Progressive Secret Image Sharing using Multiplicative Primitive Irreducible Polynomials

Xiaotian Wu, *Member, IEEE*, Xinyue Wang, Bing Chen, Zhihua Xia, *Member, IEEE*, Ching-Nung Yang, *Senior Member, IEEE*, WeiQi Yan, *Senior Member, IEEE*

***Abstract*—Progressive secret image sharing (PSIS) encodes a secret into n shadows, enabling the secret to be gradually revealed as more shared data becomes available. However, current techniques suffer from non-uniform progressive recovery and inferior visual quality of low-level decoded images. To address these issues, we combine multiplicative primitive irreducible polynomials with polynomial ring Chinese Remainder Theorem (CRT) to establish a PSIS. A method with 2 progressive levels is presented, where two polynomials $f_1(\alpha)$ and $f_2(\alpha)$ determine a main polynomial $F(\alpha) = f_1(\alpha) \times f_2(\alpha)$. A $(k-1)$-degree polynomial over $(\mathrm{mod}\ F(\alpha))$ is constructed for shadow generation. The secret can be losslessly recovered by Lagrange interpolation over $(\mathrm{mod}\ F(\alpha))$, or gradually recovered through modified shadows using Lagrange interpolation and polynomial ring CRT. The scheme is further extended to t progressive levels, and a bit rearrangement technique is devised to enhance progressive recovery performance. Experiments and comparisons illustrate the effectiveness of the proposed technique, including uniform progressive recovery and superior low-level image quality.**

***Index Terms*—Secret sharing, Secret image sharing, Chinese Remainder Theorem, Polynomial ring, Progressive recovery.**

I. Introduction

Data hiding [1], [2], encryption [3], and secret sharing [4], [5] have been regarded as crucial components for various multimedia security applications. The k-out-of-n secret sharing encodes a secret into n pieces of private information, referred to as shadows or shares, in such a way that any k shadows can decode the secret. Whereas the one gathering $(k-1)$ or fewer shadows is not able to reveal the secret.

The concept of secret sharing is extended to the domain of private image protection. The seminal work of (k, n) secret image sharing (SIS) was presented in [6]. Since then, various approaches have been proposed and studied. In SIS, shadow management is challenging. The random, noise-like appearance of shadows makes them inefficient to handle and, moreover, easily attracts the attention of malicious eavesdroppers. To address this issue, SIS schemes capable of generating meaningful shadows [7]–[9] were subsequently proposed. The traditional (k, n) scheme may constrain application scenarios. Thus, the approaches devised for essential mode [10] were established. To avoid the use of fake shadows in secret recovery, SIS schemes with capability for identifying fraudulent shares [5], [11] were put forth. Recent research on SIS has further diversified, extending into areas such as transform domain sharing [9], progressive recovery [12], [13], cheating-prevention [14], and outsourcing computation [15].

Progressive SIS (PSIS) is a branch of SIS that overcomes the "all or nothing" limitation of classical SIS. In this scenario, PSIS offers the benefit that increasing amounts of secret information are revealed as more shared data is employed. The studies in [16], [17] decode different parts of the secret in a progressive manner by using spatial property and bit plane decomposition. The smoothness of recovered images [18], the size of shadows [19], and essential sharing strategy [20] were investigated in further detail. PSIS based on Boolean operations and polynomial interpolations (PSIS-BP) [21] was recently studied. In this scheme, bit-planes of secret pixels are split into two categories: one for polynomial interpolation and the other for Boolean operation. Different progressive levels are guaranteed based on their assignment approaches. In summary, the foregoing PSIS techniques only reveal the secret losslessly when all n participants are employed. Nevertheless, it is challenging to involve all members in real-world applications. Thus, these techniques suffer from a critical drawback: the secret cannot be perfectly recovered if any shadow is lost or destroyed. Fault tolerance is not enforced.

PSIS that can retrieve a secret among k participants was introduced as a solution to the fault tolerance issue. In [12], Chinese Remainder Theorem (CRT) and polynomial based PSIS (CP-PSIS) and modified CP-PSIS (MCP-PSIS) were investigated. In the (k, n) CP-PSIS scheme, k secret pixels are embedded into the coefficients of a $(k-1)$-degree polynomial to generate n shares. Each share is further decomposed into m sub-shadows using a set of m coprime integers, and these sub-shadows are distributed to the corresponding participant. Any k participants can progressively reconstruct the secret based on their sub-shadows. Once the number of sub-shadows to be used is determined, each participant obtains an intermediate result from these sub-shadows via the CRT technique.

This work was partially supported by National Natural Science Foundation of China (Grant No. U23B2023), Guangdong Key Laboratory of Data Security and Privacy Preserving (Grant No. 2023B1212060036), the Fundamental Research Funds for the Central Universities, and National Science and Technology Council (Grant No. 112-2221-E-259-007-MY2). (Corresponding authors: Zhihua Xia and Ching-Nung Yang)

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

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

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

Subsequently, Lagrange interpolation (LI) is applied to the intermediate results from any k participants to recover the secret. The quality of the reconstructed image depends on the number of sub-shadows employed. Lossless reconstruction is achieved only when all m sub-shadows are involved. MCP-PSIS further exploits the technique of bit flipping to enhance the visual quality. By employing multiple primes, SIS using multiple primes (MP-SIS) [13] was introduced to offer progressive recovery within k participants as well.

Furthermore, an alternative PSIS, which integrates deep learning with reversible data hiding in encrypted images (RD-HEI), was introduced in [22], where a different progressive reconstruction mode is utilized. Depending on the number of shares and the encryption keys, the receiver can progressively decode a low-resolution version of the image, a mask image with the most sensitive regions removed, and finally the complete original image. In [23], a hybrid encryption system was developed that integrates secret sharing and a (k_1, k_2, n) progressive masking for dealing with JPEG images. When the number of shares k equals k_1, the DC coefficients are reconstructed through secret sharing, while the AC coefficients are processed based on the (k_1, k_2, n) progressive masking to achieve a smooth transition in visual quality. As k increases from k_1 to k_2, the AC coefficients are gradually recovered, enabling image decoding from blur to clarity.

However, progressive recovery in most recent approaches is non-uniform, and the clarity of the recovered low-level image is compromised. In this paper, we consider an almost unexplored area in PSIS: by combining multiplicative primitive irreducible polynomials and polynomial ring CRT to provide superior progressive recovery performance. Notably, uniform progressive recovery and enhanced visual quality of low-level decoded image are guaranteed. Contributions of this paper are summarized below.

- A PSIS using multiplicative primitive irreducible polynomials and polynomial ring CRT is presented to offer 2 recovery levels. First, two suitable multiplicative primitive irreducible polynomials $f_1(\alpha)$ and $f_2(\alpha)$ are chosen to determine a main polynomial via $F(\alpha) = f_1(\alpha) \times f_2(\alpha)$. More significantly, the selection of irreducible polynomials is discussed and the feasibility of the proposed technique is rigorously demonstrated. For shadow generation, a $(k-1)$-degree polynomial $g(x) = \sum_{j=0}^{k-1} a_j x^j \pmod{F(\alpha)}$ is constructed, whose k coefficients are employed for secret embedding. By assigning n distinct values x_1, $\cdots$, x_n in the finite field based on $f_1(\alpha)$ as IDs, the corresponding shadow polynomials $g(x_1)$, $\cdots$, $g(x_n)$ are calculated. These polynomial outputs are subsequently converted to pixel values to produce n final shadows. The proposed scheme provides two decoding options: direct decryption and progressive recovery. In direct decryption mode, the secret image can be losslessly recovered from the shadows by LI over $\pmod{F(\alpha)}$. As progressive recovery is performed, the participants can prepare the modified shadows from the original ones and apply LI and polynomial ring CRT to decrypt the secret gradually. The progressive recovery property is also verified theoretically.
- Enhancements on the proposed scheme are developed. By choosing t appropriate multiplicative primitive irreducible polynomials $f_1(\alpha)$, $\cdots$, $f_t(\alpha)$, t progressive levels are enabled. Meantime, theoretical examination on the progressive recovery property is presented. Furthermore, bit rearrangement is designed to improve the clarity of the low-level recovered image and to support uniform progressive recovery. Extensive experiments are performed to demonstrate the benefits of the enhancements.

The rest of this paper is organized as follows. Section II depicts traditional SIS and CRT for polynomial rings. The proposed scheme with 2 recovery levels is presented in Section III. Enhancements are formulated in Section IV. Section V demonstrates the experimental results and discussions. Concluding remark is illustrated in Section VI.

II. Preliminary

In this section, conventional SIS is described, as well as the general form of CRT for polynomial rings.

A. Traditional SIS

By utilizing a $(k-1)$-degree polynomial $g(x) = a_0 + \sum_{j=1}^{k-1} a_j x^j \pmod{p}$, Shamir's (k, n) secret sharing [24] can be implemented. Herein, p is a prime, $a_1, \cdots a_{k-1}$ are randomly-chosen numbers and the secret s is embedded into the constant coefficient a_0 (i.e., $a_0 = s$). The shadows $g(x_1), \cdots, g(x_n)$ are produced by adopting n distinct non-zero variables $x_1, \cdots, x_n$. Any collection having k shadows can recover the polynomial $g(x)$ via LI. And the secret s is achieved as $s = g(0)$. Whereas any $(k-1)$ or fewer shadows disclose no clue about the secret since the k unknown coefficients of $g(x)$ cannot be uniquely determined. Shamir's secret sharing was extended to image domain to deal with sensitive images. In a (k, n) SIS [6], the secret image is first scrambled and any k pixels are concealed into the k coefficients of a $(k-1)$-degree polynomial $g(x)$ to create shadow pixels. The size of a shadow is therefore reduced to $1/k$ of the original image.

