Optimal Access Structure Partition Methods for Image Secret Sharing

Xiaotian Wu, *Member, IEEE*, Li Tang, Zhihua Xia, Ching-Nung Yang, *Senior Member, IEEE*, WeiQi Yan, *Senior Member, IEEE*

***Abstract*—Visual cryptography scheme (VCS) and polynomial-based secret image sharing (PSIS) are two primary types of secret sharing for protecting images. VCS and PSIS have their respective pros and cons. For VCS, the benefits of perfect security and easy decoding are provided. But it suffers from the limitations of lossy secret recovery and binary image-oriented. PSIS can deal with grayscale/color images and offers lossless secret reconstruction. Whereas, the secret decoding is computationally intensive (i.e., $\mathcal{O}(k \log^2 k)$ for (k, n) threshold) and the residual-image problem in PSIS compromises the security. In this paper, we are motivated to investigate a sharing technique that can preserve the advantages of both VCS and PSIS. Differing from existing VCS and PSIS, the proposed sharing method is accomplished based on the access structure partition (ASP) result. Essentially, an ASP guided image secret sharing approach is developed and three optimal ASP algorithms are designed. When compared with existing partition method, significant improvement is offered by our partition techniques especially for the (k, n) threshold with a larger n. Take the $(2, 15)$, $(2, 18)$, and $(4, 12)$ thresholds for example, the numbers of involved sub-access structures by our method are 4, 5, and 19, while the quantities by existing approach are 8, 10, and 45. The percentages of improvement are 100%, 100%, and 137%. Further, based on the partition result from ASP algorithms, we can employ (k, k) probabilistic VCS (PVCS) to constitute a (k, n) sharing method for encoding gray-level/color images. Experiments are demonstrated to confirm the effectiveness of the sharing method and ASP algorithms. Meanwhile, comparisons are included to show that the merits of perfect security, low decoding complexity (i.e., $\mathcal{O}(d)$), lossless secret recovery (i.e., PSNR$= \infty$, SSIM$= 1$), and grayscale/color image-oriented are provided by our sharing method.**

***Index Terms*—Secret sharing, visual cryptography scheme, secret image sharing, access structure partition.**

I. Introduction

Data encryption [1]–[4], data hiding [5]–[9], and secret sharing (SS) have been considered as central building blocks in various multimedia security applications [10]–[13]. Among them, the well-known (k, n) SS [14] splits a secret into n shadows (also called shares) in such a way that any k out of n shadows can decode the secret. Whereas any collection having fewer than k shadows is unable to attain any information about the secret. SS is further extended to image domain and the technique of image secret sharing (ISS) are then developed. Based on the format of secret image, ISS is roughly divided into two categories: visual cryptography scheme (VCS) and polynomial-based secret image sharing (PSIS).

VCS is a lightweight sharing approach designed particularly for binary images. A (k, n) VCS encodes a binary image into n random-looking shadows by using base matrices. VCS delivers the convenience of easy decoding even without the use of computing devices. It is feasible to visually reveal the secret by printing shadows on transparencies and stacking them together. Nevertheless, the secret is reconstructed in a lossy manner. To accomplish a VCS, the base matrices satisfying security and contrast requirements are required. In addition, the security condition guarantees that VCS is perfectly secure. With the fundamentals of VCS [14], various techniques with different functionalities were developed. Generally, some key properties of VCS, such as pixel expansion, recovered image quality, and access structure, are discussed. Since a secret pixel is substituted by m sub-pixels of a shadow, the generated shadow is m times of the secret. m is referred to as pixel expansion. To deal with pixel expansion problem, probabilistic VCS (PVCS) [15] and VCS using integer linear programming [16] were presented. As the decoding of VCS can be accomplished physically by stacking, the background of recovered secret would be dramatically darkened when more shadows are superimposed. To improve visual quality, XOR-based VCS (XVCS) [16]–[19] was introduced. Note that, perfect secret recovery is provided by the (k, k) XVCS. To encrypt gray-level images, a halftoning technique is utilized for converting a gray-level image into an approximate binary image and existing VCS can be applied to accomplish the work of creating shadows [20]. Halftone VCS [21] employs the void and cluster algorithm to encode a binary image into halftone shares carrying significant visual information. Also, a variety of modified halftoning methods, including parallel error diffusion [22], multi-scale error diffusion [23], analysis-by-synthesis framework [24] and bit-slicing [25], are integrated with VCS for producing visually pleasing halftone shadows. XVCS [16] [17] is also utilized to provide perfect recovery so that grayscale images can be tackled. Moreover, VCS methods designed for different access structures, such as

This work was partially supported by National Key Research and Development Program of China (Grant No. 2022YFB3103100), National Natural Science Foundation of China (Grant Nos. 62122032, U23B2023), and Guangdong Key Laboratory of Data Security and Privacy Preserving (Grant No. 2023B1212060036). (Corresponding authors: Zhihua Xia and Xiaotian Wu)

Xiaotian Wu and Li Tang are with the College of Cyber Security, College of Information Science and Technology, and Guangdong Key Laboratory of Data Security and Privacy Preserving, Jinan University, Guangzhou, China. E-mail: wxt.sysu@gmail.com

Zhihua Xia is with the College of Cyber Security, Engineering Research Center of Trustworthy AI, Ministry of Education, Jinan University, Guangzhou, China. E-mail: xia_zhihua@163.com

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

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

general access structure [26], evolving access structure [27], and essential access structure [28], were developed.

PSIS is capable of sharing grayscale/color images by utilizing Shamir's SS [29]. The pioneer work of PSIS [30] adopts a $(k-1)$ degree polynomial where the k coefficients of polynomial are used to conceal k gray-level secret pixels. With the aid of Lagrange interpolation, the secret can be losslessly decoded from shadows. Further, the decoding complexity becomes computationally expensive as Lagrange interpolation is employed. Since then, various PSIS methods, focusing on steganography [31], authentication [32], [33], reversible and lossless sharing [34], [35], cheating-preventing [36], outsourcing computation [37], etc., were developed. Counting-based SS [38]–[43] was also investigated for various multimedia applications. It is worthwhile to point out that the majority of PSIS techniques suffer from a security vulnerability (i.e., residual-image problem) such that some residual information about the secret will remain on the shadows when PSIS is performed straight to an image without using permutation.

In summary, although VCS provides the benefits of easy decoding and perfect security, the drawbacks, such as lossy secret recovery and restriction to binary images, greatly limits the application scenario. On the contrary, PSIS deals with grayscale/color images and offers distortion-free secret reconstruction. However, the secret decoding requires considerable computational complexity. In addition, it suffers from the residual-image problem which compromises the security. In this paper, we are motivated to design a novel image secret sharing technique that can preserve the advantages of both VCS and PSIS, i.e., low decoding complexity, lossless secret reconstruction, grayscale/color image-oriented, and perfect security. In contrast to VCS and PSIS, our sharing method is constituted based on the access structure partition (ASP) result provided by the proposed ASP algorithms. Similar works based on access structure decomposition can be found in [18], [26], [44]. Li *et al.* [18] discussed the implementation of essential VCS by using access structure decomposition. Shyu [26] developed two effective VCS methods for general access structures based on access structure partition and threshold VCSs. Shen *et al.* [44] presented a heuristic ASP method to build an image secret sharing scheme. However, existing techniques [18], [26], [44] do not investigate the ASP problem systematically. More significantly, since the optimality of ASP problem is not fully explored, the visual performance and shadow size might be inferior. Herein, the ASP problem is well-defined and three alternative algorithms for solving ASP problem are introduced. The sharing technique based on the ASP result is presented as well. Although the ASP approaches are primarily designed for tackling the combinatorial optimization problem (i.e., ASP problem), they are regarded as a central component for constituting the sharing approach. These methods indeed fall in the topic of image secret sharing. When compared with current VCS [16], [27] and hybrid encryption [45], [46], our sharing method is constituted based on the ASP result offered by the proposed partition algorithms. While existing VCS [16], [27] employs base matrices and random number generator, and hybrid encryption [46] combines VCS with watermarking. The kernel mechanism of our sharing approach is completely different from the current approaches. Contribution of this paper is summarized below.

- A general image secret sharing approach based on ASP result is devised. Given the partition result of a desired (k, n) threshold, the proposed sharing technique adopts the (k, k) PVCS to constitute a (k, n) scheme. As the (k, k) PVCS is actually an XVCS that can losslessly recover the secret via XOR operation, the proposed sharing method provides lossless secret reconstruction with low decoding complexity. Meantime, the proposed sharing scheme is perfectly secure and applicable for grayscale/color images.
- To tackle the ASP problem, a theoretical model is presented, as well as the concept of sub-AS representation and candidate representation generation. Three distinct ASP algorithms (i.e., the techniques based on dynamic programming, beam search, and genetic algorithm with variable-length chromosomes) are designed to offer optimal division results for the sharing approach. When compared with existing heuristic partition algorithm [44], our methods significantly reduce the partition number.

The application scenario is enriched by our sharing technique. Consider the smart remote health application based on image secret sharing. In this application, clinical images form a major part of the health data. As medical images are confidential and sensitive, they should be encrypted. In this case, properties, such as perfect security, lossless image decoding and grayscale/color image-oriented, are required. In addition, various Internet of Health Things (IoHT) devices, such like wearable devices and mobile phones, are utilized in smart remote health application. These IoHT devices have limited computational capability. Thus, low decoding complexity is preferred. According to the foregoing analysis, our sharing method is more suitable for applications.

This paper is organized as follows. Section II describes the motivation of this paper, as well as the proposed image secret sharing method. Sub-access structure (sub-AS) representation and its generation are presented in Section III. Three distinct ASP algorithms are formulated in Sections IV-VI. Experimental results and comparisons are illustrated in Section VII. Section VIII concludes our work.

II. Motivation and General Sharing Methodology

A. Motivation

Characteristics of VCS and PSIS are formulated in Table I. VCS enjoys the benefit of easy decoding and provides perfect security. The decoding complexity is $\mathcal{O}(1)$ when using Boolean OR operation. The security of (k, n) VCS is guaranteed in such a way that any collection having $(k-1)$ or fewer shadows give no clue about the secret. However, VCS is limited to binary images. As gray-level/color images are frequently, the application scenario of VCS is greatly restricted. VCS also suffers from the defect of lossy reconstruction when Boolean OR/stacking operation is employed to decode the secret. In addition, the visual quality of VCS reduces dramatically as more shadows are overlaid.

TABLE I
CHARACTERISTICS OF VCS AND PSIS.

Method	Characteristic			
	Decoding complexity	Secret format	Recovery type	Security
VCS	$\mathcal{O}(1)$	B	Lossy	Perfectly secure
PSIS	$\mathcal{O}(k \log^2 k)$	G/C	Lossless	Residual-image problem

B: Binary, G/C: Grayscale/Color

Differing from VCS, PSIS adopts Shamir's SS [29] to deal with grayscale/color images. The secret can be decoded without distortion. Unfortunately, since Lagrange interpolation is utilized in secret recovery, the decoding complexity of (k, n) PSIS is proportional to $\mathcal{O}(k \log^2 k)$, which is computationally consuming. Notably, the security of PSIS might be compromised when PSIS is directly applied to a secret image without using permutation. It suffers from the so-called residual-image problem such that some confidential information is leaked on shadows when scramble is not adopted. The reason is that the shadow ID is involved in shadow generation, and the neighbor pixels have the approximate values. A $(2, 4)$ PSIS is given in Fig. 1 to illustrate the residual-image problem. Four shadows using $ID = 1$, 2, 3, and 4, respectively, of $(2, 4)$ PSIS without permuting the secret are shown in Figs. 1 (b). One can observe that the residual information is disclosed on shadows.

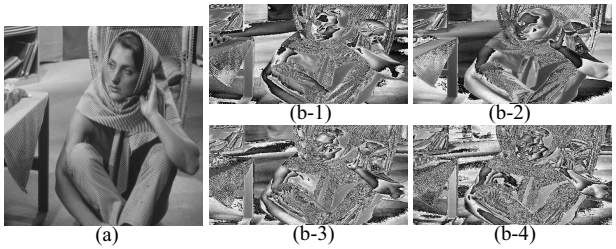

Fig. 1. The residual image problem of $(2, 4)$ PSIS when permutation is not used. (a) Secret image, (b) four shadows.

In general, VCS obtains the merits of easy decoding and perfect security. But it is limited to binary images and recovers the secret in a lossy manner. In contrast to VCS, PSIS is capable of encoding gray-level/color image and provides distortion-free decryption, while the complexity of decoding is high and the residual-image problem further compromises the security. In this paper, we are motivated to explore an image secret sharing technique that can offer the advantages of both VCS and PSIS (i.e., easy decoding, perfect security, grayscale/color image-oriented, and lossless decoding).

