadows is $O(mk)$. In recovery phase, every user recovers the secret using

Lagrange interpolation with complexity of $O(k\log^2 k)$. In this case, the complexity of reconstructing m secrets is therefore $O(mk\log^2 k)$. The authors also introduce another method for recovering the secrets. In this situation, the recovery phase comprises preprocessing and decoding. In preprocessing, D^{-1} and a^t are required. So that the complexity is $O(k^3)$. Then, each secret is decoded with complexity of $O((k-1))$. Since there are m secrets, the total complexity is $O(mk)$ if D^{-1} and a^t are calculated in a prior. For the second approach with optimal storage overhead in [26], the preprocessing of sharing phase consists of the calculations of A^{-1}, E, D^{-1} and a^t. The corresponding computation complexities are $O(k^3n^3)$, $O(kn^3)$, $O((k-1)^3)$, and $O((k-1)^2)$. Thus, the total complexity of preprocessing is $O(k^3n^3)$. When encoding the first n secrets, the complexity is $O(n^2)$. For the remaining $(m-n)$ secrets, the same encoding procedure used in the first approach is adopted. Hence, the complexity is $O((m-n)k)$. The recovery phase of this approach is the same as the first method. For the third technique in [26], both the sharing and recovery are identical to those of the second method.

For the scheme in [27], the preprocessing of sharing phase is to calculate M^{-1}, where M is a circulant matrix. M^{-1} can be obtained via the form given in [27] with complexity of $O(1)$. When encoding the $m=\binom{n}{k}$ secrets, the first $(k+1)$ secrets are encrypted with complexity of $O(k+1)$, and then the remaining $(m-k-1)$ secrets are obtained via the iterative shadow construction. For each iteration, a (k,n') scheme is converted to $(k,n'+1)$ by creating $\binom{n'}{k-1}$ new shadows. The complexity of each iteration is $O(\binom{n'}{k-1})$. As there are $(n-k-1)$ iterations, the complexity of encrypting the remaining $(m-k-1)$ secrets is $O(\sum_{i=k+1}^{n}\binom{i}{k-1})$. The total complexity of encoding is $O(k+\sum_{i=k+1}^{n}\binom{i}{k-1})$ as a result. When decoding the m secrets, the corresponding complexity is $O(m)$.

For the proposed scheme, the preprocessing of sharing phase includes the calculations of M_{nn}, E, $\tilde{E}$, and $\tilde{E}^{-1}$. The complexities of achieving M_{nn} and E are $O(n^2)$ and $O(mn)$, respectively. Let

$$\hat{n}=\begin{cases} n, \text{ if } m=n \text{ and } det(E)\neq 0 \text{ or } m<n,\\ n+1, \text{ if } m=n \text{ and } det(E)=0,\\ m, \text{ if } m>n \text{ and } det(E)=0. \end{cases} \tag{65}$$

The complexities of generating an invertible $\tilde{E}$ and obtaining $\tilde{E}^{-1}$ are both $O(\hat{n}^3)$. Consequently, the complexity of preprocessing of sharing phase is calculated as $O(n^2+mn+\hat{n}^3+\hat{n}^3)=O(\hat{n}^3)$. Note that, $\tilde{E}$ and $\tilde{E}^{-1}$, which can be derived in a prior, are calculated only once and used to encode all the secret pixels. When encoding $\hat{n}$ secret pixels, $H_j=\tilde{E}^{-1}\times S_j^{(\hat{n})}$ is computed and then used in $f^{\hat{n}}(x_i)\times H_j$ to obtain the shared pixels. To facilitate the encoding efficiency, $f^{\hat{n}}(x_i)\times\tilde{E}^{-1}$ is only computed once during the preprocessing stage for all secret pixels. Then, the result is multiplied with the $\hat{n}$ secret pixels $S_j^{(\hat{n})}$ to retrieve the shared pixels. In this case, the complexity of encoding $\hat{n}$ secret pixels is $O(\hat{n}^2)$. In recovery phase, the revealing vector w_i for each qualified set can be calculated in advance in preprocessing with complexity of $O(\hat{k}\hat{n})$, where $\hat{k}$ is the average number of nodes in a qualified subset. Note that, the revealing vector only needs to be solved once and then it is directly applied to all shadow pixels. The total complexity of obtaining m reveal vectors is $O(m\hat{k}\hat{n})$. Moreover, the decoding complexity of recovering m secret pixels is $O(m\hat{k})$.