B. CRT for Polynomial Rings

Let $m_1(\alpha)$, $m_2(\alpha)$, $\cdots$, $m_n(\alpha) \in \mathbb{F}[\alpha]$ be n pairwise coprime monic polynomials where $\mathbb{F}$ is a field. Given n polynomials $s_1(\alpha)$, $s_2(\alpha)$, $\cdots$, $s_n(\alpha) \in \mathbb{F}[\alpha]$, there exists $S(\alpha)$ satisfying the following condition: $S(\alpha) \equiv s_1(\alpha) \pmod{m_1(\alpha)}$, $\cdots$, $S(\alpha) \equiv s_n(\alpha) \pmod{m_n(\alpha)}$. The solution $S(\alpha)$ can be determined by $S(\alpha) \equiv \sum_{i=1}^{n} \lambda_i(\alpha) M_i(\alpha) s_i(\alpha) \pmod{M(\alpha)}$ where $M(\alpha) = \prod_{i=1}^{n} m_i(\alpha)$, $M_i(\alpha) = M(\alpha)/m_i(\alpha)$, and $\lambda_i(\alpha) \equiv M_i^{-1}(\alpha) \pmod{m_i(\alpha)}$. The uniqueness of $S(\alpha)$ is ensured when only the polynomials of degrees less than $\deg(\prod_{i=1}^{n} m_i(\alpha))$ are taken into consideration, where $\deg(\cdot)$ denotes the degree of input polynomial.

III. The Proposed Method with 2 Progressive Levels

In this section, we analyze previous techniques and provide the motivation of our work. The proposed approach with 2

progressive levels is then presented, including an overview and the implementation details such as shadow generation and secret recovery. The progressive recovery property is theoretically examined at the end of this section.

A. Analysis and Motivation

Current PSIS techniques [12] [13] [21] suffer from several defects, including non-uniform progressive recovery, compromised visual quality in low-level reconstruction, and distortion in the decoded image, as outlined below.

- **Uneven progressive recovery.** Existing techniques, such like CP-PSIS [12], MCP-PSIS [12], MP-SIS [13], and PSIS-BP [21], are subject to a non-uniform progressive recovery flaw. The percentage of correctly decoded secret pixels is not proportional to the amount of involved shared information. Related PSIS schemes with 3 recovery levels are illustrated in Fig. 1 to clarify the non-uniform phenomenon. In a PSIS with uniform progressive recovery, the l-th recovered image $(1 \leq l \leq t)$ is expected to reveal approximately an l/t portion of the original secret where t is the total secret levels. Herein, Fig. 1(a) shows the 3 standard reconstructed images that correctly reveal $1/3$, $2/3$, and $3/3$ of a secret. In contrast, Fig. 1(b)-(e) present the revealed results by current methods [12] [13] [21]. One can observe that previous techniques yield decoded images that deviate substantially from the standard recovered ones, confirming the defect of uneven progressive recovery.
- **Inferior image quality of low-level recovered result.** The low-level decoded images produced by CP-PSIS [12], MP-SIS [13], and PSIS-BP [21] exhibit unsatisfactory visual quality. As demonstrated in Fig. 1 (b), (c), and (e), the reconstructed results appear predominantly noise-like, making it difficult to identify the secret information. Indeed, the visual quality of the 1-st and 2-nd recovery levels in these methods is notably inferior. This lack of clarity arises because most secret pixels that belong to the low-level recovered image are not decoded correctly, leading to a noise-like appearance. Ideally, approximately a fraction l/t of the secret pixels should be accurately recovered in the l-th level $(1 \leq l \leq t)$.
- **Distorted decoded image.** MP-SIS [13] applies the $(\mathrm{mod}\ P)$ calculation. As the integer P cannot be 2^N, it is unable to cope with a secret frame with N bits well. As a result of this, the secret frames might be altered and slight image distortion is introduced.

Motivated by the aforementioned limitations of existing schemes, this paper aims to overcome these challenges. The mechanism of using multiplicative primitive irreducible polynomials and polynomial ring CRT for providing progressive decoding is investigated. Particularly, the approach with 2 progressive levels is first introduced. Overview of this technique and implementation details such as shadow construction and secret recovery are described in the following sub-sections.

Additionally, the non-uniform progressive recovery of existing methods do not compromise the security, as less than k participants cannot obtain any clue about the secret. Instead, the non-uniformity greatly restricts the application scenario. Consider the progressive decoding of the 1-st level: current methods might disclose a noise-like secret or almost the entire secret, which is unreasonable for applications.

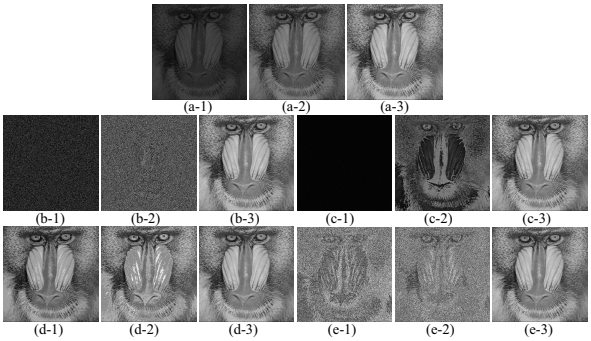

Fig. 1. The non-uniform progressive recovery phenomenon for current PSIS schemes with 3 progressive recovery levels. (a) Standard decoded images, (b) MP-SIS, (c) CP-PSIS, (d) MCP-PSIS, and (e) PSIS-BP.

B. Overview

The proposed technique with 2 progressive levels consists of 2 central building blocks: shadow generation and secret recovery, as depicted in Fig. 2. In shadow construction, two multiplicative primitive irreducible polynomials $f_1(\alpha)$ and $f_2(\alpha)$ are carefully determined according to the criteria. A main polynomial is constructed via $F(\alpha) = f_1(\alpha) \times f_2(\alpha)$. Then, the secret pixels are converted into secret polynomials which are then regarded as the k coefficients of a $(k-1)$-degree polynomial under $(\mathrm{mod}\ F(\alpha))$. Finally, the shadow polynomials, generated from the $(k-1)$-degree polynomial, are transformed to pixel values and n shadows are achieved. Note that, all calculations in our method are performed over $(\mathrm{mod}\ F(\alpha))$, where $F(\alpha) = f_1(\alpha) \times f_2(\alpha)$ is a polynomial. This is fundamentally distinct from existing approaches that operate over $(\mathrm{mod}\ p)$, where p is a prime or a product of coprime integers. The key differences are as follows. First, existing methods perform integer modular arithmetic, where the operands are integers and the modular reduction is based on integer division. In contrast, our method performs polynomial modular arithmetic, where the operands are binary polynomials and the modular reduction is based on polynomial division. Second, the progressive recovery mechanisms and reconstruction capabilities differ accordingly. Integer-based methods [13] decompose shares into sub-shares using coprime integers via the integer CRT, and suffer from image distortion because p cannot equal 2^N, making it unable to perfectly accommodate N-bit secret frames. Our method factors $F(\alpha)$ into irreducible polynomials and applies the polynomial ring CRT, which decomposes the secret at the bit level and exactly represents all $8m$-bit secret frames with $\deg(F(\alpha)) = 8m$, enabling distortion-free reconstruction.

In secret reconstruction, two decoding alternatives are available: direct decoding and progressive recovery. Direct decoding permits the lossless reconstruction of secret by applying LI under $(\mathrm{mod}\ F(\alpha))$ to the original shadows. On the other hand, progressive recovery allows the gradual decryption of

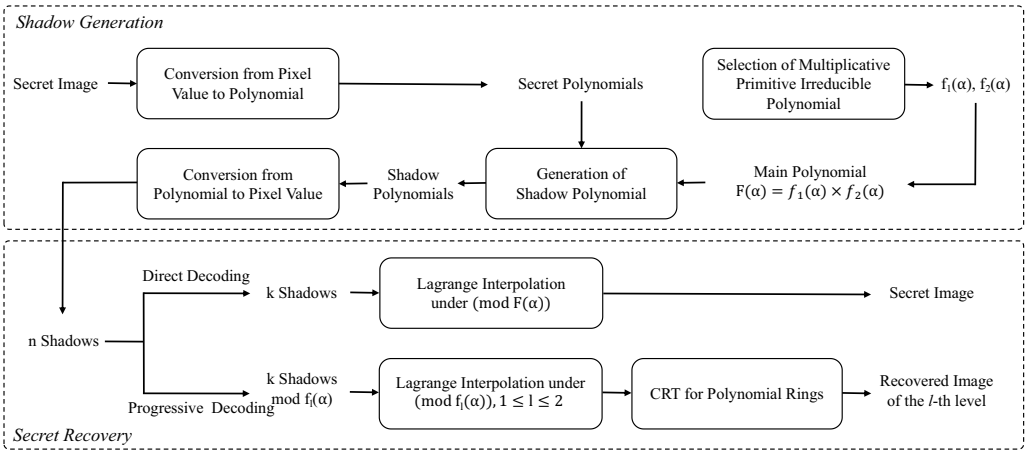

Fig. 2. Diagram of the proposed PSIS with 2 progressive levels.

secret. In specific, participants can prepare their modified shadows from the original shadows. One can disclose the secret progressively from the modified shadows through the usage of LI and polynomial ring CRT.