B. General Sharing Methodology

We introduce an ASP guided image secret sharing approach. The basic idea is to use the division result provided by the ASP algorithms to implement the desired (k, n) threshold. More concretely, our ASP methods are designed to separate a (k, n) threshold into several sub-AS representations. According to the partition result, simpler schemes (i.e., the (k, k) PVCSs) can be employed to realize a more complicated method (i.e., the (k, n) scheme). As the (k, k) PVCS is actually an XOR-based approach which is capable of recovering a secret without distortion, the (k, n) sharing method comprised by (k, k) PVCSs can deal with grayscale/color images in a lossless manner. Diagram of the proposed ASP guided sharing scheme is depicted in Fig. 2. The core steps are presented as follows.

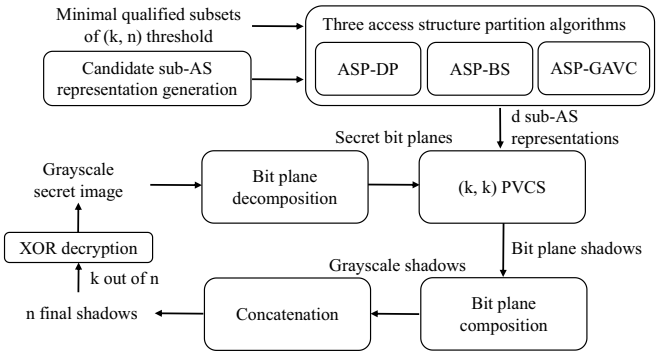

Fig. 2. Diagram of the proposed ASP guided image secret sharing.

(1) Access structure partition. A (k, n) threshold can be represented by a collection of minimal qualified subsets Γ_0. A qualified subset is the set of participants that can decode the secret, whereas, a forbidden one refers to the collection that cannot learn any information about the secret. A qualified subset is called minimal if any proper subset of it is a forbidden one. By using the proposed ASP algorithms (described in Sections IV-VI), Γ_0 is separated into d parts in such a way that all these d subsets can cover the whole collection. Each subset is termed a sub-AS and can be denoted by a representation containing k equivalent relationships. Once a participant p is equivalent to another q (depicted by $p \sim q$) in a sub-AS representation, the same shadow will be assigned to them.

Let Υ_i be the i-th sub-AS representation consisting of k equivalent relationships $\mathcal{L}_{i,1}, \cdots, \mathcal{L}_{i,k}$ where $\mathcal{L}_{i,j} = \{p_{i,j}^{(1)}, \cdots, p_{i,j}^{(m_j)}\}$ denotes the j-th $(1 \leq j \leq k)$ equivalent relationship of Υ_i having $m_j \geq 1$ different participants, i.e., $p_{i,j}^{(1)} \sim \cdots \sim p_{i,j}^{(m_j)}$. It should be noted that each equivalent relationship contains at least one participant. When only one member is involved, there is no additional equivalent participant correlated to this member. Each time we can derive a k-tuple qualified subset from Υ_i by selecting a participant from each relationship. As a result, a total of $(m_1 \times \cdots \times m_k)$ different qualified subsets are generated from Υ_i via the following set generation procedure

$$\Gamma_i = SetGen(\Upsilon_i). \tag{1}$$

In the above equation, Γ_i is referred to as a sub-AS and it can be represented by a sub-AS representation Υ_i. Example 1 demonstrates the partition of a $(3, 4)$ threshold.

Example 1. *Consider the partition of a* $(3, 4)$ *threshold. The collection of minimal qualified subsets is* $\Gamma_0 = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}$. Γ_0 *can be divided into* 2 *sub-ASs:* $\Gamma_1 = \{\{1, 2, 3\}, \{1, 2, 4\}\}$ *and* $\Gamma_2 = \{\{1, 3, 4\}, \{2, 3, 4\}\}$. *The first sub-AS is represented by* Υ_1 *having* 3 *equivalent relationships:* $\mathcal{L}_{1,1} = \{1\}$, $\mathcal{L}_{1,2} = \{2\}$,

and $\mathcal{L}_{1,3} = \{3, 4\}$. *And the second sub-AS is denoted by* Υ_2 *with* $\mathcal{L}_{2,1} = \{1, 2\}$, $\mathcal{L}_{2,2} = \{3\}$, *and* $\mathcal{L}_{2,3} = \{4\}$.

(2) Secret image decomposition and bit plane sharing. For an 8-bit gray-level image S, it can be decomposed into 8 bit planes $S^{(1)}, \cdots, S^{(8)}$. For each sub-AS representation Υ_i $(1 \leq i \leq d)$, each bit plane $S^{(x)}$ $(1 \leq x \leq 8)$ is encoded by a (k, k) PVCS to produce k bit plane shadows $SH_{i,1}^{(x)}, \cdots, SH_{i,k}^{(x)}$. Since any sub-AS representation comprises k equivalent relationships, the j-th bit plane shadow $SH_{i,j}^{(x)}$ $(1 \leq j \leq k)$ of the (k, k) PVCS is distributed to the participants involving in the j-th relationship. Specifically, let $\mathcal{L}_{i,j} = \{p_{i,j}^{(1)}, \cdots, p_{i,j}^{(m_j)}\}$ be the j-th relationship of Υ_i, the bit plane shadow $SH_{i,j}^{(x)}$ is assigned to all involved participants $p_{i,j}^{(1)}, \cdots, p_{i,j}^{(m_j)}$. Note that, a secret bit plane would be encoded by d different (k, k) PVCSs [15] which correspond to d sub-AS representations. As the d sub-AS representations can cover all minimal qualified subsets of the desired threshold, the (k, n) scheme can be implemented.

(3) Bit plane composition and final shadow concatenation. For every sub-AS representation, each participant would receive 8 bit plane shadows. Thus, he can compose the corresponding bit plane shadows together to form a gray-level shadow. As there exist d representations, each participant obtains d gray-level shadows. To ease the management of shadows, these d shadows are concatenated together to derive a final shadow. In addition, the color secret image can be shared by using the same methodology.

We remark that a (k, k) PVCS is also an XVCS which is capable of recovering the secret losslessly. As a result of this, the gray-level secret image can be decoded perfectly from the corresponding grayscale shadows by the pixel-wise XOR operation. Additionally, only one pixel-wise XOR operation is performed on the gray-level shadow pixels for recovering one gray-level secret pixel. Since there exist d sub-AS representations, the decoding complexity is $\mathcal{O}(d)$. Furthermore, the proposed scheme can be proved to be perfectly secure. The merits of both VCS and PSIS are preserved in the proposed sharing approach. Example 2 shows the sharing process based on the partition result given in Example 1.

Example 2. *Consider the proposed scheme for* $(3, 4)$ *threshold based on the partition result given in Example 1. The encoding of the first secret bit plane* $S^{(1)}$ *is described below. One can use a* $(3, 3)$ *PVCS for the first representation* Υ_1. *Let* $SH_{1,1}^{(1)}, SH_{1,2}^{(1)}, SH_{1,3}^{(1)}$ *be the* 3 *shadows generated from a* $(3, 3)$ *PVCS with secret bit plane* $S^{(1)}$. $SH_{1,1}^{(1)}$ *and* $SH_{1,2}^{(1)}$ *are distributed to participants* 1 *and* 2, *respectively.* $SH_{1,3}^{(1)}$ *is given to participants* 3 *and* 4 *as they are equivalent participants. In this case, both* $\{1, 2, 3\}$ *and* $\{1, 2, 4\}$ *can recover the secret bit plane* $S^{(1)}$. *Similarly, another* $(3, 3)$ *PVCS with shadows* $SH_{2,1}^{(2)}, SH_{2,2}^{(2)}, SH_{2,3}^{(2)}$ *is employed for the second representation* Υ_2. *Herein,* $SH_{2,1}^{(2)}$ *is delivered to the equivalent participants* 1 *and* 2. *While* $SH_{2,2}^{(2)}$ *and* $SH_{2,3}^{(2)}$ *are provided to participants* 3 *and* 4, *respectively. Consequently, both* $\{1, 3, 4\}$ *and* $\{2, 3, 4\}$ *are capable of disclosing* $S^{(1)}$. *According to the two situations, participants* 1, 2, 3, *and* 4 *receive the shadows* $\{SH_1^{(1)}, SH_1^{(2)}\}$, $\{SH_2^{(1)}, SH_1^{(2)}\}$, $\{SH_3^{(1)}, SH_2^{(2)}\}$, *and* $\{SH_3^{(1)}, SH_3^{(2)}\}$, *respectively. The* $(3, 4)$ *case is realized by two different* $(3, 3)$ *PVCSs. The remaining* 7 *bit planes are encoded by using the same approach. The corresponding bit plane shadows are combined and concatenated to form the final shadows. Any* 3 *out of* 4 *final shadows can be XOR-ed to disclose the secret. The merits of gray-level/color image-oriented, easy decoding, lossless reconstruction, and perfect security are provided.*

To comprehend the motivation and benefits of the proposed sharing method, we compare with VCS and PSIS. For VCS, the gray-level secret image cannot be encoded directly since VCS is only suitable for binary images. The grayscale image might be transformed into halftone image and the $(3, 4)$ *VCS is adopted to generate* 4 *shadows. In this case, the secret is recovered in a lossy manner. For PSIS, one can construct a* 2*-degree polynomial to deal with the gray-level image and* 4 *shadows are generated. Note that, all coefficients of the* 2*-degree polynomial are replaced by secret pixels. Once the image permutation is not employed, the residual-image problem would occur and consequently the security is compromised. In addition, as Lagrange interpolation is employed to recover the coefficients of polynomial, the time complexity is* $\mathcal{O}(k \log^2 k)$, *which is computationally intensive.*

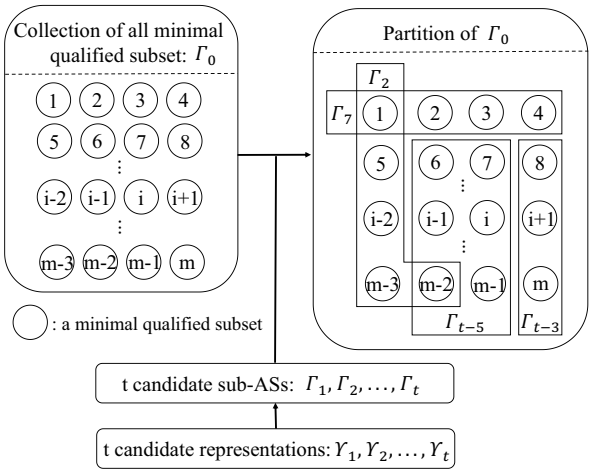

Fig. 3. Illustration of the ASP problem.

C. Key Issue of The Proposed Sharing Method

In the sharing approach, each participant receives d gray-level shadows from the d representations. The size of the final shadow is d times of the secret. Hence, d is expected to be as small as feasible so that the overhead for storing and transmitting shadows can be reduced. Actually, d is the number of sub-AS representations that can cover all minimal qualified subsets of the desired threshold. Such a covering problem is called an access structure partition (ASP) problem, as described in Fig. 3. A theoretical model of ASP problem is presented in Definition 1. A systematic investigation on achieving minimum value of d is further conducted.

Definition 1. *(Access structure partition problem) Let* Γ_0 *be the collection of minimal qualified subsets correlated to a*

(k, n) threshold. Suppose $\Upsilon_1, \cdots, \Upsilon_t$ are the t candidate sub-AS representations that can generate t sub-ASs $\Gamma_1, \cdots, \Gamma_t$ by $\Gamma_i = SetGen(\Upsilon_i)$ where $1 \leq i \leq t$. The ASP problem is to seek the minimum number of sub-AS representations such that they can cover all minimal qualified subsets presented in Γ_0. Formally, it is given by the following minimizing problem:

$$\begin{aligned} &\textit{Minimize } d = u_1 + \cdots + u_t \\ &\textit{Subject to } U_1 \cup \cdots \cup U_t = \Gamma_0 \end{aligned} \tag{2}$$

where

$$u_i = \begin{cases} 1, \ \textit{if } \Upsilon_i \textit{ is selected,} \\ 0, \ \textit{otherwise,} \end{cases} \tag{3}$$

and

$$U_i = \begin{cases} SetGen(\Upsilon_i), \ \textit{if } \Upsilon_i \textit{ is selected,} \\ \emptyset, \ \textit{otherwise.} \end{cases} \tag{4}$$

Herein, u_i is a binary variable to indicate whether Υ_i is chosen or not and U_i is the sub-AS generated from Υ_i.

To tackle this problem, two steps are considered. First, possible candidate sub-AS representations should be identified when the desired threshold is given. The way for obtaining candidates is provided in Section III. Next, a collection of possible candidates, whose cardinality should be minimum, is selected in such a way that it can cover all minimal qualified subsets of the desired threshold. In total, three alternative methods are presented in Sections IV-VI. In addition, other traditional approaches (e.g., branch and bound method and backtracking method) and meta-heuristic algorithms (e.g., particle swarm optimization and ant colony optimization) can be considered to solve the ASP problem.