E. Amount of Data Transmission

The amount of data transmitted is discussed. For a fair comparison, a secret number is encoded for the (k,n) threshold. We calculate the quantities of shadows that are delivered in sharing and recovering phases. For the proposed method, the total number of shadows used in sharing phase is

$$\hat{n}=\begin{cases} n, \text{ if } m=n \text{ and } det(E)\neq 0 \text{ or } m<n,\\ n+1, \text{ if } m=n \text{ and } det(E)=0,\\ m, \text{ if } m>n \text{ and } det(E)=0. \end{cases} \tag{66}$$

Among the $\hat{n}$ shadows, the first m shadows are delivered to the nodes and the remaining $(\hat{n}-m)$ shadows are made public if $\hat{n}>m$. In recovering phase, k shadows from nodes are required to decode a secret. Note that, additional public shadows would be needed for recovering. Since these shadows are publicly accessible, there is no need to transmit them. As m users are engaged, the total number of shadows being transmitted in recovery phase is mk. In summary, the quantities of shadows that are sent in sharing and recovering phases are m and mk, respectively.

On the other hand, the numbers of shadows that are transmitted in [26] and [27] are also evaluated under the same configuration, as provided in Table VII. When sharing secrets, the proposed method delivers fewer shadows than the first method of [26]. When recovering secrets, all the mentioned techniques transmit the same number of shadows.

TABLE VII
COMPARISON OF DATA TRANSMISSION FOR (k,n) THRESHOLD.

Phase	Scheme			
	Method 1 of [26]	Methods 2, 3 of [26]	[27]	Our
Sharing	$m+n$	m	m	m
Recovering	mk	mk	mk	mk

F. Sharing Capacity

Sharing capacity refers to the amount of secret information that was encoded by a sharing technique. Current DSSP methods are designed for numerical secrets, while the proposed method is suitable for image data. To fairly compare our technique with previous methods, we adopt the number of secrets that are encrypted by a sharing scheme as the measurement of sharing capacity.

For existing (k,n) DSSPs [26] [27], a secret is encoded into a corresponding collection having k nodes. There are totally $\binom{n}{k}$ different k-node collections. Thus, the numbers of shared secrets by the approaches in [26] [27] are $\binom{n}{k}$. On the other hand, the proposed technique can implement the same type of (k,n) scheme as [26] [27]. At this time, the number of encoded secret images is $\binom{n}{k}$ as well. Further, our method can adopt any collection that contains $k,k+1,\cdots,n$ nodes to share a different secret image for the (k,n) threshold.

TABLE VI
COMPARISON OF COMPUTATION COMPLEXITY.

Scheme	Sharing		Recovery	
	Preprocessing	Encoding	Preprocessing	Decoding
Ref. [26], Method 1	$O(k^3)$	$O(mk)$	$O(k^3)$, -	$O(mk)$, $O(mk\log^2 k)$
Ref. [26], Methods 2, 3	$O(k^3n^3)$	$O(n^2+mk)$	$O(k^3)$, -	$O(mk)$, $O(mk\log^2 k)$
Ref. [27]	$O(1)$	$O(k+\sum_{i=k+1}^{n}\binom{i}{k-1})$	-	$O(m)$
Our	$O(\hat{n}^3)$	$O(\hat{n}^2)$	$O(m\hat{k}\hat{n})$	$O(m\hat{k})$

TABLE IX
FEATURE COMPARISON AMONG RELATED SCHEMES.

Scheme	Feature					
	Type of Secret	Distributed Multi-User	Access Structure	Storage Overhead	Sharing Capacity	Shadow Construction
Ref. [7]	Image	No	(k,n)	-	-	Direct
Ref. [18]	Image	No	(k,n)	-	-	Direct
Ref. [26]	Numerical	Yes	(k,n)	High	Low	Direct
Ref. [27]	Numerical	Yes	(k,n)	High	Low	Step-by-Step
Our	Image	Yes	GAS	Low	High	Direct

TABLE VIII
COMPARISON OF SHARING CAPACITY.