Direct decoding and progressive recovery are designed for different application scenarios. Direct decoding is suitable for conventional applications where the secret must be recovered completely in a single attempt. In contrast, progressive recovery is particularly advantageous for real-time and resource-constrained applications. This method initially delivers a coarse-grained version of the secret, allowing for immediate recognition without incurring substantial latency. Finer reconstructions are then generated on demand. Users maintain full control to terminate the refinement process at any stage, thereby dynamically balancing final reconstruction quality with storage overhead. This flexible, on-demand refinement mechanism reduces the volume of data that must be transmitted per secret, which in turn lowers communication latency and saves bandwidth, which are critical factors for real-time and resource-limited systems.

C. Shadow Construction

Two suitable multiplicative primitive irreducible polynomials $f_1(\alpha)$ and $f_2(\alpha)$ are selected and used to calculate $F(\alpha) = f_1(\alpha) \times f_2(\alpha)$. Then, a $(k-1)$-degree polynomial $g(x) = \sum_{j=0}^{k-1} a_j x^j \pmod{F(\alpha)}$ is established to produce shadows. If all k coefficients ($a_0 \sim a_{k-1}$) are exploited for secret embedding, the secret image should be randomly permuted first and then separated into non-overlapping frames. For each frame, it is supposed to have m 8-bit pixels. Hence, totally $8m$ bits would be processed each time. More concretely, every coefficient a_j is actually a secret polynomial transformed from a frame having m pixels. To guarantee that the $8m$ bits of a secret frame can be correctly converted, the two selected primitive irreducible polynomials $f_1(\alpha)$ and $f_2(\alpha)$ must satisfy $\deg(f_1(\alpha)) + \deg(f_2(\alpha)) = 8m$. As $\deg(F(\alpha)) = \deg(f_1(\alpha)) + \deg(f_2(\alpha)) = 8m$, the converted values would fall within the valid range. Some possible selections of $f_1(\alpha)$ and $f_2(\alpha)$ can refer to Table I. By considering n different values x_1, x_2, $\cdots$, and x_n (except value "0") in $GF(f_1(\alpha))$ as IDs, n shadow polynomials $g(x_i)$, $1 \le i \le n$, are derived. Herein, $GF(f_1(\alpha))$ is the finite field defined by $f_1(\alpha)$. These shadow polynomials are transformed back to pixel values for constructing n shadow images. Finally, these n shadows are securely and privately sent to n participants.

Main steps of shadow construction are depicted in Algorithm 1. The kernel procedures, such as selection of multiplicative primitive irreducible polynomial and conversion between pixel value and polynomial, are given as follows.

*1) **Selection of Multiplicative Primitive Irreducible Polynomial**:* In our PSIS, two multiplicative primitive irreducible polynomials $f_1(\alpha)$ and $f_2(\alpha)$ should be carefully chosen to satisfy the following progressive requirements: (i) Partial secret bits can be correctly recovered when one irreducible polynomial (e.g., $f_2(\alpha)$) is employed; (ii) All secret bits are decoded with the usage of two irreducible polynomials. As more secret bits are recovered by using two irreducible polynomials, the 2-level decoding is guaranteed.

Without loss of generality, let $\deg(f_1(\alpha)) \le \deg(f_2(\alpha))$. When dealing with a secret frame having m 8-bit pixels, the degrees of $f_1(\alpha)$ and $f_2(\alpha)$ should satisfy $\deg(f_1(\alpha)) + \deg(f_2(\alpha)) = 8m$. Then, a main polynomial is established by $F(\alpha) = f_1(\alpha) \times f_2(\alpha)$. Table I shows some different combinations of two alternative multiplicative primitive irreducible polynomials[1], to tackle a secret frame e containing 2, 3 and 4 pixels, respectively, each time. Different combinations offer different progressive recovery capabilities. The polynomial $f_1(\alpha)$ used in our scheme can be $\alpha^3+\alpha+1$ or $\alpha^4+\alpha+1$. Note: our approach can cope with more pixels using a polynomial $f_2(\alpha)$ of higher degree.

Consider $F(\alpha) = f_1(\alpha) \times f_2(\alpha)$ with any two different primitive irreducible polynomials in Table I. $F(\alpha) \in \mathbb{Z}_2[\alpha]$ is a polynomial of degree 16, 24 and 32 for handling a secret frame e having 2, 3 and 4 pixels, respectively, where $\mathbb{Z}_2[\alpha]$ denotes the polynomial ring over the binary field $\mathbb{Z}_2 = \{0, 1\}$. One can use $F(\alpha)$ to construct a set G_F, which is defined as

$$G_F = \{e(\alpha) \in \mathbb{Z}_2[\alpha] | \deg(e(\alpha)) < \deg(F(\alpha))\}. \quad (1)$$

[1]The list of primitive irreducible polynomials with degree 2 to 32 can be found via the following link: https://www.partow.net/programming/polynomials/index.html

TABLE I
MULTIPLICATIVE PRIMITIVE IRREDUCIBLE POLYNOMIALS BY OUR PSIS FOR 2, 3 AND 4 PIXELS

$f_1(\alpha)$	$f_2(\alpha)$ for m pixels		
	$m=2$	$m=3$	$m=4$
$\alpha^3+\alpha+1$	$\alpha^{13}+\alpha^4+\alpha^3+\alpha+1$	$\alpha^{21}+\alpha^2+1$	$\alpha^{29}+\alpha^2+1$ $\alpha^{29}+\alpha^{20}+\alpha^{11}+\alpha^2+1$ $\alpha^{29}+\alpha^{20}+\alpha^{16}+\alpha^{11}+\alpha^8+\alpha^4+\alpha^3+\alpha^2+1$
$\alpha^4+\alpha+1$	$\alpha^{12}+\alpha^9+\alpha^3+\alpha^2+1$ $\alpha^{12}+\alpha^9+\alpha^8+\alpha^3+\alpha^2+\alpha+1$	$\alpha^{20}+\alpha^3+1$	$\alpha^{28}+\alpha^3+1$

More importantly, the polynomial $e(\alpha)$ is actually converted from a secret frame e. Obviously, the less $\deg(f_1(\alpha))$ and suitable $f_1(\alpha)$ may not affect some most significant bits (MSBs) in some pixels when decoding the secret gradually. Theorem 1 shows that the proposed technique with 2 progressive levels based on a $(k-1)$-degree polynomial can be accomplished when the value n of our (k,n) PSIS is restricted to $\deg(f_1(\alpha))$. In Theorem 1, we consider the primitive irreducible polynomials $f_1(\alpha)$ and $f_2(\alpha)$ which are depicted in Table I. To deal with $m=2$ pixels each time, one can choose $f_1(\alpha)=\alpha^3+\alpha+1$ and $f_2(\alpha)=\alpha^{13}+\alpha^4+\alpha^3+\alpha+1$ to construct the main polynomial $F(\alpha)=f_1(\alpha)\times f_2(\alpha)$. Additionally, the $(k-1)$-degree polynomial is defined as $g(x)=a_0+a_1x+\cdots+a_{k-1}x^{k-1} \pmod{F(\alpha)}$, where each coefficient of $g(x)$ is a polynomial modulo $F(\alpha)$. The uniqueness of interpolation over $\pmod{F(\alpha)}$ is verified by Theorem 2. To comprehend the process of LI over $\pmod{F(\alpha)}$, Example 1 is presented.

Theorem 1. *Our (k,n) PSIS based on a $(k-1)$-degree polynomial $g(x)$ modulo $F(\alpha)$ can be accomplished over two suitable multiplicative primitive irreducible polynomials $f_1(\alpha)$ and $f_2(\alpha)$. The number of involved participants n in this (k,n) PSIS can be realized up to 7 and 15, when $f_1(\alpha)=\alpha^3+\alpha+1$ and $f_1(\alpha)=\alpha^4+\alpha+1$ are employed, respectively.*

Proof. When generating shadows by a $(k-1)$-degree polynomial $g(x)=a_0+a_1x+\cdots+a_{k-1}x^{k-1} \pmod{F(\alpha)}$, it is obvious that we can calculate n shadow pixels $g(x_i)$, $1\le i\le n$. W.l.o.g., let x_i be n distinct nonzero elements in $GF(\alpha^3+\alpha+1)$ or $GF(\alpha^4+\alpha+1)$. Thus, the generation of shadows can be accomplished and the number of participants is realized up to 7 or 15.

About recovering $g(x)$ from any k shadow pixels (say $g(x_1)\sim g(x_k)$) by LI, we may obtain the interpolation polynomial

$$g(x)=\sum_{i=1}^{k}\left(g(x_i)\prod_{\substack{j\in[1,k]\\ j\neq i}}\frac{x-x_j}{x_i-x_j}\right) \pmod{F(\alpha)}. \quad (2)$$

The interpolation utilizes addition, multiplication, and their inverse operations: subtraction and division. Among these, addition, subtraction, and multiplication can be accomplished directly modulo $F(\alpha)$. We now examine the division operation in this environment. For the interpolation described, division becomes feasible provided that the multiplicative inverse of (x_i-x_j) modulo $F(\alpha)$ exists. According to the modular multiplicative inverse theorem, a polynomial $s(\alpha)$ has a multiplicative inverse modulo $F(\alpha)$ if and only if $s(\alpha)$ and $F(\alpha)$ are coprime. Thus, by applying this theorem, we can choose the IDs from $GF(f_1(\alpha))$ to ensure that the multiplicative inverse of (x_i-x_j) modulo $F(\alpha)$ does indeed exist. The proof is given below.