III. Sub-AS Representation and Its Generation

In this section, a brief representation of a sub-AS is depicted. Then, we show how to generate possible candidate representations for a desired threshold.

A. Sub-AS Representation

For a (k, n) threshold, a sub-AS Γ_i is generated by a representation Υ_i having k equivalent relationships, as given by $\Gamma_i = SetGen(\Upsilon_i)$. Υ_i is denoted as $\{\sigma_i, \beta_i\}$. To be more concrete, $\sigma_i = \{n_1^{(i)}, \cdots, n_k^{(i)}\}$ is a sequence of k numbers $n_1^{(i)}, \cdots, n_k^{(i)}$ satisfying

$$\begin{cases} n_k^{(i)} \geq \cdots \geq n_1^{(i)} \geq 1, \\ n_1^{(i)} + \cdots + n_k^{(i)} = n. \end{cases} \tag{5}$$

Each $n_j^{(i)}$, $1 \leq j \leq k$, indicates the quantity of participants involved in the j-th relationship. Let the n participants be $\{1, \cdots, n\}$. β_i is defined as a permuted sequence of $\{1, \cdots, n\}$. For example, when we have 4 participants $\{1, 2, 3, 4\}$, β_i can be $\{3, 2, 4, 1\}$. Further, $\sigma_i = \{n_1^{(i)}, \cdots, n_k^{(i)}\}$ divides $\beta_i = \{b_1^{(i)}, \cdots, b_n^{(i)}\}$ into k non-overlapping segments by

$$\begin{cases} \{b_1^{(i)}, \cdots, b_{n_1^{(i)}}^{(i)}\}, \\ \{b_{n_1^{(i)}+1}^{(i)}, \cdots, b_{n_1^{(i)}+n_2^{(i)}}^{(i)}\}, \\ \quad \vdots \\ \{b_{n_1^{(i)}+\cdots+n_{k-1}^{(i)}+1}^{(i)}, \cdots, b_{n_1^{(i)}+\cdots+n_{k-1}^{(i)}+n_k^{(i)}}^{(i)}\}. \end{cases} \tag{6}$$

Each element in the above segments is actually a participant. For the j-th ($1 \leq j \leq k$) segment, it contains exactly $n_j^{(i)}$ participants which are equivalent to each other. A segment actually denotes an equivalent relationship. By combining $\sigma_i = \{n_1^{(i)}, \cdots, n_k^{(i)}\}$ with $\beta_i = \{b_1^{(i)}, \cdots, b_n^{(i)}\}$, one can achieve k equivalent relationships which would yield a corresponding sub-AS. For brevity, $\Upsilon_i = \{\sigma_i, \beta_i\}$ is termed a sub-AS representation. The following example demonstrates a representation of $(3, 6)$ threshold.

Example 3. *Consider a sub-AS representation of $(3, 6)$-threshold Υ_i: $\sigma_i = \{1, 2, 3\}$, $\beta_i = \{2, 5, 6, 1, 4, 3\}$. According to σ_i, 3 segments divided from β_i are $\{2\}$, $\{5, 6\}$, $\{1, 4, 3\}$. The corresponding 3 equivalent relationships are $\{2\}$, $\{5 \sim 6\}$, and $\{1 \sim 4 \sim 3\}$. Each time one participant is chosen from every relationship to derive a 3-tuple qualified subset. Hence, a sub-AS having $6 (= 1 \times 2 \times 3)$ qualified subsets $\{2, 5, 1\}$, $\{2, 5, 4\}$, $\{2, 5, 3\}$, $\{2, 6, 1\}$, $\{2, 6, 4\}$, $\{2, 5, 3\}$ is achieved.*

B. Candidate Representation Generation

We demonstrate the way to create possible candidate sub-AS representations for a given threshold. First of all, a collection containing different sequences, denoted as $\mathcal{A} = \{\sigma_1, \cdots, \sigma_{t_1}\}$, is achieved via the sequence generation approach given in Algorithm 1. Essentially, the sequence generation is a recursive algorithm that partitions an integer n into a set of various sequences, where each sequence should meet the condition in (5). Then, Algorithm 2 is utilized to produce candidate sub-AS representations. Let $\mathcal{B} = \{\beta_1, \cdots, \beta_{t_2}\}$ be a set having all possible permutations of $\{1, \cdots, n\}$. For each time, we select one sequence σ_i from $\mathcal{A}$ and one permutation β_j from $\mathcal{B}$ to form $\Upsilon_r = \{\sigma_i, \beta_j\}$. Consequently, a sub-AS is derived as $\Gamma_r = SetGen(\Upsilon_r)$. Let Γ be the union of subsets generated from all previous sub-ASs. If $\Gamma_r \not\subseteq \Gamma$, add Γ_r to Γ and add Υ_r to Θ. Finally, a collection of candidate representations Θ is obtained. Remember that, any Υ_r in Θ is unique, meaning that the corresponding sub-AS generated from Υ_r is different from those based on the remaining elements in Θ.

IV. Partition scheme based on Dynamic Programming

In this section, we introduce an ASP algorithm based on dynamic programming (ASP-DP) to search the minimum number of sub-AS representations.

A. Analysis and Design Concept

The ASP problem illustrates the properties of optimal substructure and overlapping sub-problems. For optimal substructure, the best solution to the overall problem can be built

Algorithm 1 Sequence Generation (SG)

Input: $k, n, \sigma, \mathcal{A}$
Output: $\mathcal{A}$ or $\emptyset$
if $k = 0$ and $n = 0$ **then**
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\sigma\}$
  return $\mathcal{A}$
end if
if $n < k$ or $k < 1$ **then**
  return $\emptyset$
end if
if $\sigma = \emptyset$ **then**
  $start \leftarrow 1$
else
  $start \leftarrow$ last element in σ
end if
for $i = start$ to n **do**
  let $\sigma' \leftarrow \sigma$
  Append i to σ'
  $SG\left(k-1, n-i, \sigma', \mathcal{A}\right)$
end for

Algorithm 2 Candidate Representation Generation

Input: $\mathcal{A}, \mathcal{B}$
Output: Θ
$\Theta \leftarrow \emptyset$, $\Gamma \leftarrow \emptyset$, $r \leftarrow 0$
$t_1 \leftarrow |\mathcal{A}|$, $t_2 \leftarrow |\mathcal{B}|$
for $i = 1$ to t_1 **do**
  for $j = 1$ to t_2 **do**
    $r \leftarrow r + 1$
    $\Upsilon_r \leftarrow \{\sigma_i, \beta_j\}$
    $\Gamma_r \leftarrow SetGen(\Upsilon_r)$
    if $\Gamma_r \not\subseteq \Gamma$ **then**
      $\Gamma \leftarrow \Gamma \cup \{\Gamma_r\}$, $\Theta \leftarrow \Theta \cup \{\Upsilon_r\}$
    end if
  end for
end for

from the optimal solutions of its sub-problems. It implies that if an optimal solution to the whole problem is achieved, the components of a solution that address sub-problems must likewise be optimum for those sub-problems. For overlapping sub-problems, the same sub-problems are encountered repeatedly during the calculation of the overall solution. These two properties ensure that dynamic programming can be adopted to seek the optimal solution of ASP problem.

When dynamic programming is applied, the actions listed below are taken into consideration. At the beginning, sub-problems are identified. The primary ASP problem is split into more manageable and related sub-problems. After that, a recurrence relation is established to describe how the solution to a larger sub-problem can be derived from the solutions of smaller sub-problems. Simultaneously, the base cases, i.e., the solutions for the simplest sub-problems, should be defined to serve as starting points for the recurrence relation. Finally, the primary problem is tackled by recursively invoking functions for sub-problems. To facilitate the calculation, the results of sub-problems are stored in a data structure as they are computed, and retrieve them if needed again.

B. Main Procedure

Let Ω be the AS to be partitioned. $\Theta = \{\Upsilon_1, \cdots, \Upsilon_t\}$ is a collection having t candidate sub-AS representations. The proposed ASP-DP, denoted by ASP-DP$(\Omega, \Theta, i, \tilde{\Phi})$, returns a subset of representations from $\Upsilon_1, \cdots, \Upsilon_i$ in Θ such that they can cover all qualified subsets in Ω. Meantime, the quantity of representations in this returned subset is minimum. Additionally, i indicates the first i representations in Θ are used for the partition problem and $\tilde{\Phi}$ is the collection to record optimal solutions of sub-problems.

The ASP problem can be broken down into simpler sub-problems by considering whether the last candidate representation in Θ is contained in the optimal solution or not. Essentially, the best result can be obtained via the following recurrence relation

$$\begin{aligned} &|\text{ASP-DP}(\Omega, \Theta, i, \tilde{\Phi})| = \min\{|\text{ASP-DP}(\Omega, \Theta, i-1, \tilde{\Phi})|, \\ &|\text{ASP-DP}(\Omega \setminus SetGen(\Upsilon_i), \Theta, i-1, \tilde{\Phi})| + 1\} \end{aligned} \tag{7}$$

where $|\cdot|$ represents the quantity of elements of a collection. An optimal decision for particular Ω and i can be made by choosing the minimum number of sub-AS representations with $(i-1)$ such that they can cover all elements in Ω. Once the i-th representation Υ_i is chosen, Ω is updated as $\Omega \setminus SetGen(\Upsilon_i)$ and the number of selected representations is incremented by 1. Otherwise, Ω remains the same.

Algorithm 3 ASP-DP

Input: $\Omega, \Theta, i, \tilde{\Phi}$
Output: $\Phi_{\Omega,i}$ or $\Phi_{\Omega,i-1}$ or $\Phi_{\Omega',i-1}$ or Error
if $\Omega = \emptyset$ **then**
  return $\emptyset$
else
  if $i \leq 0$ **then**
    return Error
  else if $\Phi_{\Omega,i} \neq \emptyset$ **then**
    return $\Phi_{\Omega,i}$
  else
    $\Phi_{\Omega,i-1} \leftarrow$ ASP-DP$(\Omega, \Theta, i-1, \Phi)$
    $\Omega' \leftarrow \Omega \setminus SetGen(\Upsilon_i)$
    $\Phi_{\Omega',i-1} \leftarrow$ ASP-DP$(\Omega', \Theta, i-1, \Phi) \cup \{\Upsilon_i\}$
    if $|\Phi_{\Omega,i-1}| < |\Phi_{\Omega',i-1}|$ **then**
      return $\Phi_{\Omega,i-1}$
    else
      return $\Phi_{\Omega',i-1}$
    end if
  end if
end if

Algorithm 3 shows the ASP-DP. Memorization is employed to facilitate the calculation. Herein, $\tilde{\Phi}$ is a collection containing optimal solutions for sub-problems (e.g., $\Phi_{\Omega,i}$, $\Phi_{\Omega,i-1}$, $\Phi_{\Omega',i-1} \in \tilde{\Phi}$). Every solution in $\tilde{\Phi}$ is initialized as $\emptyset$. We set $\Omega = \Gamma_0$ and $i = t$ (i.e., ASP-DP$(\Gamma_0, \Theta, t, \tilde{\Phi})$) to obtain the whole optimal solution. For the base case, if $\Omega = \emptyset$, an empty set is returned. If $i \leq 0$ (i.e., no sub-AS representation in Ω can be used), an error is given. If a solution (i.e., $\Phi_{\Omega,i}$) had been calculated, it is obtained from the collection $\tilde{\Phi}$ directly without computation. In addition, $\Phi_{\Omega',i-1}$ and $\Phi_{\Omega,i-1}$ indicate the solutions with and without the i-th representation Υ_i, respectively. An optimal decision is made when the solution has fewer number of sub-AS representations.

V. Partition scheme based on Beam Search

When the number of candidate sub-AS representations is large, ASP-DP is computationally expensive since it explores the overall searching space. It is not easy to seek the optimal solution. In this section, we introduce an ASP algorithm based on beam search (ASP-BS).

A. A General Approach

In contrast to ASP-DP, the proposed ASP-BS is a heuristic search algorithm that efficiently finds approximate solutions

by exploring a limited number of the most promising partial solutions at each step, referred to as the "beam width". By altering the beam width to retain only a small number of potential partial solutions, it drastically lowers the computational cost while still striving for satisfactory outcomes. Although it occasionally misses the globally optimal solution, ASP-BS maintains a balance between accuracy and computational efficiency by keeping only the top candidates. The main concept of ASP-BS is given. At first, the search begins with an initial set of solutions. In each step, the algorithm expands the current solutions by generating potential ones for the subsequent steps. Then, each new solution is scored based on its evaluation function. Only a fixed number, i.e., the beam width, of the highest-scoring solutions are retained for the next step, while less promising ones are pruned. This process repeats until a complete solution emerges, balancing efficiency with the ability to retrieve satisfactory approximations.