Scheme	Number of secrets	
	(k,n) Threshold	GAS
Ref. [26]	$\binom{n}{k}$	N/A
Ref. [27]	$\binom{n}{k}$	N/A
Our	$\binom{n}{k}, \sum_{i=k}^{n}\binom{n}{i}$	m

Therefore, the maximum number of encrypted secrets grows to $\binom{n}{k}+\binom{n}{k+1}+\cdots+\binom{n}{n}=\sum_{i=k}^{n}\binom{n}{i}$. For any $k<n$, we have $\sum_{i=k}^{n}\binom{n}{i}>\binom{n}{k}$. Our method offers enhanced sharing capacity for the (k,n) case. Moreover, the proposed scheme is applicable to GAS. The number of encrypted secret images for GAS is m, where m is the number of qualified subsets. Table VIII summarizes the above analysis.

G. Influence of GAS structure

We examine the influence of GAS structure on the sharing capacity of a shadow and the dimension of target matrix. The GAS structure is represented by the qualified subsets. As the quantity of authorized subsets in GAS grows, the GAS structure becomes more complex. The sharing capacity of a shadow is evaluated by $\mathcal{C}=m/n$ where m is the number of secrets (i.e., number of authorized subsets in GAS) and n is the quantity of shadows.

When m increases (i.e., a more complicated GAS structure is employed), the sharing capacity of a shadow and the dimension of target matrix are analyzed from the following two scenarios. (1) For the case of $m<n$, (i) when m rises to $\overline{m}$ with $\overline{m}<n$, the sharing capacity of a shadow grows from m/n to $\overline{m}/n$, and the dimension of target matrix remains the same (i.e., $n\times n$). On the other hand, (ii) when m increases to $\overline{m}$ with $\overline{m}>=n$, we have n shadows delivered to the nodes and $(\overline{m}-n)$ virtual shadows that are made public. The number of shadows is still n since the virtual shadows are not considered. As a result of this, the sharing capacity of a shadow also goes up to $\overline{m}/n$. If $\overline{m}=n$ and the corresponding initial target matrix is invertible, the dimension of target matrix keeps unchanged (i.e., $n\times n$). If $\overline{m}=n$ and the corresponding initial target matrix is not invertible, or $\overline{m}>n$, the dimension of target matrix grows to $(n+1)\times(n+1)$ or $\overline{m}\times\overline{m}$, respectively. (2) For the situation of $m>=n$, when m grows to $\overline{m}$, the sharing capacity of a shadow increases to $\overline{m}/n$ since the $(\overline{m}-n)$ virtual shadows are not regarded, and the dimension of target matrix becomes $\overline{m}\times\overline{m}$.

H. More Comparison and Discussion

When compared with existing SIS techniques [7] [18] and DSSP methods [26] [27], the proposed method achieves the following benefits, as depicted in Table IX.

- **Oriented for distributed multi-user environment.** Existing SIS methods mainly consider the scenario that the dealer has direct and secure channels with users. Additionally, only one secret image is shared. The distributed multi-user scenario cannot be implemented. However, the proposed method is suitable for this scenario.
- **Suitable for GAS.** Previous DSSP approaches [26] [27] are limited to (k,n) threshold. Complicated sharing strategy, such as GAS, cannot be implemented. Whereas, the proposed technique is designed for GAS.
- **Improved storage overhead.** Existing DSSP methods [26] [27] would store multiple shadows in a node. For some special cases, the number of shadows delivered to a node would be dramatically large. However, only one shadow is distributed to each node by our technique.
- **Enhanced sharing capacity.** For the (k,n) threshold, the maximum number of secrets can be increased to $\sum_{i=k}^{n}\binom{n}{i}$ by the proposed technique. While existing methods [26] [27] only encrypt $\binom{n}{k}$ secrets. The sharing capacity is boosted.