Given that x_i and x_j are distinct nonzero elements in $GF(f_1(\alpha))$, the difference (x_i-x_j) is a nonzero element in $GF(f_1(\alpha))$. Since $f_1(\alpha)$ is primitive and irreducible, and (x_i-x_j) is a nonzero element of the field, (x_i-x_j) is coprime to $f_1(\alpha)$. Moreover, as indicated in Table I, $\deg(f_2(\alpha))\ge\deg(f_1(\alpha))$ and $f_2(\alpha)$ is also a primitive irreducible polynomial, it follows that (x_i-x_j) is likewise coprime to $f_2(\alpha)$. Therefore, we have (x_i-x_j) is coprime to $F(\alpha)$ because $F(\alpha)=f_1(\alpha)\times f_2(\alpha)$ and (x_i-x_j) is coprime to both factors. Based on the modular multiplicative inverse theorem, the multiplicative inverse of (x_i-x_j) modulo $F(\alpha)$ exists as (x_i-x_j) and $F(\alpha)$ are coprime. In this case, the division operation is applicable. Since all operations of LI can be realized over $\pmod{F(\alpha)}$, the secret recovery becomes achievable.

□

Theorem 2. *Let $g(x)=\sum_{j=0}^{k-1}a_jx^j \pmod{F(\alpha)}$ be a polynomial of degree at most $(k-1)$ with coefficients over $\pmod{F(\alpha)}$, where $F(\alpha)=f_1(\alpha)\times f_2(\alpha)$ and $f_1(\alpha)$, $f_2(\alpha)$ are two distinct primitive irreducible polynomials over $\mathbb{Z}_2[\alpha]$ with $\deg(f_1(\alpha))\le\deg(f_2(\alpha))$. Given k distinct evaluation points $x_1,\ldots,x_k$ chosen from $GF(f_1(\alpha))$, the polynomial $g(x)$ is uniquely determined by its values $g(x_1),\ldots,g(x_k)$.*

Proof. Suppose $g(x)$ and $g_1(x)$ are two polynomials of degree at most $(k-1)$ over $\pmod{F(\alpha)}$ such that $g(x_i)=g_1(x_i)$ for all $i=1,\cdots,k$. Define $h(x)=g(x)-g_1(x)=\sum_{j=0}^{k-1}c_jx^j$. Then $h(x)$ has degree at most $(k-1)$ and satisfies $h(x_i)=0$ for all $i=1,\cdots,k$. We show that $c_j=0$ for all j, which implies $g(x)=g_1(x)$.

Since $f_1(\alpha)$ and $f_2(\alpha)$ are distinct irreducible polynomials, they are coprime. By the CRT for polynomial rings, the arithmetic over $\pmod{F(\alpha)}$ can be decomposed as:

$$(\bmod F(\alpha))\cong(\bmod f_1(\alpha))\times(\bmod f_2(\alpha)), \quad (3)$$

i.e., computations over $\pmod{F(\alpha)}$ are equivalent to performing computations over $\pmod{f_1(\alpha)}$ and $\pmod{f_2(\alpha)}$ simultaneously. Under this decomposition, each coefficient c_j over $\pmod{F(\alpha)}$ corresponds to a pair $(c_j^{(1)},c_j^{(2)})$ where

$c_j^{(1)}$ is over $(\mathrm{mod}\ f_1(\alpha))$ and $c_j^{(2)}$ is over $(\mathrm{mod}\ f_2(\alpha))$. The polynomial $h(x)$ decomposes accordingly into:

$$h^{(1)}(x) = \sum_{j=0}^{k-1} c_j^{(1)} x^j \pmod{f_1(\alpha)}, \tag{4}$$

$$h^{(2)}(x) = \sum_{j=0}^{k-1} c_j^{(2)} x^j \pmod{f_2(\alpha)}. \tag{5}$$

The condition $h(x_i) = 0$ over $(\mathrm{mod}\ F(\alpha))$ is equivalent to $h^{(1)}(x_i) = 0$ over $(\mathrm{mod}\ f_1(\alpha))$ and $h^{(2)}(x_i) = 0$ over $(\mathrm{mod}\ f_2(\alpha))$, for all $i = 1, \cdots, k$. We now show that both $h^{(1)}(x)$ and $h^{(2)}(x)$ are zero polynomials.

For $h^{(1)}(x)$: The points $x_1, \cdots, x_k$ are k distinct elements in $GF(f_1(\alpha))$, and $h^{(1)}(x)$ is a polynomial of degree at most $(k-1)$ over the field $GF(f_1(\alpha))$ that vanishes at all k points. Since a nonzero polynomial of degree at most $(k-1)$ over a field has at most $(k-1)$ roots, $h^{(1)}(x)$ must be the zero polynomial, i.e., $c_j^{(1)} = 0$ for all j.

For $h^{(2)}(x)$: The points $x_1, \cdots, x_k$ are elements of $GF(f_1(\alpha))$, i.e., they are polynomials in $\mathbb{Z}_2[\alpha]$ of degree less than $\deg(f_1(\alpha))$. Since $\deg(f_1(\alpha)) \leq \deg(f_2(\alpha))$, these polynomials also have degree less than $\deg(f_2(\alpha))$, and hence represent valid elements in $GF(f_2(\alpha))$. Moreover, distinct polynomials of degree less than $\deg(f_2(\alpha))$ correspond to distinct elements in $GF(f_2(\alpha))$, so $x_1, \cdots, x_k$ remain pairwise distinct in $GF(f_2(\alpha))$. Therefore, $h^{(2)}(x)$ is a polynomial of degree at most $(k-1)$ over the field $GF(f_2(\alpha))$ with k distinct roots, and by the same argument, $h^{(2)}(x)$ is the zero polynomial, i.e., $c_j^{(2)} = 0$ for all j.

Since $c_j^{(1)} = 0$ and $c_j^{(2)} = 0$ for all j, under the CRT decomposition we have $c_j = 0$ over $(\mathrm{mod}\ F(\alpha))$ for all j. Therefore $h(x)$ is the zero polynomial, which implies $g(x) = g_1(x)$.

□

Example 1. *Consider a* $(2,2)$ *PSIS based on a* 1*-degree polynomial* $g(x) = a_0 + a_1 x \pmod{F(\alpha)}$. *Herein,* $f_1(\alpha) = \alpha^3 + \alpha + 1$, $f_2(\alpha) = \alpha^{13} + \alpha^4 + \alpha^3 + \alpha + 1$, *and* $F(\alpha) = f_1(\alpha) \times f_2(\alpha) = \alpha^{16} + \alpha^{14} + \alpha^{13} + \alpha^7 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^2 + 1$. *Let the four embedded secret pixels be* $64, 129, 33, 8$ *with the following polynomial forms:*

$$\{64, 129\} \Rightarrow \alpha^{14} + \alpha^7 + 1, \{33, 8\} \Rightarrow \alpha^{13} + \alpha^8 + \alpha^3. \tag{6}$$

The polynomial $g(x)$ *for generating shadows is denoted by*

$$g(x) = (\alpha^{14} + \alpha^7 + 1)x + (\alpha^{13} + \alpha^8 + \alpha^3) \pmod{F(\alpha)}. \tag{7}$$

One can choose 2 *participant IDs* x_i *and* x_j *from* $GF(f_1(\alpha))$ *(e.g.,* $x_i = \alpha^2 + 1, x_j = \alpha$*) to produce two shadows:*

$$\begin{cases} g(x_i) &= \alpha^9 + \alpha^8 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3, \\ g(x_j) &= \alpha^{15} + \alpha^{13} + \alpha^3 + \alpha. \end{cases} \tag{8}$$

Based on $(x_i, g(x_i))$ *and* $(x_j, g(x_j))$, *the LI over* $(\mathrm{mod}\ F(\alpha))$ *can be performed as follows for direct decoding.*

$$\begin{cases} g(x) = g(x_i)\frac{x-x_j}{x_i-x_j} + g(x_j)\frac{x-x_i}{x_j-x_i} \pmod{F(\alpha)} \\ = g(x_i)\frac{x-\alpha}{\alpha^2+1-\alpha} + g(x_j)\frac{x-(\alpha^2+1)}{\alpha-(\alpha^2+1)} \pmod{F(\alpha)} \\ = (\alpha^9 + \alpha^8 + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3)(x-\alpha)(\alpha^2+\alpha+1)^{-1} \\ +(\alpha^{15} + \alpha^{13} + \alpha^3 + \alpha)(x-(\alpha^2+1))(\alpha^2+\alpha+1)^{-1} \\ \pmod{F(\alpha)} \end{cases} \tag{9}$$

Note that, since x_i, x_j $(x_i \neq x_j)$ *are nonzero elements chosen from* $GF(f_1(\alpha))$, *we have totally* 7 *different values of* $(x_i - x_j) \pmod{F(\alpha)}$. *Table II lists all possible values of* $(x_i - x_j) \pmod{F(\alpha)}$ *and their multiplicative inverses. In this example, we have* $(\alpha^2+\alpha+1)^{-1} = \alpha^{15} + \alpha^{14} + \alpha^{13} + \alpha^{12} + \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^4 \pmod{F(\alpha)}$. *By replacing the multiplicative inverse into Eq. (9), we can reconstruct* $g(x)$ *by*

$$g(x) = (\alpha^{14} + \alpha^7 + 1)x + (\alpha^{13} + \alpha^8 + \alpha^3) \pmod{F(\alpha)}. \tag{10}$$

The secret pixels can be extracted as $64, 129, 33, 8$.