Specifically, two main procedures are included in ASP-BS: the potential partial solution generation and the best-w-partial-solution selection where w is the beam width. Potential partial solution generation is established via $UdpSln(\cdot)$ and the best-w-partial-solution selection is accomplished by $BestSln(\cdot)$. $UpdSln(\Phi, \Theta)$ updates a current partial solution Φ by adding every sub-AS representation in Θ to Φ. Consequently, a collection of updated partial solutions Λ_U is obtained. Detailed information of $UpdSln(\Phi, \Theta)$ can refer to Algorithm 4. $BestSln(\Lambda_U, \Gamma_0, w)$ returns a collection of the best w partial solutions Λ from a set of potential partial results Λ_U, as depicted in Algorithm 5, where w is the beam width. Let Γ_{Φ_i} be the collection of minimal qualified subsets that is generated by a partial solution Φ_i. More concretely, for each partial solution Φ in Λ_U, the corresponding Γ_{Φ_i} is obtained by

$$\Gamma_{\Phi_i} = \bigcup_{\Upsilon \in \Phi_i} SetGen(\Upsilon) \tag{8}$$

where Υ denotes a sub-AS in the partial solution Φ_i. The number of overlapping minimal qualified subsets between Γ_0 and Γ_{Φ_i} is then calculated. Larger number of overlapping subsets indicates the corresponding partial solution is more possible to cover Γ_0. Thus, we sort all the partial results based on their numbers of overlapping subsets in a descending order and select the top w solutions as the best ones.

Algorithm 4 Update Solution (UpdSln)

Input: Φ, Θ
Output: Λ_U
1: $\Lambda_U \leftarrow \emptyset$
2: **for** each Υ_i in Θ **do**
3: $\quad \Lambda_U \leftarrow \Lambda_U \cup \{\Phi \cup \Upsilon_i\}$
4: **end for**
5: **return** Λ_U

The ASP-BS is outlined in Algorithm 6 with the inputs of original AS Γ_0, the collection of candidate sub-AS representations Θ, and the beam width w. A collection of initial partial solutions is generated via Step 1, and Step 2 yields a set of best w partial solutions. The iterative process is performed via Steps 3-18 until a feasible solution is discovered. Steps 5-9 demonstrate the potential partial solution generation. For every partial solution Φ_i in Λ, an extra sub-AS representation in Θ is added into Φ_i to create various updated solutions. Among the candidate results, we choose the finest w updated solutions in Step 10. Each of the best w solutions is then examined whether it can cover all the minimal qualified subsets in Γ_0. A feasible solution is returned once the examination succeeds.

Algorithm 5 Best Partial Solutions (BestSln)

Input: Λ_U, Γ_0, w
Output: Λ
1: **for** each Φ_i in Λ_U **do**
2: $\quad \Gamma_{\Phi_i} = \bigcup_{\Upsilon \in \Phi_i} SetGen(\Upsilon)$
3: $\quad n_i = |\Gamma_{\Phi_i} \cap \Gamma_0|$
4: **end for**
5: Sort all Φ_is in Λ_U based on the n_is
6: Select the top w Φ_is in Λ_U to Λ
7: **return** Λ

Algorithm 6 ASP-BS

Input: Γ_0, Θ, w
Output: Φ
1: $\Lambda_U \leftarrow UpdSln(\emptyset, \Theta)$
2: $\Lambda \leftarrow BestSln(\Lambda_U, \Gamma_0, w)$
3: **while** no feasible solution found **do**
4: $\quad \Lambda_C \leftarrow \emptyset$
5: $\quad$ **for** each Φ_i in Λ $(1 \le i \le w)$ **do**
6: $\quad\quad \Theta \leftarrow \Theta \setminus \Phi_i$
7: $\quad\quad \Lambda_U^{(i)} \leftarrow UpdSln(\Phi_i, \Theta)$
8: $\quad\quad \Lambda_C \leftarrow \Lambda_C \cup \Lambda_U^{(i)}$
9: $\quad$ **end for**
10: $\quad \Lambda_C \leftarrow BestSln(\Lambda_C, \Gamma_0, w)$
11: $\quad$ **for** each Φ_i in Λ_C **do**
12: $\quad\quad \Gamma_{\Phi_i} \leftarrow \bigcup_{\Upsilon \in \Phi_i} SetGen(\Upsilon)$
13: $\quad\quad$ **if** $\Gamma_{\Phi_i} = \Gamma_0$ **then**
14: $\quad\quad\quad$ **return** Φ_i
15: $\quad\quad$ **end if**
16: $\quad$ **end for**
17: $\quad \Lambda \leftarrow \Lambda_C$
18: **end while**

The ASP-BS offers a reasonable trade-off between the depth and breadth of the searching, ensuring the computational resources can be effectively utilized. The most promising partial solutions at each stage are kept while maintaining a wide enough pool of candidates to diversify the exploration.

B. Special Case: Greedy Algorithm

When the beam width is 1, ASP-BS is reduced to a greedy algorithm (i.e., ASP-G). For each time, a sub-AS representation Φ_i is chosen from the collection Θ in such a way that the updated partial solution by Φ_i can maximize the number of overlapping qualified subsets between Γ_0 and the collection of subsets derived from the updated solution. It is noteworthy that the proposed ASP-G may not always provide a global optimal solution. However, in certain scenarios, it delivers an approximation of the best solution while maintaining low computational efficiency. Its intuitive nature makes it suitable for practical applications especially when the scale of the ASP problem is large and a rapid solution is required.

VI. Partition scheme based on Genetic Algorithm with Variable-Length Chromosomes

Reaching the global optimal solution may become challenging as the number of candidate sub-AS representations increases. In order to effectively find a near-optimal solution, we utilize the ASP technique based on genetic algorithm

with variable-length chromosomes (ASP-GAVC). As a meta-heuristic algorithm, the proposed method can solve complicated optimization problems on a large scale and has the capacity to escape local optima to find global optimal solutions. ASP-GAVC starts with a population of random potential solutions, which are evaluated for "fitness" based on a fitness function. The fittest solutions are then selected for "reproduction" using genetic operators like selection, crossover, and mutation. This process creates a new generation of solutions, and the cycle repeats until a satisfactory solution is found. Main components of this approach are provided as follows.

A. Chromosome with Variable-Length

Generic algorithm usually comprises a population of strings of character, called chromosome, in each generation. Every chromosome used in this paper is defined to be a feasible solution of ASP problem. It consists of several sub-AS representations such that they can cover all minimal qualified subsets presented in Γ_0.

In traditional genetic algorithms, the chromosome length is determined a priori and it cannot change in subsequent generations. However, fixed chromosome length is not suitable for ASP problem. If short chromosome is utilized, we may not achieve a feasible solution. On the other hand, if the chromosome length is excessive, it probably causes a high computational burden without much performance benefit. As a result of this, chromosome with variable-length is considered.

Formally, a chromosome is a collection of sub-AS representations that can cover Γ_0, as presented by $\{\Upsilon_1, \cdots, \Upsilon_m\}$. For each representation Υ_i, it consists of a sequence σ_i and a permutation β_i. For better understanding, an instance of chromosome is given in the following example.

Example 4. *Consider the ASP problem for* $(2, 4)$ *threshold. A chromosome might comprise* 2 *sub-AS representations:* $\{\Upsilon_1, \Upsilon_2\}$*, where* $\Upsilon_1 = \{\sigma_1, \beta_1\} = \{22, 2314\}$ *and* $\Upsilon_2 = \{\sigma_2, \beta_2\} = \{22, 1234\}$*. The corresponding qualified subsets generated by* Υ_1 *are* $\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4\}$*. And the subsets derived from* Υ_2 *are* $\{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}$*. The union of these subsets covers the* $(2, 4)$ *threshold.*

B. Population Initialization

Population initialization requires the random generation of a predefined number of chromosomes. Given the population size, each time a chromosome $\mathcal{CH}$ is produced at random by

$$\mathcal{CH} = ChrGen(\Gamma_0, \mathcal{A}, \mathcal{B}) \tag{9}$$

where $ChrGen$ is implemented by Algorithm 7, Γ_0 denotes the collection of minimal qualified subsets, $\mathcal{A}$ is the collection of various sequences provided by Algorithm 1, and $\mathcal{B}$ represents the collection of all permutations of $\{1, \cdots, n\}$. The generation of a chromosome is actually deriving a feasible random solution for the partition problem. A sub-AS representation is considered as a gene which is a basic unit of a chromosome. For each representation, it is formed by randomly selecting one sequence σ from $\mathcal{A}$ and one permutation β from $\mathcal{B}$. The generated representation is checked by function $Valid$. If it is not a feasible solution, another random representation is produced and examined until a valid one is constructed. The feasible representation is later concatenated to the current chromosome and Γ_0 is updated as $\Gamma_0 \setminus SetGen(\Upsilon)$. The algorithm terminates when $\Gamma_0 = \emptyset$. In this situation, Γ_0 is covered by the chromosome.

Algorithm 7 Random generation of a chromosome (ChrGen).

Input: Γ_0, $\mathcal{A}$, $\mathcal{B}$
Output: $\mathcal{CH}$
$\mathcal{CH} \leftarrow \emptyset$, $i \leftarrow 1$
while $\Gamma_0 \neq \emptyset$ **do**
  Randomly select a sequence σ from $\mathcal{A}$
  Randomly select a permutation β from $\mathcal{B}$
  $\Upsilon \leftarrow \{\sigma, \beta\}$
  while $Valid(\Gamma_0, \Upsilon)$ is not true **do**
    Randomly select a sequence σ from $\mathcal{A}$
    Randomly select a permutation β from $\mathcal{B}$
    $\Upsilon \leftarrow \{\sigma, \beta\}$
  end while
  $i \leftarrow i + 1$
  $\mathcal{CH} \leftarrow \mathcal{CH} || \Upsilon$
  $\Gamma_0 \leftarrow \Gamma_0 \setminus SetGen(\Upsilon)$
end while

Additionally, for function $Valid$, the sub-AS correlated to the representation is generated by $\Gamma_\Upsilon = SetGen(\Upsilon)$. If $\Gamma_0 \cap \Gamma_\Upsilon \neq \emptyset$, the representation is a feasible gene for a chromosome and a TRUE value is returned. Otherwise, it is not a valid one and a FALSE value is provided.

C. Fitness Function

A quantitative measurement, called fitness function, is adopted to evaluate the score of each individual within the environment. Higher fitness score of a chromosome indicates a better solution is achieved. In the proposed ASP-GAVC, the fitness value of a chromosome Φ is defined as

$$f(\Phi) = \binom{n}{k} - |\Phi| \tag{10}$$

where $|\Phi|$ represents the number of genes in a chromosome (i.e., the quantity of sub-AS representations in a solution). To achieve a high score, $|\Phi|$ should be as small as feasible. As a result of this, an optimal partition result is guaranteed.

D. Crossover

Crossover plays a significant role in producing innovative solutions. In our method, the two-point crossover is utilized. Let $\mathcal{CH}_1 = \{\Upsilon_1^{(1)}, \cdots, \Upsilon_L^{(1)}\}$ and $\mathcal{CH}_2 = \{\Upsilon_1^{(2)}, \cdots, \Upsilon_K^{(2)}\}$ be the two chosen parent chromosomes. Two crossover points x and y are randomly selected where $2 \leq x \leq L$ and $2 \leq y \leq K$. By utilizing the two-point crossover, two offspring chromosomes $\mathcal{CH}_1'$ and $\mathcal{CH}_2'$ are produced by exchanging the genes $\Upsilon_x^{(1)}, \cdots, \Upsilon_L^{(1)}$ and $\Upsilon_y^{(2)}, \cdots, \Upsilon_K^{(2)}$, as denoted by

$$\begin{cases} \mathcal{CH}_1' = \{\Upsilon_1^{(1)}, \cdots, \Upsilon_{x-1}^{(1)}, \Upsilon_y^{(2)}, \cdots, \Upsilon_K^{(2)}\}, \\ \mathcal{CH}_2' = \{\Upsilon_1^{(2)}, \cdots, \Upsilon_{y-1}^{(2)}, \Upsilon_x^{(1)}, \cdots, \Upsilon_L^{(1)}\}. \end{cases} \tag{11}$$

The two offspring might be unfeasible solutions for the ASP problem. In this scenario, a procedure $Repair$ is introduced to turn an unworkable result into feasible, as depicted in Algorithm 8. Suppose $\mathcal{CH}' = \{\Upsilon_1, \cdots, \Upsilon_{x-1}, \Upsilon_x, \cdots, \Upsilon_L\}$ is a newly formed chromosome. The first $(x-1)$ genes come

from one parent and the remaining $(L-x+1)$ genes are obtained from the other. $Repair$ examines whether the last $(L-x+1)$ genes $\Upsilon_x, \cdots, \Upsilon_L$ are feasible or not. If a gene is unworkable, a new one is randomly created and then examined again. This process repeats until a valid gene is constructed. The current gene is subsequently replaced with the new one. The $Repair$ procedure finally produces a valid chromosome (i.e., a feasible solution). Note that, additional genes would be created and added to the chromosome if it cannot cover Γ_0.