- **Direct shadow construction.** The DSSP in [27] generates shadows in a step-by-step manner. A smaller scheme is iteratively employed to build a larger scheme. Whereas, the proposed approach can directly create all desired shadows.

VI. Conclusion

A DM-SIS was introduced to safeguard sensitive images for distributed multi-user scenario. To produce shadows, a base matrix with Vandermonte coordinates is constructed. Based on the GAS, an initial target matrix is built by linearly combining the vectors of base matrix. To ensure the target matrix is invertible, three matrix-adjusting procedures were presented. The shadows are generated and delivered to the corresponding nodes of a network. Every user is able to reveal a secret from specific nodes. When compared with previous DSSPs [26], [27], benefits, such as GAS sharing, improved storage overhead, enhanced sharing capacity, and direct shadow construction, were provided by our method.

References

[1] W. Lu, Y. Xue, Y. Yeung, H. Liu, J. Huang, and Y.-Q. Shi, "Secure halftone image steganography based on pixel density transition," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1137–1149, 2021.

[2] X. Liao, J. Yin, M. Chen, and Z. Qin, "Adaptive payload distribution in multiple images steganography based on image texture features," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 897–911, 2022.

[3] B. Chen, W. Lu, J. Huang, J. Weng, and Y. Zhou, "Secret sharing based reversible data hiding in encrypted images with multiple data-hiders," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 978–991, 2022.

[4] Z. Yin, Y. Peng, and Y. Xiang, "Reversible data hiding in encrypted images based on pixel prediction and bit-plane compression," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 992–1002, 2022.

[5] Z. Hua, Y. Wang, S. Yi, Y. Zheng, X. Liu, Y. Chen, and X. Zhang, "Matrix-based secret sharing for reversible data hiding in encrypted images," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 3669–3686, 2023.

[6] Y. Zhang, R. Zhao, X. Xiao, R. Lan, Z. Liu, and X. Zhang, "Hf-tpe: High-fidelity thumbnail-preserving encryption," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 3, pp. 947–961, 2022.

[7] C.-C. Thien and J.-C. Lin, "Secret image sharing," *Computers & Graphics*, vol. 26, no. 5, pp. 765–770, 2002.

[8] G. Shen, F. Liu, Z. Fu, and B. Yu, "Perfect contrast xor-based visual cryptography schemes via linear algebra," *Designs, Codes and Cryptography*, vol. 85, no. 1, pp. 15–37, 2017.

[9] Z. Fu, Y. Cheng, and B. Yu, "Perfect recovery of xor-based visual cryptography scheme," *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2367–2384, 2019.

[10] X. Wu, J. Fang, and W. Q. Yan, "Contrast optimization for size invariant visual cryptography scheme," *IEEE Transactions on Image Processing*, vol. 32, pp. 2174–2189, 2023.

[11] X. Wu and X. Feng, "Size invariant visual cryptography schemes with evolving threshold access structures," *IEEE Transactions on Multimedia*, vol. 26, pp. 1488–1503, 2024.

[12] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[13] X. Jia, Y. Guo, X. Luo, D. Wang, and C. Zhang, "A perfect secret sharing scheme for general access structures," *Information Sciences*, vol. 595, pp. 54–69, 2022.

[14] X.-Z. Xie, C.-C. Chang, and C.-C. Lin, "Reversibility-oriented secret image sharing mechanism with steganography and authentication based on code division multiplexing," *IET Image Processing*, vol. 13, no. 9, pp. 1411–1420, 2019.

[15] Y.-Y. Lin and R.-Z. Wang, "Scalable secret image sharing with smaller shadow images," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 316–319, 2009.

[16] Y.-X. Hu and Y.-N. Liu, "A progressively essential secret image sharing scheme using hierarchy shadow," *Journal of Information Security and Applications*, vol. 47, pp. 371–376, 2019.

[17] M. K. Sardar, J. Pramanik, and A. Adhikari, "(t, k, n) regional secret image sharing over finite fields," *Signal Processing*, p. 109082, 2023.