TABLE II
$(x_i - x_j) \pmod{F(\alpha)}$ AND ITS MULTIPLICATIVE INVERSE

$(x_i - x_j) \pmod{F(\alpha)}$	Multiplicative inverse
1	1
α	$\alpha^{15} + \alpha^{13} + \alpha^{12} + \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha$
$\alpha + 1$	$\alpha^{14} + \alpha^{12} + \alpha^{11} + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + 1$
α^2	$\alpha^{15} + \alpha^{14} + \alpha^{12} + \alpha^{11} + \alpha^{10} + \alpha^9 + \alpha^8 + \alpha^7 + \alpha^5 + \alpha^3 + \alpha^2$
$\alpha^2 + 1$	$\alpha^{15} + \alpha^{12} + \alpha^{10} + \alpha^8 + \alpha^5 + \alpha^4 + \alpha^2 + \alpha + 1$
$\alpha^2 + \alpha$	$\alpha^{14} + \alpha^{13} + \alpha^{11} + \alpha^{10} + \alpha^9 + \alpha^8 + \alpha^7 + \alpha^6 + \alpha^4 + \alpha^2 + \alpha$
$\alpha^2 + \alpha + 1$	$\alpha^{15} + \alpha^{14} + \alpha^{13} + \alpha^{12} + \alpha^{10} + \alpha^9 + \alpha^7 + \alpha^4 + \alpha^2 + 1$

2) ***Conversion between Pixel Value and Polynomial****:* Let $e = \{s_1, \cdots, s_m\}$ be a secret frame having m pixels and its polynomial form is defined as $e(\alpha)$. Suppose $s_i = (s_i^{(0)} \cdots s_i^{(7)})_2$ is the binary representation of s_i where $s_i^{(0)}$ and $s_i^{(7)}$ are the least and most significant bits of s_i, respectively. The binary representation of e is given as

$$e = (s_1^{(0)} \cdots s_1^{(7)} \cdots s_m^{(0)} \cdots s_m^{(7)})_2. \tag{11}$$

Based on the degrees of $f_1(\alpha)$ and $f_2(\alpha)$, e is decomposed into two components e_0 and e_1 where e_0 contains the first $(8m - \deg(f_1(\alpha))$ bits of e and e_1 comprises the remaining $\deg(f_1(\alpha))$ bits. We further denote e_0 and e_1 as

$$\begin{cases} e_0 = (e_{0,0} e_{0,1} \cdots e_{0,8m-1-\deg(f_1(\alpha))})_2, \\ e_1 = (e_{1,0} e_{1,1} \cdots e_{1,\deg(f_1(\alpha))-1})_2. \end{cases} \tag{12}$$

The polynomial forms of e_0 and e_1 are represented by

$$\begin{cases} e_0(\alpha) = & e_{0,0} + e_{0,1}\alpha + \cdots + \\ & e_{0,8m-1-\deg(f_1(\alpha))}\alpha^{8m-1-\deg(f_1(\alpha))}, \\ e_1(\alpha) = & e_{1,0} + e_{1,1}\alpha + \cdots + \\ & e_{1,\deg(f_1(\alpha))-1}\alpha^{\deg(f_1(\alpha))-1}. \end{cases} \tag{13}$$

Based on $e_0(\alpha)$ and $e_1(\alpha)$, $e(\alpha)$ is rewritten as

$$\begin{aligned} e(\alpha) =& e_0(\alpha) + e_1(\alpha) \times \alpha^{8m-\deg(f_1(\alpha))} \\ =& \underbrace{e_{0,0} + \cdots + e_{0,8m-1-\deg(f_1(\alpha))}\alpha^{8m-1-\deg(f_1(\alpha))}}_{e_0(\alpha)} + \\ & \underbrace{e_{1,0}\alpha^{8m-\deg(f_1(\alpha))} + \cdots + e_{1,\deg(f_1(\alpha))-1}\alpha^{8m-1}}_{e_1(\alpha)\times\alpha^{8m-\deg(f_1(\alpha))}} \\ =& \overline{e}_0 + \overline{e}_1\alpha + \cdots + \overline{e}_{8m-1}\alpha^{8m-1} \end{aligned} \tag{14}$$

where $\overline{e}_0 = e_{0,0}$, $\cdots$, $\overline{e}_{8m-1} = e_{1,\deg(f_1(\alpha))-1}$. For brevity, symbol $\tau(\cdot)$ stands for the above conversion: $e(\alpha) = \tau(e)$.

On the other hand, given a $(8m-1)$-degree polynomial $e(\alpha) = \overline{e}_0 + \overline{e}_1\alpha + \cdots + \overline{e}_{8m-1}\alpha^{8m-1}$, by collecting the binary coefficients $\overline{e}_0$, $\cdots$, $\overline{e}_{8m-1}$, one can derive m pixels by $c_1 = (\overline{e}_0\overline{e}_1\cdots\overline{e}_7)_2$, $\cdots$, $c_m = (\overline{e}_{8m-8}\overline{e}_{8m-7}\cdots\overline{e}_{8m-1})_2$. The following procedure denotes the polynomial-to-pixel value conversion: $[c_1, \cdots, c_m] = \phi(e(\alpha))$.

Algorithm 1 Shadow Generation of the (k, n) proposed scheme with 2 progressive levels

Input: a secret image S and frame size m.
Output: n shadows $C_1, \cdots, C_n$.
(1) Scramble the secret image randomly;
(2) Choose 2 multiplicative primitive irreducible polynomials $f_1(\alpha)$ and $f_2(\alpha)$ whose degrees satisfy $\deg(f_1(\alpha)) + \deg(f_2(\alpha)) = 8m$;
(3) Calculate $F(\alpha) = f_1(\alpha) \times f_2(\alpha)$;
(4) Encode the permuted image by the followings:
For every k frames in a permuted image **Do**
  For every frame e_j having m pixels in the k frames **Do**
    $e_j(\alpha) = \tau(e_j)$, $1 \le j \le k$;
    /* Convert e_j to its polynomial form $e_j(\alpha)$; */
  End For
End For
Use k secret polynomials $e_1(\alpha)$, $\cdots$, $e_k(\alpha)$ as the coefficients of a $(k-1)$-degree polynomial $g(x) = a_0 + \cdots + a_{k-1}x^{k-1} \pmod{F(\alpha)}$, i.e., $a_0 = e_1(\alpha), \cdots, a_{k-1} = e_k(\alpha)$, to produce n shadow polynomials $g(x_1)$, $\cdots$, $g(x_n)$;
For every shadow polynomial in the n ones **Do**
  $[c_1^{(i)}, \cdots, c_m^{(i)}] = \phi(g(x_i))$, $1 \le i \le n$;
  /* Convert $g(x_i)$ to m shadow pixels $c_1^{(i)}, \cdots, c_m^{(i)}$.*/
End For
End For
(5) Collect all the i-th shadow pixels to derive a shadow C_i, $1 \le i \le n$.

D. Secret Recovery

Two decoding options, direct recovery and progressive decoding, are available. When direct decoding is considered, one can apply LI under $\pmod{F(\alpha)}$ to the original shadows for recovering the polynomial $g(x)$. Consequently, the secret image is perfectly decrypted. On the other hand, participants are capable of preparing modified shadows from the original shadows modulo $f_1(\alpha)$ and $f_2(\alpha)$. For the 1-st recovery level, partial secret information is restored from the modified shadows modulo $f_2(\alpha)$ by employing LI under $\pmod{f_2(\alpha)}$. For the 2-nd secret level, the polynomial $g_l(x)$ (i.e., $g(x) \pmod{f_{3-l}(\alpha)}$), $1 \le l \le 2$, can be reconstructed via LI over $\pmod{f_{3-l}(\alpha)}$. With $g_1(x)$ and $g_2(x)$, the 2-nd secret level is decoded by using CRT for polynomial rings

$$g^{(2)}(x) = CRT(g_1(x), g_2(x)). \tag{15}$$

Algorithm 2 Direct Decoding of the (k, n) proposed scheme with 2 progressive levels

Input: any k shadows (say C_1, $\cdots$, C_k)
Output: secret image S
For every k shadow frames in $C_1, \cdots, C_k$ **Do**
  Convert the k shadow frames to their polynomial forms $g(x_1), \cdots, g(x_k)$;
  Use $g(x_1), \cdots, g(x_k)$ to decode $g(x) = \sum_{i=1}^{k}(g(x_i)\prod_{j\in[1,k],j\ne i}\frac{x-x_j}{x_i-x_j}) \pmod{F(\alpha)}$;
  Collect the k coefficients $e_1(\alpha)$, $\cdots$, $e_k(\alpha)$ of $g(x)$;
  /* Each coefficient is a secret polynomial.*/
  Convert $e_j(\alpha)$ to m secret pixels $s_{j,1}, \cdots, s_{j,m}$;
End For
Collect all recovered pixels to form the secret image S.