Algorithm 8 Repair

Input: Γ_0, $\mathcal{CH}'$, x, $\mathcal{A}$, $\mathcal{B}$
Output: $\mathcal{CH}^U$
Assign the first $x-1$ genes of $\mathcal{CH}'$ to $\mathcal{CH}^U$
$\Gamma_0 \leftarrow \Gamma_0 \setminus \bigcup_{i=1}^{x-1} SetGen(\Upsilon_i)$
$i = x$.
while $\Gamma_0 \neq \emptyset$ **do**
  if $x \leq i \leq L$ **then**
    Assign the sequence of the i-th gene to σ
    Assign the permutation of the i-th gene to β
  else
    Randomly select a sequence σ from $\mathcal{A}$
    Randomly select a permutation β from $\mathcal{B}$
  end if
  $\Upsilon \leftarrow \{\sigma, \beta\}$
  while $Valid(\Gamma_0, \Upsilon)$ is not true **do**
    Randomly select a sequence σ from $\mathcal{A}$
    Randomly select a permutation β from $\mathcal{B}$
    $\Upsilon \leftarrow \{\sigma, \beta\}$
  end while
  $\mathcal{CH}^U \leftarrow \mathcal{CH}^U || \Upsilon$
  $\Gamma_0 \leftarrow \Gamma_0 \setminus SetGen(\Upsilon)$
  $i \leftarrow i+1$.
end while

E. Mutation

Mutation serves as an operation for discovering new solutions. It provides a random diversity in population as well. The mutation operation used in our method is defined as follows. Let $\mathcal{CH} = \{\Upsilon_1, \cdots, \Upsilon_L\}$ be the chosen chromosome and let x be the mutation point where $1 \leq x \leq L$. A randomly generated gene Υ'_x is employed to replace Υ_x in $\mathcal{CH}$ to obtain the mutated chromosome, as represented by

$$\mathcal{CH}' = \{\Upsilon_1, \cdots, \Upsilon_{x-1}, \Upsilon'_x, \Upsilon_{x+1}, \cdots, \Upsilon_L\}. \quad (12)$$

Similar to the crossover operation, the mutated chromosome might be invalid. Therefore, the $Repair$ procedure (i.e., Algorithm 8) is also utilized to examine and update the genes $\Upsilon'_x, \Upsilon_{x+1}, \cdots, \Upsilon_L$ in $\mathcal{CH}'$.

F. Selection

It is necessary to select the best solutions of current generation to direct ASP-GAVC to reach the global optimum. Selection is the process of choosing individuals with high fitness scores from current population, so that good individuals can potentially serve as parents to derive offspring. In this paper, the roulette wheel selection is employed.

First, the score of every chromosome $f(\Phi_i)$, $1 \leq i \leq N$, in current generation is calculated, where N is the total number of chromosomes (i.e., population size). Then, the selection probability of each chromosome $p(\Phi_i)$ is given as

$$p(\Phi_i) = \frac{f(\Phi_i)}{\sum_{j=1}^{N} f(\Phi_j)}. \quad (13)$$

In the next step, the cumulative probability of each chromosome $p_c(\Phi_i)$ is computed based on the selection probabilities of all previous chromosomes, as given by

$$p_c(\Phi_i) = \sum_{j=1}^{i} p(\Phi_j). \quad (14)$$

Finally, a random number r from the interval $[0, 1]$ is generated and compared with $p_c(\Phi_i)$ to determine the selection of a chromosome. If $p_c(\Phi_{t-1}) < r < p_c(\Phi_t)$, the t-th chromosome is chosen. This selection procedure repeats for N_S times to choose N_S offspring for the next generation.

G. Main Procedure

Algorithm 9 depicts the main steps of ASP-GAVC, where N is the population size, N_S is the number of offspring produced by the selection operation, $\tilde{p}_c$ and $\tilde{p}_m$ are the crossover and mutation rates, respectively, and T is the maximum iterations. According to the experiments, the following parameters can be regarded to produce satisfactory solutions: $N = 2\binom{n}{k}$, $N_S = 2\binom{n}{k}$, $\tilde{p}_c = 0.2$, $\tilde{p}_m = 0.1$, and $T = 50\binom{n}{k}$.

Algorithm 9 ASP-GAVC

Input: Γ_0, N, N_S, $\tilde{p}_c$, $\tilde{p}_m$, T
Output: The best chromosome
$t \leftarrow 0$
Initialize the population with size N
while $t < T$ **do**
  Calculate the fitness values of all chromosomes
  Record the chromosome with the highest fitness value
  Perform select operation with N_S
  Perform crossover operation with $\tilde{p}_c$
  Perform mutation operation with $\tilde{p}_m$
  $t \leftarrow t+1$
end while

In Step 2, a population of N chromosomes is randomly initialized using the method described in Section VI-B. The proposed ASP-GAVC iteratively searches the solution space to obtain an optimal solution via Steps 3-10. For Steps 4-5, the fitness value of each chromosome in the current generation is calculated and the best one is identified. In Step 6, the roulette wheel selection given in Section VI-F is employed to choose the best chromosomes as parents to derive offspring. As stated in Step 7, the crossover operation is performed among selected pairs of chromosomes from the parent population to produce offspring, at a predetermined crossover rate p_c. Then, the mutation operation is utilized to create new offspring with a mutation rate p_m in Step 8. We can achieve a new generation of chromosomes via the selection, crossover, and mutation operations. The proposed algorithm terminates when the maximum number of iteration reaches T. Finally, the best chromosome (i.e., an optimal solution) is provided.

VII. Experimental Result and Discussion

In this section, visual examples by the proposed sharing technique are provided. Optimal solutions of the ASP problem by using the proposed partition algorithms are demonstrated, as well as analysis on computational efficiency, secret recovery and security. In depth discussion and comparison are illustrated at the end of this section. In addition, all experiments are implemented by Python 3.11 under the following environment: Apple M3 Pro processor, 18 GB RAM and macOS Sonoma.

A. Visual Experiments of ASP Guided Image Secret Sharing

Experiments of the $(2, 4)$ and $(3, 5)$ schemes are conducted. For the $(2, 4)$ threshold, the collection of minimal qualified subsets is Γ_0=$\{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$. By applying Algorithms 1 and 2, we obtain a collection of 7 candidate sub-AS representations $\Theta = \{\Upsilon_1, \cdots, \Upsilon_7\}$:

$$\begin{cases} \Upsilon_1 = \{\{1,3\},\{1,2,3,4\}\}, \Upsilon_2 = \{\{1,3\},\{2,3,4,1\}\}, \\ \Upsilon_3 = \{\{1,3\},\{3,4,1,2\}\}, \Upsilon_4 = \{\{1,3\},\{4,3,2,1\}\}, \\ \Upsilon_5 = \{\{2,2\},\{1,2,3,4\}\}, \Upsilon_6 = \{\{2,2\},\{4,2,3,1\}\}, \\ \Upsilon_7 = \{\{2,2\},\{1,4,2,3\}\}. \end{cases} \tag{15}$$

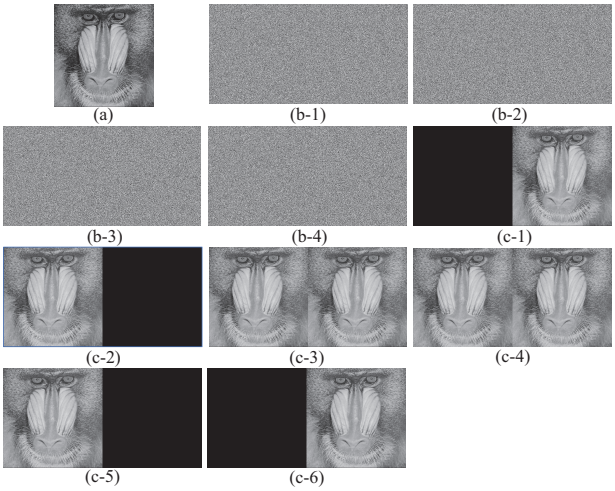

Fig. 4. A $(2, 4)$ experiment by our sharing approach based on the partition result by ASP-DP/ASP-BS. (a) Gray-level secret image from the USC-SIPI image database, (b) 4 final shadows, (c) decoded results by XOR-ing any 2 of 4 final shadows.

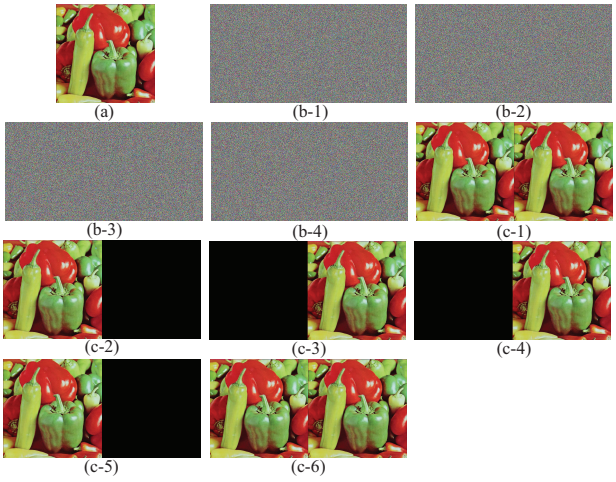

Fig. 5. A $(2, 4)$ experiment by our sharing approach based on the partition result by ASP-GAVC. (a) Color secret image from the USC-SIPI image database, (b) 4 final shadows, (c) decoded results by XOR-ing any 2 of 4 final shadows.

When using ASP-DP and ASP-BS (with $w = 2$), Γ_0 is divided into 2 sub-AS representations: Υ_5 and Υ_6. When applying ASP-GAVC, the best chromosome is found as $\{221432224213\}$ which comprises 2 representations: $\Upsilon_7 = \{\{2, 2\}, \{1, 4, 2, 3\}\}$, $\Upsilon_6 = \{\{2, 2\}, \{4, 2, 3, 1\}\}$. With the partition result, we can use two different $(2, 2)$ PVCSs [15] to realize the $(2, 4)$ threshold. Fig. 4 shows the $(2, 4)$ experiment based on the partition result provided by ASP-DP/ASP-BS. The gray-level secret image with 512×512 pixels is given in Fig. 4 (a) and the 4 final shadows are illustrated in Figs. 4 (b). Any 2 of the participants can reveal the secret, as given in Figs. 4 (c). Moreover, the $(2, 4)$ experiment for color secret image is illustrated in Fig. 5, where the partition result is provided by ASP-GAVC. When dealing with color images, the RGB decomposition is adopted to break a color image into 3 channels. Some issues for color images, such as channel correlations or quantization errors, might be considered as future work for enhancing the performance of the proposed sharing method.

For the $(3, 5)$ threshold, 25 candidate sub-AS representations are achieved as

$$\begin{cases} \Upsilon_1 = \{\{1,1,3\},\{1,4,2,3,5\}\}, \Upsilon_2 = \{\{1,1,3\},\{1,2,3,4,5\}\}, \\ \Upsilon_3 = \{\{1,1,3\},\{1,3,2,4,5\}\}, \Upsilon_4 = \{\{1,1,3\},\{1,5,2,3,4\}\}, \\ \Upsilon_5 = \{\{1,1,3\},\{2,3,1,4,5\}\}, \Upsilon_6 = \{\{1,1,3\},\{2,5,1,3,4\}\}, \\ \Upsilon_7 = \{\{1,1,3\},\{2,4,1,3,5\}\}, \Upsilon_8 = \{\{1,1,3\},\{3,4,1,2,5\}\}, \\ \Upsilon_9 = \{\{1,1,3\},\{3,5,1,2,4\}\}, \Upsilon_{10} = \{\{1,1,3\},\{4,5,1,2,3\}\}, \\ \Upsilon_{11} = \{\{1,2,2\},\{1,4,2,3,5\}\}, \Upsilon_{12} = \{\{1,2,2\},\{1,2,3,4,5\}\}, \\ \Upsilon_{13} = \{\{1,2,2\},\{1,2,5,3,4\}\}, \Upsilon_{14} = \{\{1,2,2\},\{2,3,5,1,4\}\}, \\ \Upsilon_{15} = \{\{1,2,2\},\{2,1,5,3,4\}\}, \Upsilon_{16} = \{\{1,2,2\},\{3,1,5,2,4\}\}, \\ \Upsilon_{17} = \{\{1,2,2\},\{2,1,3,4,5\}\}, \Upsilon_{18} = \{\{1,2,2\},\{4,2,3,1,5\}\}, \\ \Upsilon_{19} = \{\{1,2,2\},\{3,2,5,1,4\}\}, \Upsilon_{20} = \{\{1,2,2\},\{3,1,2,4,5\}\}, \\ \Upsilon_{21} = \{\{1,2,2\},\{4,1,2,3,5\}\}, \Upsilon_{22} = \{\{1,2,2\},\{4,2,5,1,3\}\}, \\ \Upsilon_{23} = \{\{1,2,2\},\{5,2,3,1,4\}\}, \Upsilon_{24} = \{\{1,2,2\},\{5,1,2,3,4\}\}, \\ \Upsilon_{25} = \{\{1,2,2\},\{5,2,4,1,3\}\}. \end{cases} \tag{16}$$

Both ASP-DP and ASP-BS offer the same division result. The minimal qualified subsets of $(3, 5)$ threshold are partitioned into 3 parts: $\Upsilon_1, \Upsilon_5, \Upsilon_{23}$. When using ASP-GAVC, we achieve the optimal solution which consists of 3 sub-AS representations $\{\{2, 2, 1\}, \{4, 3, 1, 5, 2\}\}$, $\{\{2, 2, 1\}, \{4, 5, 2, 3, 1\}\}$, $\{\{1, 2, 2\}, \{4, 3, 1, 2, 5\}\}$ (equivalent to $\Upsilon_{15}, \Upsilon_{12}, \Upsilon_{22}$, respectively). The experiment of $(3, 5)$ threshold by the proposed sharing scheme is demonstrated in Fig. 6, where the partition result is provided by ASP-DP/ASP-BS.