[18] C.-N. Yang, P. Li, and H.-C. Kuo, "(k, n) secret image sharing scheme with privileged set," *Journal of Information Security and Applications*, vol. 73, p. 103413, 2023.

[19] Y. Liu, C. Yang, and Q. Sun, "Thresholds based image extraction schemes in big data environment in intelligent traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3952–3960, 2021.

[20] X. Yan, Y. Lu, C.-n. Yang, X. Zhang, and S. Wang, "A common method of share authentication in image secret sharing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2896–2908, 2020.

[21] L. Xiong, X. Zhong, C.-N. Yang, and X. Han, "Transform domain-based invertible and lossless secret image sharing with authentication," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2912–2925, 2021.

[22] X. Yan, L. Li, L. Sun, J. Chen, and S. Wang, "Fake and dishonest participant immune secret image sharing," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 4, pp. 1–26, 2023.

[23] M. Naor and A. Shamir, "Visual cryptography," *Lecture Notes in Computer Science*, vol. 950, no. 1, pp. 1–12, 1995.

[24] S. J. Shyu and M. C. Chen, "Minimizing pixel expansion in visual cryptographic scheme for general access structures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 9, pp. 1557–1561, 2015.

[25] X. Jia, T. Yu, X. Luo, D. Wang, and H. Zhou, "Maximizing contrast in xor-based visual cryptography schemes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 12, pp. 12 849–12 861, 2024.

[26] M. Soleymani and H. Mahdavifar, "Distributed multi-user secret sharing," *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 164–178, 2021.

[27] R. De Prisco, A. De Santis, and F. Palmieri, "Improved protocols for distributed secret sharing," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 3558–3571, 2023.

[28] C. Yang, T. Chen, K. Yu, and C. Wang, "Improvements of image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 80, no. 7, pp. 1070–1076, 2007.

[29] G. Ateniese, C. Blundo, A. De Santis, and D. Stinson, "Visual Cryptography for General Access Structures," *Information and Computation*, vol. 129, no. 2, pp. 86–106, 1996.

Xiaotian Wu received the Ph.D. degree in computer science from the School of Information Science and Technology, Sun Yat-sen University, in 2013. He was a visiting scholar at Auckland University of Technology in 2017. He is currently an Associate Professor with the College of Cyber Security, Jinan University, China. His research interests include visual cryptography, secret image sharing, data hiding, and multimedia security.

Yuyang Xiong received the master's degree from the Department of Computer Science, Jinan University, China. Her research interests are secret sharing and multimedia security.

Bing Chen received the PhD degree in computer science and technology from the Sun Yat-sen University, Guangzhou, China, in 2020. His research interests include multimedia security, information hiding, and secret sharing.

Ching-Nung Yang received the B.S. and M.S. degrees in telecommunication engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1983 and 1985, respectively, and the Ph.D. degree in electrical engineering from National Cheng Kung University, Tainan City, Taiwan, in 1997. He is currently a Full Professor with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. His research interests include coding theory, information security, and cryptography. Prof. Yang is a Fellow of IET.

WeiQi Yan is with AUT computer science, his expertise covers intelligent surveillance, deep learning, robotics, computer vision, and multimedia computing. Dr. Yan has served as an Associate Editor of ACM Transactions on Multimedia Computing, Communications and Applications (TOMM), an Associate Editor of Frontiers in Neuroscience, an Associate Editor of Springer Nature Computer Science, the Editor-in-Chief (EiC) of the International Journal of Digital Crime and Forensics (IJDCF), and he has worked as an exchange computer scientist between the Royal Society Te Apārangi (RSNZ) and the Chinese Academy of Sciences (CAS) in China. He is a guest (adjunct) professor at the Chinese Academy of Sciences and has been a visiting professor at the University of Auckland in New Zealand and the National University of Singapore. In 2022, Dr. Yan was recognised as one of the world's top 2% cited scientists by Stanford University, USA. He currently holds the position of Chair of ACM Multimedia Chapter of New Zealand and a member of the ACM, a senior member of the IEEE and a TC member of the IEEE.

Qing-Yu Peng is currently a Professor with the Department of Computer Science, Jinan University, China. His research interests include image processing and high precision measurement.