Algorithm 3 Progressive Recovery of the (k, n) proposed scheme with 2 progressive levels

Input: any modified shadows from k participants (say $C_1^{(l)}$, $\cdots$, $C_k^{(l)}$, $1 \le l \le 2$)
Output: the recovered images with 2 progressive levels S_1 and S_2.
Prepare k modified shadows $C_1^{(l)}, \cdots, C_k^{(l)}$ from the original shadows modulo a primitive irreducible polynomial $f_{3-l}(\alpha)$ for $l = 1$ to 2 in sequence;
For $(l = 1; l \le 2; l++)$ **Do**
  For every k frames in $C_1^{(l)}, \cdots, C_k^{(l)}$ **Do**
    Convert the k shadow frames to their polynomial forms $g_l(x_1), \cdots, g_l(x_k)$;
    Use $g_l(x_1), \cdots, g_l(x_k)$ to decode $g_l(x) = \sum_{i=1}^{k}(g_l(x_i)\prod_{j\in[1,k],j\ne i}\frac{x-x_j}{x_i-x_j}) \pmod{f_{3-l}(\alpha)}$;
    Obtain $g^{(l)}(x) = CRT(g_1(x), \cdots, g_l(x))$;
    /* Use polynomial ring CRT to decode $g^{(l)}(x)$. */
    Collect the k coefficients of $g^{(l)}(x)$;
    /* Each coefficient of $g^{(l)}(x)$ is a progressive secret polynomial.*/
    For every progressive secret polynomial in k ones **Do**
      Convert the polynomial to pixel values;
    End For
  End For
  Collect all recovered pixels to progressively decode S_l where $1 \le l \le 2$ and $S = S_2$.
End For

Algorithms 2 and 3 illustrate the two recovery procedures. Notably, the proposed method offers two alternative decoding strategies tailored to distinct application scenarios. For conventional contexts in which a secret must be recovered in a single decryption attempt, direct decoding is employed. In scenarios

requiring gradual revelation of the secret, progressive decoding is adopted.

E. Progressive Recovery Property

The 2-level progressive recovery property is verified theoretically. Essentially, this functionality is determined by the two chosen multiplicative primitive irreducible polynomials $f_1(\alpha)$ and $f_2(\alpha)$. Herein, we apply various $f_1(\alpha)$ and $f_2(\alpha)$ in Table I to produce 9 different main polynomials $F_1(\alpha)$, $\cdots$, $F_9(\alpha)$, as shown in Eq. (16).

$$\begin{cases} F_1(\alpha) =(\alpha^3+\alpha+1)(\alpha^{13}+\alpha^4+\alpha^3+\alpha+1) \\ \quad =\alpha^{16}+\alpha^{14}+\alpha^{13}+\alpha^7+\alpha^6+\alpha^5+\alpha^4+\alpha^2+1, \\ F_2(\alpha) =(\alpha^3+\alpha+1)(\alpha^{21}+\alpha^2+1) \\ \quad =\alpha^{24}+\alpha^{22}+\alpha^{21}+\alpha^5+\alpha^2+1, \\ F_3(\alpha) =(\alpha^3+\alpha+1)(\alpha^{29}+\alpha^2+1) \\ \quad =\alpha^{32}+\alpha^{30}+\alpha^{29}+\alpha^5+\alpha^2+1, \\ F_4(\alpha) =(\alpha^3+\alpha+1)(\alpha^{29}+\alpha^{20}+\alpha^{11}+\alpha^2+1) \\ \quad =\alpha^{32}+\alpha^{30}+\alpha^{29}+\alpha^{23}+\alpha^{21}+\alpha^{20}+\alpha^{14} \\ \qquad +\alpha^{12}+\alpha^{11}+\alpha^5+\alpha^2+1, \\ F_5(\alpha) =(\alpha^3+\alpha+1)(\alpha^{29}+\alpha^{20}+\alpha^{16}+\alpha^{11}+\alpha^8 \\ \qquad +\alpha^4+\alpha^3+\alpha^2+1) \\ \quad =\alpha^{32}+\alpha^{30}+\alpha^{29}+\alpha^{23}+\alpha^{21}+\alpha^{20}+\alpha^{19} \\ \qquad +\alpha^{17}+\alpha^{16}+\alpha^{14}+\alpha^{12}+\alpha^9+\alpha^8+\alpha^7 \\ \qquad +\alpha^6+\alpha^3+\alpha^2+\alpha+1, \\ F_6(\alpha) =(\alpha^4+\alpha+1)(\alpha^{12}+\alpha^9+\alpha^3+\alpha^2+1) \\ \quad =\alpha^{16}+\alpha^{12}+\alpha^{10}+\alpha^9+\alpha^7+\alpha^6+\alpha^2+\alpha+1, \\ F_7(\alpha) =(\alpha^4+\alpha+1)(\alpha^{12}+\alpha^9+\alpha^8+\alpha^3+\alpha^2+\alpha+1) \\ \quad =\alpha^{16}+\alpha^{10}+\alpha^8+\alpha^7+\alpha^6+\alpha^5+1, \\ F_8(\alpha) =(\alpha^4+\alpha+1)(\alpha^{20}+\alpha^3+1) \\ \quad =\alpha^{24}+\alpha^{21}+\alpha^{20}+\alpha^7+\alpha^3+\alpha+1, \\ F_9(\alpha) =(\alpha^4+\alpha+1)(\alpha^{28}+\alpha^3+1) \\ \quad =\alpha^{32}+\alpha^{29}+\alpha^{28}+\alpha^7+\alpha^3+\alpha+1. \end{cases} \tag{16}$$

Fig. 3 demonstrates the types of bits of our PSIS for the 1-st progressive level when using $f_1(\alpha)$ and $f_2(\alpha)$ for a secret frame having $m = 2$, 3 and 4 pixels. The 9 recovered main polynomials $F_1(\alpha)$, $\cdots$, $F_9(\alpha)$ are illustrated in Figs. 3. Moreover, Figs. 3(a), (b), and (c) show the reconstructed main polynomials correlated to a secret frame having $m = 2$, 3 and 4 pixels using $f_1(\alpha) = \alpha^3 + \alpha + 1$, and Figs. 3(d), (e) and (f) depict the recovered main polynomials corresponding to a secret frame containing $m = 2$, 3 and 4 pixels using $f_1(\alpha) = \alpha^4 + \alpha + 1$. It should be noted that, when $e(\alpha)$ is reduced modulo $f_2(\alpha)$, the result does not simply yield $e_0(\alpha)$. Specifically, when recovering $e(\alpha)$ over $(\bmod\ f_2(\alpha))$, the term $e_1(\alpha)\times\alpha^{8m-\deg(f_1(\alpha))}$ is reduced modulo $f_2(\alpha)$, causing some coefficients of $e_1(\alpha)$ to interfere with those of $e_0(\alpha)$. As a result, the recovered polynomial contains three categories of coefficients: (i) correctly decoded coefficients, which are unaffected by the interference; (ii) possibly erroneously decoded coefficients, which might be altered by the superimposed terms from $e_1(\alpha)$; and (iii) non-decoded coefficients, which correspond to the information lost due to the degree reduction. This categorization is illustrated in detail in Fig. 3. The 2-level progressive recovery property based on $F_1(\alpha)$ in Eq. (16) is examined theoretically in Example 2.

Example 2. *Consider the proposed PSIS with* 2 *progressive levels using* $F_1(\alpha)$ *in Eq. (16). A secret frame having* 2 *pixels is processed each time. When recovering the* 1*-st level secret,* $GF(f_2(\alpha))$ *(i.e.,* $GF(\alpha^{13}+\alpha^4+\alpha^3+\alpha+1)$*) is applied. Eq. (17) describes three categories of bits in the decoded* 2 *pixels when the* 1*-st level is reconstructed. A diagrammatical representation for Eq. (17) is illustrated in Fig. 3(a).*

$$\begin{cases} e(\alpha) =e_0(\alpha)+e_1(\alpha)\times\alpha^{13} \\ \qquad \textit{over } GF(\alpha^{13}+\alpha^4+\alpha^3+\alpha+1) \\ \quad =e_0(\alpha)+e_1(\alpha)\times(\alpha^4+\alpha^3+\alpha+1) \\ \quad =e_{0,0}+e_{0,1}\alpha+\cdots+e_{0,12}\alpha^{12} \\ \qquad +(e_{1,0}+e_{1,1}\alpha+e_{1,2}\alpha^2)(\alpha^4+\alpha^3+\alpha+1) \\ \Rightarrow \textit{Three types of bits in a 2-pixel frame } e(\alpha)\textit{:} \\ \textit{(i) correctly decoded bits:} \overline{e}_7\textit{(1-st pixel)}, \overline{e}_8 \sim \overline{e}_{12}\textit{(2-nd pixel)}; \\ \textit{(ii) possibly erroneously decoded bits: } \overline{e}_0 \sim \overline{e}_6\textit{(1-st pixel)}; \\ \textit{(iii) non-decoded bits: } \overline{e}_{13} \sim \overline{e}_{15}\textit{(2-nd pixel)}. \\ \textit{(note: 2-pixel frame structure } e(\alpha) = \\ \overbrace{\overline{e}_0+\overline{e}_1\alpha+\cdots+\overline{e}_7\alpha^7}^{\textit{1-st pixel with LSB } \overline{e}_0 \textit{ and MSB } \overline{e}_7} + \overbrace{\overline{e}_8\alpha^8+\overline{e}_9\alpha^9+\cdots+\overline{e}_{15}\alpha^{15}}^{\textit{2-nd pixel with LSB } \overline{e}_8 \textit{ and MSB } \overline{e}_{15}}) \\ \Rightarrow \textit{the 1-st level progressive recovery process:} \\ \textit{(i) 1-st pixel: keep MSB } \overline{e}_7\textit{, use "0" for other bits,} \\ \textit{(ii) 2-nd pixel: keep } \overline{e}_8 \sim \overline{e}_{12}\textit{, use "0" for other bits.} \end{cases} \tag{17}$$

In summary, 6 *secret bits (i.e.,* $\overline{e}_7 \sim \overline{e}_{12}$*) are decoded in the* 1*-st level and consequently a draft secret image is achieved. After that, we may decode* $e(\alpha)$ *over* $GF(f_1(\alpha))$ *(i.e.,* $GF(\alpha^3+\alpha+1)$*) and then use CRT for polynomial rings to obtain the original* $e(\alpha)$ *in the* 2*-nd level recovery. In this case, all secret bits are reconstructed and the secret image is perfectly revealed. According to the foregoing theoretical analysis, progressive recovery is provided.*