For various real-world applications, the threshold parameters might be small (e.g., $k \leq n \leq 6$). Hence, we only show the $(2, 4)$ and $(3, 5)$ experiments. The proposed sharing method is also applicable for large-scale thresholds. The implementation of large-scale thresholds is identical to that of small-scale thresholds. For instance, consider the $(2, 18)$ scheme which can be partitioned into 5 sub-ASs by ASP-GAVC, as indicated in Fig. 7 (a). One can utilize 5 different $(2, 2)$ PVCSs to realize the $(2, 18)$ scheme. The final shadow for each participant would be 5 times of the original image.

For real-world deployment, a hardware architecture of the proposed sharing technique to accelerate the generation of shadows and reconstruction of secret can be developed in the future. As each pixel is processed by the same method, the architecture can employ Field Programmable Gate Array (FPGA) to parallelizes the secret sharing process and secret recovery procedure. Computing efficiency is improved when parallel processing is adopted. In addition, the partition algorithms become computationally expensive when the threshold

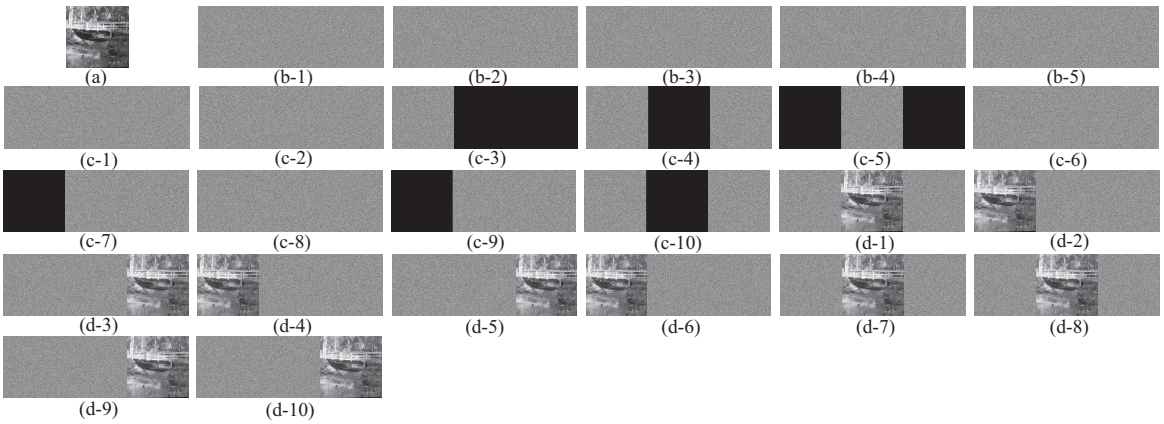

Fig. 6. A $(3, 5)$ experiment by our sharing approach based on the partition result by ASP-DP/ASP-BS. (a) Grayscale secret image from the USC-SIPI image database, (b) 5 final shadows, (c) decoded results by XOR-ing any 2 of 5 final shadows, (d) decoded results by XOR-ing any 3 of 5 final shadows.

becomes large. For real-world application, the partition solutions for various thresholds should be computed in advance and stored for future usage.

TABLE II
PARTITION RESULTS BY ASP-DP.

n	k				
	2	3	4	5	6
3	2	-	-	-	-
4	2	2	-	-	-
5	3	3	3	-	-
6	3	3	5	3	-
7	3	4	7	6	4

B. Optimal Solutions by The Partition Algorithms

Division results of some commonly-used thresholds by using the three proposed partition algorithms are provided in this sub-section, as well as the discussion and comparison.

TABLE III
PARTITION RESULTS (WITH BEAM WIDTH) BY ASP-BS.

n	k					
	2	3	4	5	6	7
3	2(2)	-	-	-	-	-
4	2(2)	2(2)	-	-	-	-
5	3(2)	3(2)	3(2)	-	-	-
6	3(2)	3(2)	5(2)	3(2)	-	-
7	3(2)	4(2)	7(2)	6(2)	4(2)	-
8	3(2)	4(2)	7(3)	9(2)	8(2)	4(2)
9	4(2)	4(2)	9(2)	12(2)	14(3)	10(3)
10	4(2)	5(3)	10(4)	12(3)	19(3)	18(4)

An optimal solution is guaranteed by ASP-DP since it explores the whole searching space of the problem. However, the searching space becomes extremely large as k and n increase. In this situation, ASP-DP is not capable of investigating the searching space effectively. Consequently, ASP-DP is not possible to exploit the best solutions of sub-problems to create an overall optimal one. Table II demonstrates the best partition results where k and n are relatively small.

TABLE IV
PARTITION RESULTS BY ASP-G (GREEDY ALGORITHM).

n	k					
	2	3	4	5	6	7
3	2	-	-	-	-	-
4	2	2	-	-	-	-
5	3	3	3	-	-	-
6	3	4	5	3	-	-
7	3	4	7	6	4	-
8	3	4	7	10	9	4
9	4	4	10	12	14	10
10	4	7	11	13	19	19

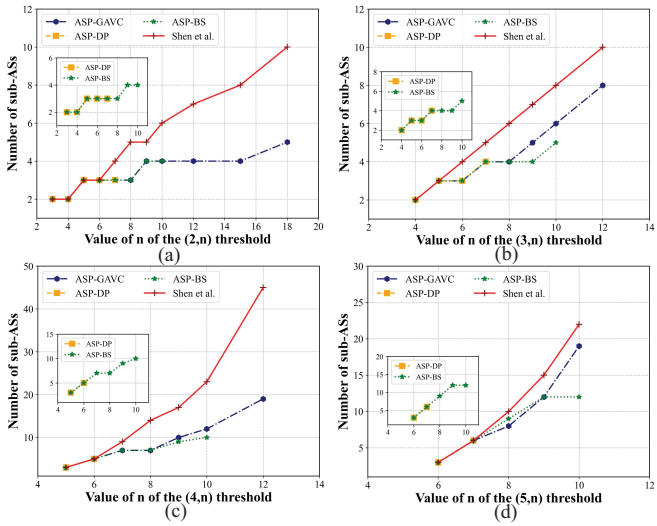

Fig. 7. Comparison of partition results between our methods and Shen *et al.*'s approach [44]. (a) $(2, n)$, (b) $(3, n)$, (c) $(4, n)$, (d) $(5, n)$.

When compared with ASP-DP, ASP-BS provides a more focused exploration of the searching space. By adjusting the beam width to retain a limited number of promising partial solutions, it reduces the computational complexity dramatically while still seeking good results in many cases. The partition results by ASP-BS are illustrated in Table III where the beam widths are given in parentheses. The optimal

TABLE V
PARTITION RESULTS BY ASP-GAVC.

n	k						
	2	3	4	5	6	7	8
3	2	-	-	-	-	-	-
4	2	2	-	-	-	-	-
5	3	3	3	-	-	-	-
6	3	3	5	3	-	-	-
7	3	4	7	6	4	-	-
8	3	4	7	8	8	4	-
9	4	5	10	12	15	10	5
10	4	6	12	19	25	20	13

TABLE VI
PARTITION RESULTS BY ASP-GAVC FOR LARGER VALUES OF k AND n.

n	k			
	2	3	4	5
12	4	8	19	36
15	4	10	27	61
18	5	12	36	85
20	5	13	41	112

solutions provided by ASP-DP are the same as those by ASP-BS. Moreover, when the beam width is set to be 1, we have the greedy method for the ASP problem. The corresponding results are illustrated in Table IV.

The ASP-DP and ASP-BS cannot efficiently tackle the problem when the design space is sufficiently large. However, the proposed ASP-GAVC can address this problem. Table V provides the best partition results by ASP-GAVC. More solutions are illustrated in Table VI.

The three proposed partition algorithms are compared with a similar technique in [44]. Shen *et al.*'s partition method [44] is likely to be a heuristic approach for the ASP problem. However, their approach only produces feasible solutions. The optimality of a solution is not discussed. In other words, optimal partition results are not ensured by their technique. In addition, integer linear programming (ILP) might be regarded as the baseline method for comparison. However, ILP cannot be applied directly since additional functions for ASP problem are required. Thus, only Shen *et al.*'s solution [44] is included as baseline approach for comparison. Fig. 7 illustrates the comparison of partition results between our methods and Shen *et al.*'s approach [44]. For small values of n (e.g., $(2, n)$ with $3 \leq n \leq 6$, $(3, n)$ with $4 \leq n \leq 5$, $(4, n)$ with $5 \leq n \leq 6$, and $(5, n)$ with $6 \leq n \leq 7$), our methods, as well as Shen *et al.*'s technique, produce the same partition results. Our ASP-GAVC yields better solutions that are significantly smaller than the outcomes of Shen *et al.* for the cases with a larger n. Consider the $(2, 15)$, $(2, 18)$, and $(4, 12)$ thresholds, the numbers of partition sub-ASs by Shen *et al.*'s method are 8, 10, and 45, whereas, the quantities by our ASP-GAVC are 4, 5, and 19. Further, the partition results also indicate that our ASP-DP and ASP-BS outperform Shen *et al.*'s approach. In conclusion, the proposed approaches perform noticeably better than Shen *et al.*'s partition strategy.

C. Computational Efficiency

Candidate representation generation (i.e., Algorithms 1 and 2) offers candidate sub-AS representations. The computation of deriving candidate representations becomes expensive as n increases. Scalability analysis of the candidate representation generation is given. The amount of time required to produce candidate representations is illustrated in Table VII. It is evident that it would be computationally efficient when $n \leq 8$.

TABLE VII
TIME (BY SECONDS) OF GENERATING CANDIDATE REPRESENTATIONS.

n	k					
	2	3	4	5	6	7
3	0.00005	-	-	-	-	-
4	0.00007	0.00005	-	-	-	-
5	0.0003	0.0004	0.0003	-	-	-
6	0.0026	0.0037	0.0031	0.0019	-	-
7	0.0165	0.0278	0.0259	0.0222	0.0140	-
8	0.1582	0.2751	0.3571	0.2692	0.2269	0.1371
9	1.5058	3.7655	4.2253	4.4609	3.2949	2.6403
10	23.558	47.703	69.064	67.814	59.199	42.578

TABLE VIII
TIME (BY SECONDS) COMPARISON.

(k, n)	ASP-DP	ASP-BS	ASP-GAVC	Shen *et al.*
$(2, 3)$	**0.000008**	0.000020	0.019760	0.000043
$(2, 4)$	**0.000014**	0.000035	0.038003	0.000164
$(2, 5)$	0.000102	**0.000087**	0.073226	0.000622
$(2, 6)$	**0.000143**	0.000171	0.509792	0.001380
$(2, 7)$	0.014763	**0.000465**	0.762519	0.004198
$(3, 4)$	**0.000014**	0.000035	0.042519	0.000084
$(3, 5)$	0.000181	**0.000148**	0.250851	0.000178
$(3, 6)$	0.080108	**0.000663**	0.545436	0.000846
$(3, 7)$	10.22213	**0.003688**	2.823513	0.006623
$(4, 5)$	**0.000024**	0.000052	0.168099	0.000074
$(4, 6)$	0.107170	0.000679	0.589574	**0.000371**

Comparison of time for achieving partition results between our techniques and Shen *et al.*'s approach [44] is demonstrated in Table VIII. The best performance is highlighted. According to the results, ASP-DP and ASP-BS work well for small thresholds with a limited number of candidate representations, whereas ASP-GAVC and Shen *et al.*'s scheme [44] may require more processing time. On the other hand, ASP-DP becomes computationally intensive when the searching space is large. For example, it requires more than 10 seconds to partition the $(3, 7)$ threshold. In contrast, partition problems with larger k and n may benefit more from the use of ASP-BS and ASP-GAVC. For practical applications, the sets of candidates and partition solutions for various thresholds can be calculated in advance and stored for future usage.