Consider another case using the same $f_1(\alpha) = \alpha^3 + \alpha + 1$ to deal with 3 pixels. $f_2(\alpha)$ is therefore $\alpha^{21} + \alpha^2 + 1$. By the same argument derived in Eq. (17), three types of bits in a 3-pixel recovered frame of the 1-st level can refer to Fig. 3(b) using $F_2(\alpha)$. The correctly decoded bits of a 3-pixel frame are ($\overline{e}_5$, $\overline{e}_6$, $\overline{e}_7$) of the 1-st pixel, all bits of the 2-nd pixel, and ($\overline{e}_{16} \sim \overline{e}_{20}$) of the 3-rd pixel. The possibly erroneously decoded bits are ($\overline{e}_0 \sim \overline{e}_4$) of the 1-st pixel. And ($\overline{e}_{21} \sim \overline{e}_{23}$) of the 3-rd pixel are the non-decoded bits. Totally 16 secret bits (i.e., $\overline{e}_5 \sim \overline{e}_{20}$) are disclosed from the 1-st level recovery. The 1-st level of our progressive recovery process using $F_2(\alpha)$ is defined as follows: (i) 1-st pixel: keep 3 MSBs ($\overline{e}_5$, $\overline{e}_6$, $\overline{e}_7$) and use "0" for other bits, (ii) 2-nd pixel: keep all bits, (iii) 3-rd pixel: keep ($\overline{e}_{16} \sim \overline{e}_{20}$) and use "0" for other bits.

When dealing with 4 pixels by $F_3(\alpha)$, we obtain 24 secret bits (i.e., $\overline{e}_5 \sim \overline{e}_{28}$) that are correctly decoded from the 1-st secret level. Fig. 3(c-1) further illustrates the three types

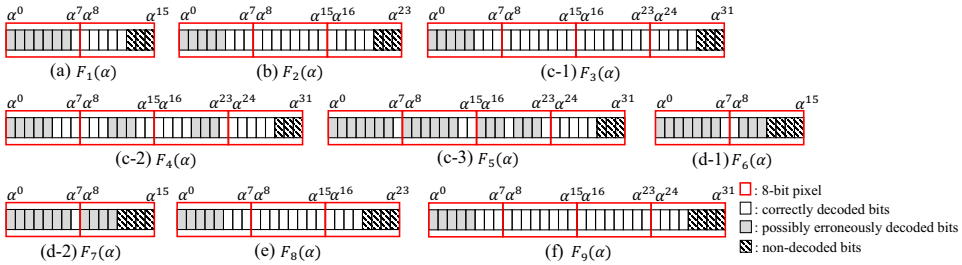

Fig. 3. Bits of the recovered $e(\alpha)$ by the proposed scheme for the 1-st secret level using $F_1(\alpha) \sim F_9(\alpha)$ to deal with: (a) 2 pixels, (b) 3 pixels, (c) 4 pixels, (d) 2 pixels, (e) 3 pixels, (f) 4 pixels.

of bits of the four recovered pixels, on which we can design the 1-st level of progressive recovery: (i) 1-st pixel: keep 3 MSBs ($\overline{e}_5$, $\overline{e}_6$, $\overline{e}_7$) and use "0" for other bits, (ii) 2-nd and 3-rd pixels: keep all bits, (iii) 4-th pixel: keep ($\overline{e}_{24} \sim \overline{e}_{28}$) and use "0" for other bits. There are 2 correct pixels among 4 pixels in the 1-st level recovery when using $F_3(\alpha)$. We can also select $F_4(\alpha)$ (see Fig. 3(c-2)) and $F_5(\alpha)$ (see Fig. 3(c-3)) to implement the two-stage progressive recovery. Figs. 3(d)-(f) further depict the decoded results for 2, 3 and 4 pixels using $f_2(\alpha) = \alpha^4 + \alpha + 1$. The 1-st level recovery processes can be designed by the same argument. According to the theoretical verification, all the 9 main polynomials are capable of realizing a method with 2-level progressive recovery.

In addition, we have $\deg(f_2(\alpha))$ bits decoded from the 1-st level. When preparing a recovered secret frame having $8m$ bits, additional $(8m - \deg(f_2(\alpha)))$ non-decoded bits should be provided. Furthermore, for those decoded $\deg(f_2(\alpha))$ bits, some would be correctly decoded while the others might be erroneously recovered. It depends on the two chosen irreducible polynomials. Usually, a default value (e.g., "0") is assigned to these possibly erroneously decoded bits and non-decoded bits.

IV. Enhancements on The Proposed Approach

In this section, our method is extended to offer t-level progressive recovery. Meanwhile, bit rearrangement is presented in order to enhance the visual performance.

A. PSIS with More Progressive Levels

*1) **Shadow Construction**:* To offer t-level recovery, t multiplicative primitive irreducible polynomials $f_1(\alpha), \cdots, f_t(\alpha)$ should be carefully selected to meet the following progressive requirements: (i) Partial secret bits are correctly decrypted when one irreducible polynomial (e.g., $f_t(\alpha)$) is used; (ii) More secret bits are decoded by l irreducible polynomials when compared with the case using $(l-1)$ irreducible polynomials where $2 \le l \le t$; (iii) All secret bits are reconstructed with the usage of t irreducible polynomials.

Without loss of generality, let $\deg(f_1(\alpha)) \le \deg(f_2(\alpha)) \le \cdots \le \deg(f_t(\alpha))$. The degrees of $f_1(\alpha), \cdots, f_t(\alpha)$ must satisfy $\sum_{i=1}^{t} \deg(f_i(\alpha)) = 8m$ so that a secret frame having m 8-bit pixels can be processed each time. Then, a main polynomial $F(\alpha)$ is constructed as $F(\alpha) = \prod_{i=1}^{t} f_i(\alpha)$. Further, a $(k-1)$-degree polynomial $g(x)$ over $\pmod{F(\alpha)}$, which is responsible for creating shadows for the (k,n) threshold, is established by $g(x) = \sum_{j=0}^{k-1} a_j x^j \pmod{F(\alpha)}$. Remembering that, k secret frames, each having m pixels, should be transformed to k secret polynomials which are considered as the coefficients of $g(x)$. By fetching n different non-zero values $x_1, \cdots, x_n$ from $GF(f_1(\alpha))$ and using them as IDs, n shadow polynomials $g(x_i)$, $1 \le i \le n$ are constructed. These shadow polynomials are converted back to pixel values for producing n shadow images.

*2) **Secret Reconstruction**:* Similar to the 2-level approach, the scheme with t levels also provides two decoding options. In direct decoding mode, LI under $\pmod{F(\alpha)}$ is utilized straightly to restore the polynomial $g(x) \pmod{F(\alpha)}$. By collecting all coefficients of $g(x) \pmod{F(\alpha)}$, a distortion-free secret image is ensured. In addition, by combining LI with polynomial ring CRT, progressive recovery is available. Any k participants can provide the modified shadows modulo $f_1(\alpha), \cdots, f_t(\alpha)$ for progressive decoding. Specifically, let $g(x_1) \pmod{f_{t+1-l}(\alpha)}, \cdots, g(x_k) \pmod{f_{t+1-l}(\alpha)}$ be the modified shadow pixels where $1 \le l \le t$. One can adopt LI to decode $g_l(x) = \sum_{i=1}^{k} (g_l(x_i) \prod_{j \in [1,k], j \ne i} \frac{x - x_j}{x_i - x_j}) \pmod{f_{t+1-l}(\alpha)}$ and employ CRT for polynomial rings to reconstruct $g^{(l)}(x) = CRT(g_1(x), \cdots, g_l(x))$. By collecting the k coefficients of $g^{(l)}(x)$, the l-th $(1 \le l \le t)$ secret level is disclosed. When $l = t$, the secret image is losslessly decrypted.

$$\begin{cases} F_{10}(\alpha) = f_1(\alpha) \times f_2(\alpha) \times f_3(\alpha), \\ F_{11}(\alpha) = f_1(\alpha) \times f_4(\alpha) \times f_5(\alpha), \\ F_{12}(\alpha) = f_1(\alpha) \times f_6(\alpha) \times f_7(\alpha), \\ F_{13}(\alpha) = f_2(\alpha) \times f_8(\alpha) \times f_9(\alpha), \\ F_{14}(\alpha) = f_{10}(\alpha) \times f_{10}(\alpha) \times f_{11}(\alpha) \times f_{12}(\alpha), \\ F_{15}(\alpha) = f_{10}(\alpha) \times f_{10}(\alpha) \times f_6(\alpha) \times f_{13}(\alpha), \\ F_{16}(\alpha) = f_{10}(\alpha) \times f_{10}(\alpha) \times f_8(\alpha) \times f_9(\alpha). \end{cases} \tag{18}$$

*3) **Progressive Recovery Property**:* The progressive recovery property depends on the t chosen polynomials $f_1(\alpha), \cdots, f_t(\alpha)$. Except for $f_t(\alpha)$, the degrees of $f_1(\alpha), \cdots, f_{t-1}(\alpha)$ should be small. When dealing with m secret pixels each time, the degree of $f_t(\alpha)$ is figured out as $(8m - \sum_{i=1}^{t-1} \deg(f_i(\alpha)))$. In this case, we can select suitable $f_1(\alpha), \cdots, f_t(\alpha)$ such that some bits in the revealed pixels belonging to 1-st level recovery may not be changed. Moreover, the number of secret bits of the l-th level recovery should be larger than that of the $(l-1)$-th level where $2 \le l \le t$, and the final level recovery reveals the original image. Herein, $F_{10}(\alpha) \sim F_{13}(\alpha)$

and $F_{14}(\alpha) \sim F_{16}(\alpha)$ for offering 3 and 4 progressive levels, respectively, are depicted in Eq. (18). The involved irreducible polynomials are provided in Eq. (19).