D. Secret Recovery and Security

The sharing method provides lossless decoding and perfect security. The property of distortion-free recovery can be verified theoretically. According to [47], a (k, k) PVCS is actually an XVCS, suggesting that Boolean XOR operation can be applied to decode the secret without distortion. The (k, k)

PVCS serves as a central building block in the sharing method to handle the bit planes of a secret image. As every bit plane can be losslessly decoded, the gray-level secret image can be reconstructed in a distortion-free manner.

The property of perfect security is examined experimentally and theoretically. For experimental verification, the following 4 objective measurements [48] [49] are usually employed to examine the security of shadows: Peak Signal-to-Noise Ratio (PSNR) of the shadow, Structural Similarity (SSIM) of the shadow, Correlation Coefficients (CC) of adjacent shadow pixels in horizontal, vertical, and diagonal directions, and Entropy of the shadow. The way to calculate the 4 metrics can refer to [48] [49]. A sharing scheme is supposed to be with the highest security level [48] [49], when the following criteria are met: PSNR ≈ 8.6845, SSIM ≈ 0, CC ≈ 0, and Entropy ≈ 7.9973. Herein, the $(2, 4)$ scheme given in Fig. 4 is verified experimentally. Every final shadow consists of 2 gray-level shadows. The 4 objective measurements are applied to each gray-level shadow. Statistical results are demonstrated in Table IX where $SH_{i,j}$, $1 \leq i \leq 4, 1 \leq j \leq 2$, denotes the j-th grayscale shadow of the i-th final shadow. According to Table IX, it is evident that the PSNRs, SSIMs, CCs and Entropies are approximately the same as the values suggested in [48] [49]. The generated shadows are unable to reveal any information about the secret. Highest security level is achieved by the proposed sharing technique.

TABLE IX
PSNR, SSIM, CC, AND ENTROPY OF THE SHADOWS GIVEN IN FIG. 4.

Shadow	PSNR	SSIM	CC			Entropy
			Horizontal	Vertical	Diagonal	
$SH_{1,1}$	8.5233	0.0024	-0.0011	0.0009	0.0014	7.9713
$SH_{1,2}$	8.6187	0.0018	0.0021	-0.0008	-0.0013	7.9524
$SH_{2,1}$	8.4759	0.0023	-0.0009	0.0016	0.0024	7.9665
$SH_{2,2}$	8.5927	0.0019	-0.0015	0.0008	0.0019	7.9742
$SH_{3,1}$	8.6745	0.0032	0.0016	-0.0018	0.0022	7.9522
$SH_{3,2}$	8.5389	0.0038	0.0027	0.0017	0.0016	7.9703
$SH_{4,1}$	8.6461	0.0027	-0.0034	0.0022	0.0009	7.9852
$SH_{4,2}$	8.5253	0.0032	0.0017	0.0000	-0.0025	7.9575

The security of the sharing method is also verified theoretically. Generally, there exist two security threats that might compromise the security. The one is the concern within one sub-AS. As equivalent participants are used, identical bit plane shadows belonging to one (k, k) PVCS of a sub-AS are distributed to equivalent members. In this scenario, adversaries would collect identical bit plane shadows for recovering the secret without satisfying the threshold requirement. The other is the threat among different sub-ASs. Suppose the (k, n) threshold is divided into d sub-ASs. A secret bit plane is encoded by d different (k, k) PVCSs. Adversaries would obtain bit plane shadows across different sub-ASs to reconstruct the secret without meeting the threshold condition. Herein, Theorem 1 is presented to ensure that the foregoing threats can be prevented.

Theorem 1. *The proposed (k, n) sharing scheme is secure such that any $(k-1)$ or fewer participants cannot achieve any clue about the secret.*

Proof. Let $(k-1)$ participants are involved in recovering the x-th secret bit plane $S^{(x)}$, $1 \leq x \leq 8$. We prove that any $(k-1)$ participants cannot achieve any information about $S^{(x)}$.

The case of a single sub-AS is considered. For each sub-AS, the secret bit plane $S^{(x)}$ is encoded via a (k, k) PVCS to create k bit plane shadows. The participants in the same equivalent relationship would receive the same bit plane shadow. The $(k-1)$ participants would be involved in $(k-1)$ different equivalent relationships at most. At this time, the maximum number of shadows collected by the $(k-1)$ participants is $(k-1)$. For a (k, k) PVCS, these $(k-1)$ bit plane shadows cannot disclose any clue about the secret bit plane $S^{(x)}$. The first security threat is prevented.

The scenario of a union of sub-ASs is regarded. Suppose that the (k, n) threshold is separated into d sub-ASs. For each sub-AS, a (k, k) PVCS is adopted. Thus, the secret bit plane $S^{(x)}$ is encrypted by d different (k, k) PVCSs. It is worthwhile to point out that the shadows belonging to one (k, k) PVCS are independent of those belonging to another (k, k) scheme. The analysis is given below.

Recall an alternative shadow construction of a (k, k) PVCS for encoding a secret bit s. In the beginning, $(k-1)$ random bits $b_1, \cdots, b_{k-1}$ are generated as $(k-1)$ shared bits. The k-th shared bit b_k is created by $b_k = s \oplus b_1 \oplus \cdots \oplus b_{k-1}$ where $\oplus$ denotes the Boolean XOR operation. For any two (k, k) PVCSs that encodes the same secret s, we have one scheme with $b_1, \cdots, b_k$ and the other method with $b'_1, \cdots, b'_k$. $b_1, \cdots, b_{k-1}$ are independent of $b'_1, \cdots, b'_k$ since these $(k-1)$ bits are randomly generated. b_k is also independent of $b'_1, \cdots, b'_{k-1}$ because $b'_1, \cdots, b'_{k-1}$ are random bits as well. We further consider b_k and b'_k. The secret s is involved in producing both b_k and b'_k. However, since $b_1, \cdots, b_{k-1}$ and $b'_1, \cdots, b'_{k-1}$ are randomly produced, $b_k (= s \oplus b_1 \oplus \cdots \oplus b_{k-1})$ is independent of $b'_k (= s \oplus b'_1 \oplus \cdots \oplus b'_{k-1})$ as a result. In conclusion, as randomness is utilized in shadow construction, for any two different (k, k) PVCSs, the k shared bits $b_1, \cdots, b_k$ from one approach are independent of the k ones $b'_1, \cdots, b'_k$ belonging to the other. It implies that we cannot learn any information about the shared bits of one (k, k) approach from those belonging to the other scheme. When $(k-1)$ participants are involved in d sub-ASs, they achieve at most $(k-1)$ bit plane shadows of a (k, k) PVCS for each sub-AS since every (k, k) PVCS is independent of each other. Information leakage among different sub-ASs is prevented. The second security threat is avoided. According to the foregoing analysis, the proposed sharing technique is proved to be secure.

□

Additionally, our sharing method is immune against brute force attack, single-point attack, and noise interference. In the proposed sharing technique, each bit is generated randomly. For an 8-bit grayscale secret image with $W \times H$ pixels, we have 2^{8WH} combinations for each shadow. This vast sample size makes share prediction infeasible. As at least k shadows are required to decode the secret, guessing all shares becomes exponentially difficult. The brute force attack can be prevented. Moreover, our (k, n) sharing technique breaks the secret into n shadows. Even when one shadow is destroyed, additional

TABLE X
FEATURE COMPARISON AMONG VARIOUS SCHEMES.

Scheme	Feature					
	Key Technique	Secret Format	Decoding Method	Decoding Complexity	Security	Recovery Type
Ref. [14]	Base Matrices	B	OR(Boolean)	$\mathcal{O}(1)$	Perfectly Secure	Lossy
Ref. [16]	ILP	B	XOR(Boolean)	$\mathcal{O}(1)$	Perfectly Secure	Lossy
Ref. [27]	EAS, SA	B	OR, XOR(Boolean)	$\mathcal{O}(1)$	Perfectly Secure	Lossy
Ref. [30]	SS	G/C	LI (Modular Arithmetic)	$\mathcal{O}(k \log^2 k)$	Residual-image Problem	Lossless
Ref. [31]	SS, Steg., Auth.	G/C	LI (Modular Arithmetic)	$\mathcal{O}(k \log^2 k)$	Residual-image Problem	Lossless
Ref. [34]	T-ILSS, Auth.	G/C	LI (Modular Arithmetic)	$\mathcal{O}(k \log^2 k)$	Perfectly Secure	Lossless
Ref. [33]	SS, Auth.	G/C	LI (Modular Arithmetic)	$\mathcal{O}(k \log^2 k)$	Perfectly Secure	Lossless
Ref. [37]	SS, BC, OC	G/C	LI (Modular Arithmetic)	$\mathcal{O}(k \log^2 k)$	Residual-image Problem	Lossless
Our	ASP, PVCS	G/C	XOR(Boolean)	$\mathcal{O}(d)$	Perfectly Secure	Lossless

ILP: Integer Linear Programming, EAS: Evolving Access Structure, SA: Simulated Annealing method, SS: Shamir's Secret Sharing, BC: Blockchain
Steg.: Steganography, Auth.: Authentication, T-ILSS: Transform domain-based Invertible and Lossless Secret Sharing, OC: Outsource Computation
ASP: Access Structure Partition, PVCS: Probabilistic VCS, LI: Lagrange Interpolation, B: Binary, G/C: Grayscale/Color

shadow from the remaining $(n-k)$ ones can be employed to decode the secret. Single-point attack is precluded. According to the analysis given in [50] [51], VCS has noise immunity for secret recovery. Even though black or white pixels in shadows suffer from interference by noise, the color may still retain the corresponding darkness with high probability. The noise interference does not severely affect secret recovery in VCS. As the (k, k) PVCS is considered as a basic building block in our sharing technique, the final shadows of our method are robust to noise interference as well.

E. Comparison and Discussion

Features of the proposed sharing scheme, such as key technique, secret image format, decoding method and complexity, security, shadow size, and secret recovery type, are discussed. Table X summarizes the comparison of features across various sharing approaches.

1) ***Key Technique for Constituting the Sharing Method****:* Our sharing method utilizes the ASP algorithms to divide the (k, n) threshold into sub-ASs and employs PVCS to realize the desired threshold based on the partition result. For conventional VCS [14] and XVCS [16], the base matrices and integer linear programming (ILP) are adopted to build the schemes, respectively. VCS designed for evolving access structure (EAS) [27] is implemented via simulated annealing (SA) method. For PSIS methods [30], [31], [33], [34], [37], Shamir's SS is used. Other schemes employ additional techniques, such as steganography [31], authentication [31], [33], transform domain-based invertible and lossless SS (T-ILSS) [34], Blockchain (BC) [37], and outsource computation (OC) [37], to constitute the sharing methods.

2) ***Secret Image Format****:* As a variety of grayscale/color images are utilized in various applications, the secret image format is preferred to be grayscale/color. However, VCS [14], [16], [18], [27], [52] is limited to binary images. Grayscale/color images are beyond its capabilities. Consequently, VCS would become far less applicable. However, by employing the division results offered by the partition algorithms, our sharing approach can deal with grayscale/color images. The application scenario is enriched.

3) ***Decoding Method and Complexity****:* Various approaches [32]–[34], [37] utilize PSIS [30], [31] as a central building component for encoding a secret. When performing decryption, Lagrange interpolation is involved. The computation complexity is therefore $\mathcal{O}(k \log^2 k)$. Furthermore, all of the calculations in PSIS rely on modular arithmetic operation, which is more computational demanding than Boolean operation. For the proposed sharing technique, bit-wise XOR operation can be used to decode every secret bit plane from the bit plane shadows. Therefore, a pixel-wise XOR operation can be employed to directly recover the gray-level secret image from the grayscale shadows. In this situation, bit plane decomposition and composition are no longer required in secret recovery. In addition, since each gray-level secret pixel is encoded into d groups of grayscale shadows where d is the number of sub-ASs, d pixel-wise XOR operations are performed for secret reconstruction. As a result of this, the decoding complexity is $\mathcal{O}(d)$. The proposed sharing scheme also offers the simplicity of secret recovery.

4) ***Security****:* When permutation is not performed, some PSIS approaches [30], [31], [37], which use k secret pixels as the coefficients of the $(k-1)$ degree polynomial, suffer from the residual-image problem. However, other techniques [32]–[34], merely incorporating a single pixel into a coefficient of a polynomial, are entirely secure. On the other hand, VCS and our sharing technique are perfectly secure techniques for sharing a secret such that completely noise-like shadows are generated.

5) ***Secret Recovery Type****:* VCS superimposes the shadows to decipher the secret in a lossy manner. For many applications, lossy decoding is unsuitable. Nonetheless, our sharing technique can perfectly rebuild the secret using XOR operation. Secret recovery without distortion is provided. Contrast is usually employed to evaluate the visual performance of VCS. Comparison of contrast among related VCS methods is demonstrated in Table XI. The methods in [16], [52] produce various contrasts. Only the best contrast is adopted for comparison. On the other hand, our sharing method utilizes PVCS to process each bit plane of a secret image. The contrast of recovered bit plane is calculated. According to the results, optimal contrast is ensured by our method.

TABLE XI
COMPARISON OF CONTRAST.

Threshold	Ref. [52]	Ref. [16]	Our
$(2,3)$	0.6667	0.6667	1
$(2,4)$	0.5000	0.6667	1
$(2,5)$	0.4000	0.6000	1
$(2,6)$	0.3556	0.6000	1
$(2,7)$	0.3265	0.5714	1
$(3,4)$	0.5333	0.6667	1
$(3,5)$	0.4444	0.5000	1
$(3,6)$	0.3429	0.4000	1
$(3,7)$	0.3000	0.4000	1
$(4,5)$	0.5333	0.5333	1
$(4,6)$	0.3333	0.4444	1

*6) **Shadow Size**:* Shadow size determines the storage overhead. Smaller shadow size indicates improved storage overhead. In VCS, shadow size is represented by pixel expansion. The generated shadow in traditional VCS [14] is usually $m \geq 2$ times of the secret where m is referred to as pixel expansion. In the proposed sharing method, we adopt PVCS to generate the bit plane shadows. Both shadow and secret are with the same size. Thus, the final shadow size is d times of the secret where d is the number of sub-AS representations that can cover all minimal qualified subsets. Since optimal partition algorithms are developed in this paper, the proposed sharing method yields optimal final shadow size when compared with the sharing technique using Shen *et al.*'s partition result [44]. Furthermore, as k pixels are usually embedded into the coefficients of a $(k-1)$-degree polynomial in PSIS [30], [31], [37], the size of a shadow is usually $1/k$ times of the secret. It is worth noticed that, for some PSIS techniques [32]–[34], the size of a shadow is the same as the secret when only one pixel is concealed into a coefficient of a polynomial. When compared with PSIS approaches [30]–[34], [37], the storage overhead by the proposed sharing technique is inferior.

VIII. CONCLUSION

The primary contributions of this paper are the systematical investigation of optimal partition algorithms for ASP problem and the proposal of ASP guided image secret sharing method. Based on the result from the partition algorithms, we are able to constitute a sharing approach that can encode grayscale/color images while providing benefits of easy decoding, perfect security, and lossless secret reconstruction. Extensive experiments and comparisons were illustrated, demonstrating the effectiveness and advantages of the proposed sharing method and partition techniques.

Future research could focus on the areas of study listed below. The proposed partition algorithms are mainly based on traditional method and meta-heuristic algorithm. Recently, AI-based optimization techniques (e.g., reinforcement learning method) have emerged as potential approaches for solving combinatorial problems. It becomes essential to integrate AI-based optimization into the proposed sharing method for offering optimal partition results. We also consider the hardware-level secure implementation for the sharing method. In this scenario, hardware-level vulnerabilities should be identified, investigated, and analyzed theoretically. Defense solutions, that can ensuring confidentiality, integrity, and availability, should be provided to prevent the security issues. Moreover, the proposed sharing method can be extended to real-time IoHT environments. The characteristics of medical records should be further studied. Meantime, as the proposed partition algorithms become computationally expensive when the threshold becomes large, efficient partition technique is required to reduce the running time so that the sharing method can be suitable for real-time environments.

REFERENCES

[1] H. Ali-Pacha, A. Hadj Brahim, A. Ali Pacha, and A. Gutub, "Data reorganization for image encryption: presentation in the data of hare," *Journal of Cyber Security Technology*, pp. 1–16, 2024.
[2] B. O. Al-Roithy and A. Gutub, "Remodeling randomness prioritization to boost-up security of rgb image encryption," *Multimedia Tools and Applications*, vol. 80, no. 18, pp. 28 521–28 581, 2021.
[3] A. Gutub and B. Al-Roithy, "Varying prng to improve image cryptography implementation," *Journal of Engineering Research*, vol. 9, no. 3A, 2021.
[4] B. Al-Roithy and A. Gutub, "Trustworthy image security via involving binary and chaotic gravitational searching within prng selections," *International Journal of Computer Science and Network Security*, vol. 20, no. 12, pp. 167–176, 2020.
[5] F. S. Hassan and A. Gutub, "Novel embedding secrecy within images utilizing an improved interpolation-based reversible data hiding scheme," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 2017–2030, 2022.
[6] A. Gutub and F. Al-Shaarani, "Efficient implementation of multi-image secret hiding based on lsb and dwt steganography comparisons," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 2631–2644, 2020.
[7] F. S. Hassan and A. Gutub, "Efficient image reversible data hiding technique based on interpolation optimization," *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 8441–8456, 2021.
[8] F. Al-Shaarani and A. Gutub, "Securing matrix counting-based secret-sharing involving crypto steganography," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 6909–6924, 2022.
[9] F. S. Hassan and A. Gutub, "Improving data hiding within colour images using hue component of hsv colour space," *CAAI Transactions on Intelligence Technology*, vol. 7, no. 1, pp. 56–68, 2022.
[10] R. Wang, L. Li, G. Yang, X. Yan, and W. Yan, "Secret cracking and security enhancement for the image application of crt-based secret sharing," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 9819–9834, 2024.
[11] J. Wu, N. Liu, and W. Kang, "The capacity region of distributed multi-user secret sharing under perfect secrecy," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 9954–9969, 2024.
[12] Y.-C. Chen, J.-K. Yang, H.-C. Yen, and P.-W. Lin, "Dual-cloud multi-secret sharing architecture for privacy preserving persistent computation," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 7523–7535, 2024.
[13] R. Bi, J. Xiong, C. Luo, J. Ning, X. Liu, Y. Tian, and Y. Zhang, "Communication-efficient privacy-preserving neural network inference via arithmetic secret sharing," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 6722–6737, 2024.
[14] M. Naor and A. Shamir, "Visual cryptography," *Lecture Notes in Computer Science*, vol. 950, no. 1, pp. 1–12, 1995.
[15] C. Yang, "New visual secret sharing schemes using probabilistic method," *Pattern Recognition Letters*, vol. 25, no. 4, pp. 486–494, 2004.
[16] 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.
[17] 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.
[18] P. Li, J. Ma, and Q. Ma, "(t, k, n) xor-based visual cryptography scheme with essential shadows," *Journal of Visual Communication and Image Representation*, vol. 72, p. 102911, 2020.

[19] S. J. Shyu, "Xor-based visual cryptographic schemes with monotonously increasing and flawless reconstruction properties," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2397–2401, 2018.

[20] C. Lin and W. Tsai, "Visual cryptography for gray-level images by dithering techniques," *Pattern Recognition Letters*, vol. 24, no. 1-3, pp. 349–358, 2003.

[21] Z. Zhou, G. Arce, and G. Di Crescenzo, "Halftone visual cryptography," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2441–2453, 2006.

[22] Z. Wang, G. Arce, and G. Di Crescenzo, "Halftone visual cryptography via error diffusion," *Information Forensics and Security, IEEE Transactions on*, vol. 4, no. 3, pp. 383–396, 2009.

[23] P.-C. Su, T.-F. Tsai, and Y.-C. Chien, "Visual secret sharing in halftone images by multi-scale error diffusion," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12 111–12 138, 2018.

[24] B. Yan, Y. Xiang, and G. Hua, "Improving the visual quality of size-invariant visual cryptography for grayscale images: an analysis-by-synthesis (abs) approach," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 896–911, 2018.

[25] V. V. Panchbhai and S. W. Varade, "Enhanced block based progressive visual secret sharing scheme for grayscale and color image," *Multimedia Tools and Applications*, vol. 83, no. 17, pp. 50 519–50 555, 2024.

[26] S. Shyu, "Visual cryptograms of random grids for general access structures," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 3, pp. 414–424, 2013.

[27] 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.

[28] Z. Liu, G. Zhu, Y.-G. Wang, J. Yang, and S. Kwong, "A novel (t, s, k, n)-threshold visual secret sharing scheme based on access structure partition," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 16, no. 4, pp. 1–21, 2020.

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

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

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] W. Li, C. Peng, W. Tan, Y. Xu, and K. Niu, "A reversible and lossless secret image sharing scheme with authentication for color images," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 10, p. 101854, 2023.

[36] Z. Liu, G. Zhu, Y. Zhang, H. Zhang, and S. Kwong, "An efficient cheating-detectable secret image sharing scheme with smaller share sizes," *Journal of Information Security and Applications*, vol. 81, p. 103709, 2024.

[37] Z. Zhou, Y. Wan, Q. Cui, K. Yu, S. Mumtaz, C.-N. Yang, and M. Guizani, "Blockchain-based secure and efficient secret image sharing with outsourcing computation in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 1, pp. 423–435, 2024.

[38] A. Gutub, N. Al-Juaid, and E. Khan, "Counting-based secret sharing technique for multimedia applications," *Multimedia Tools and Applications*, vol. 78, no. 5, pp. 5591–5619, 2019.

[39] A. Gutub, "Smart fine-tuning of target key generation for trusted counting-based secret sharing," *Arabian Journal for Science and Engineering*, vol. 50, no. 10, pp. 7677–7696, 2025.

[40] A. Gutub, F. Al-Shaarani, and K. Alharthi, "Novel key-integration to safeguard counting-based secret-sharing from possibilities of cyberattack breaches," *Multimedia Tools and Applications*, vol. 84, no. 9, pp. 6445–6471, 2025.

[41] F. Al-Shaarani and A. Gutub, "Increasing participants using counting-based secret sharing via involving matrices and practical steganography," *Arabian Journal for Science and Engineering*, vol. 47, no. 2, pp. 2455–2477, 2022.

[42] A. Gutub and M. Al-Ghamdi, "Hiding shares by multimedia image steganography for optimized counting-based secret sharing," *Multimedia Tools and Applications*, vol. 79, no. 11, pp. 7951–7985, 2020.

[43] A. Gutub, "Watermarking images via counting-based secret sharing for lightweight semi-complete authentication," *International Journal of Information Security and Privacy*, vol. 16, no. 1, pp. 1–18, 2022.

[44] 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.

[45] V. A. Mohammed and P. Samundiswary, "A comprehensive study on medical image security algorithms," in *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies*. IEEE, 2022, pp. 887–892.

[46] A. Mohammed and P. Samundiswary, "Tamper detection and self-recovery in a visual secret sharing based security mechanism for medical records," *IEEE Journal of Biomedical and Health Informatics*, 2024.

[47] C.-N. Yang and D.-S. Wang, "Property analysis of xor-based visual cryptography," *IEEE transactions on circuits and systems for video technology*, vol. 24, no. 2, pp. 189–197, 2014.

[48] X. Yan, L. Liu, Y. Lu, and Q. Gong, "Security analysis and classification of image secret sharing," *Journal of Information Security and Applications*, vol. 47, pp. 208–216, 2019.

[49] A. Mohammed and P. Samundiswary, "Secmiss: secured medical image secret sharing mechanism for smart health applications," *The Visual Computer*, vol. 40, no. 6, pp. 4251–4271, 2024.

[50] C.-N. Yang and Y.-Y. Yang, "On the analysis and design of visual cryptography with error correcting capability," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2465–2479, 2020.

[51] C.-N. Yang, X. Wu, and M.-J. Chung, "Enhancement of information carrying and decoding for visual cryptography with error correction," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 1, pp. 1–24, 2023.

[52] 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.

Xiaotian Wu is currently an Associate Professor with the College of Cyber Security, Jinan University, Guangzhou, China. His research interests include visual cryptography, secret image sharing and multimedia security.

Li Tang received the master's degree with the Department of Computer Science, Jinan University, China. Her research interests are secret image sharing and multimedia security.

Zhihua Xia is currently a Professor with the College of Cyber Security, Jinan University, Guangzhou, China. His research interests include digital forensic and encrypted image processing.

Ching-Nung Yang 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. He is a Fellow of IET.

WeiQi Yan is with AUT Department of Computer and Information Systems. Dr. Yan currently holds the position of Chair of ACM Multimedia Chapter of New Zealand, he is a Fellow of Engineering New Zealand.