$$\begin{cases} f_1(\alpha) = \alpha^3 + \alpha + 1,\ f_2(\alpha) = \alpha^4 + \alpha + 1, \\ f_3(\alpha) = \alpha^{25} + \alpha^{11} + \alpha^9 + \alpha^8 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha^2 + 1, \\ f_4(\alpha) = \alpha^6 + \alpha + 1,\ f_5(\alpha) = \alpha^{23} + \alpha^5 + 1, \\ f_6(\alpha) = \alpha^7 + \alpha + 1,\ f_7(\alpha) = \alpha^{22} + \alpha + 1, \\ f_8(\alpha) = \alpha^7 + \alpha^6 + \alpha^4 + \alpha^2 + 1,\ f_9(\alpha) = \alpha^{21} + \alpha^2 + 1, \\ f_{10}(\alpha) = \alpha^2 + \alpha + 1,\ f_{11}(\alpha) = \alpha^5 + \alpha^2 + 1, \\ f_{12}(\alpha) = \alpha^{23} + \alpha^{22} + \alpha^7 + \alpha^5 + 1, \\ f_{13}(\alpha) = \alpha^{21} + \alpha^8 + \alpha^7 + \alpha^4 + \alpha^3 + \alpha^2 + 1. \end{cases} \tag{19}$$

Example 3 theoretically verifies the progressive recovery property of the proposed scheme with 3 levels where $F_{10}(\alpha)$ in Eq. (18) is employed. Based on the same argument, one can theoretically examine the scheme using various irreducible polynomials.

Example 3. *Consider the proposed technique with 3-level recovery. We can select 3 multiplicative primitive irreducible polynomials $f_1(\alpha)$, $f_2(\alpha)$, and $f_3(\alpha)$ in Eq. (19) to offer 3 recovery levels. The main polynomial is computed as $F_{10}(\alpha) = f_1(\alpha) \times f_2(\alpha) \times f_3(\alpha) = \alpha^{32} + \alpha^{30} + \alpha^{28} + \alpha^{27} + \alpha^{25} + \alpha^{18} + \alpha^{15} + \alpha^{13} + \alpha^{12} + \alpha^{11} + \alpha^8 + \alpha^7 + \alpha^6 + 1$. As $\deg(F_{10}(\alpha)) = 32$, a frame having 4 pixels (e.g., 32 bits) is processed each time.*

$$\begin{cases} e(\alpha) = e_0(\alpha) + e_1(\alpha) \times \alpha^{25} \text{ over } GF(f_3(\alpha)) \\ \Rightarrow e(\alpha) \text{ over } GF(\alpha^{25} + \alpha^{11} + \alpha^9 + \alpha^8 + \alpha^6 + \alpha^4 + \alpha^3 \\ +\alpha^2 + 1) \\ = e_0(\alpha) + e_1(\alpha) \times (\alpha^{11} + \alpha^9 + \alpha^8 + \alpha^6 + \alpha^4 + \alpha^3 \\ +\alpha^2 + 1) \\ = e_{0,0} + e_{0,1}\alpha + \cdots + e_{0,24}\alpha^{24} + (e_{1,0} + e_{1,1}\alpha + \cdots \\ +e_{1,6}\alpha^6) \times (\alpha^{11} + \alpha^9 + \alpha^8 + \alpha^6 + \alpha^4 + \alpha^3 + \alpha^2 + 1) \\ \Rightarrow \textit{Three types of bits in a 4-pixel frame } e(\alpha): \\ \textit{(i) correctly decoded bits:} \\ \overline{e}_{18} \sim \overline{e}_{23}(\textit{3-rd pixel}), \overline{e}_{24}(\textit{4-th pixel}); \\ \textit{(ii) possibly erroneously decoded bits:} \\ \overline{e}_0 \sim \overline{e}_{15}(\textit{1-st, 2-nd pixels}), \overline{e}_{16}, \overline{e}_{17}(\textit{3-rd pixel}); \\ \textit{(iii) non-decoded bits: } \overline{e}_{25} \sim \overline{e}_{31}(\textit{4-th pixel}). \\ (\textit{note: 4-pixel frame structure} \\ e(\alpha) = \overbrace{\overline{e}_0 + \cdots + \overline{e}_7\alpha^7}^{\textit{1-st pixel}} + \overbrace{\overline{e}_8\alpha^8 + \cdots + \overline{e}_{15}\alpha^{15}}^{\textit{2-nd pixel}} \\ \overbrace{\overline{e}_{16}\alpha^{16} + \cdots + \overline{e}_{23}\alpha^{23}}^{\textit{3-rd pixel}} + \overbrace{\overline{e}_{24}\alpha^{24} + \cdots + \overline{e}_{31}\alpha^{31}}^{\textit{4-th pixel}}) \end{cases} \tag{20}$$

For the 1-st level recovery, one can use $GF(f_3(\alpha)) = GF(\alpha^{25}+\alpha^{11}+\alpha^9+\alpha^8+\alpha^6+\alpha^4+\alpha^3+\alpha^2+1)$ for decoding, as given in Eq. (20). 7 bits in the 3-rd and 4-th pixels are correctly recovered. For the 2-nd level decoding, the original shadows modulo $f_2(\alpha)$ and $f_3(\alpha)$ are employed. The recovery is equivalent to recover the secret over $(\mathrm{mod}\ f_2(\alpha) \times f_3(\alpha))$, as depicted in Eq. (21). Totally 10 bits in the 2-nd, 3-rd and 4-th pixels are correctly disclosed. When compared with the 1-st level, the 2-level reveals 3 extra bits (i.e., $\overline{e}_8 \sim \overline{e}_{10}$) in the 2-nd pixel. For the 3-rd recovery level, all secret bits are losslessly reconstructed. Progressive recovery property is guaranteed.

$$\begin{cases} e(\alpha) = e_0(\alpha) + e_1(\alpha) \times \alpha^{29} \quad (\mathrm{mod}\ f_2(\alpha) \times f_3(\alpha)) \\ \Rightarrow e(\alpha) \quad (\mathrm{mod}\ f_2(\alpha) \times f_3(\alpha)) \\ = e_0(\alpha) + e_1(\alpha) \times (\alpha^{26} + \alpha^{25} + \alpha^{15} + \alpha^{13} + \alpha^{11} + \alpha^5 \\ +\alpha^4 + \alpha^2 + \alpha + 1) \\ = e_{0,0} + e_{0,1}\alpha + \cdots + e_{0,28}\alpha^{28} + (e_{1,0} + e_{1,1}\alpha + e_{1,1}\alpha^2) \\ \times(\alpha^{26} + \alpha^{25} + \alpha^{15} + \alpha^{13} + \alpha^{11} + \alpha^5 + \alpha^4 + \alpha^2 + \alpha + 1) \\ \Rightarrow \textit{Three types of bits in a 4-pixel frame } e(\alpha): \\ \textit{(i) correctly decoded bits:} \\ \overline{e}_8 \sim \overline{e}_{10}(\textit{2-nd pixel}), \overline{e}_{18} \sim \overline{e}_{23}(\textit{3-rd pixel}), \overline{e}_{24}(\textit{4-th pixel}); \\ \textit{(ii) possibly erroneously decoded bits:} \\ \overline{e}_0 \sim \overline{e}_7(\textit{1-st pixel}), \overline{e}_{11} \sim \overline{e}_{15}(\textit{2-nd pixel}), \\ \overline{e}_{16}, \overline{e}_{17}(\textit{3-rd pixel}), \overline{e}_{25} \sim \overline{e}_{28}(\textit{4-th pixel}); \\ \textit{(iii) non-decoded bits: } \overline{e}_{29} \sim \overline{e}_{31}(\textit{4-th pixel}). \end{cases} \tag{21}$$

B. Visual Enhancement using Bit Rearrangement

In the proposed scheme, bits from a secret frame e are sequentially fetched to form the coefficients of polynomial $e(\alpha)$. When performing progressive decoding, once the t irreducible polynomials $f_1(\alpha)$, $\cdots$, $f_t(\alpha)$ are determined, the positions of correctly recovered coefficients of $e(\alpha)$ are fixed as well. The performance of progressive recovery might become unsatisfactory because some desired secret bits (i.e., coefficients of $e(\alpha)$) are not successfully decrypted.

Based on this observation, a bit rearrangement technique is devised to improve the visual performance. To ease the description, a secret frame e having m pixels is rewritten as

$$\begin{aligned} e =& (s_1^{(0)} \cdots s_1^{(7)} \cdots s_m^{(0)} \cdots s_m^{(7)})_2 \\ =& (s^{(0)} \cdots s^{(7)} \cdots s^{(8m-8)} \cdots s^{(8m-1)})_2. \end{aligned} \tag{22}$$

The rearrangement procedure inputs the $8m$-bit secret frame e and a mapping key $K_m = \{j_0, j_1, \cdots, j_{8m-1}\}$ and outputs a rearranged frame e^A whose $8m$ bits are organized according to the mapping key K_m, as represented by $e^A = \sigma(e, K_m)$. More concretely, the rearranged secret frame $e^A = (s^{(j_0)} s^{(j_1)} \cdots s^{(j_{8m-1})})_2$ is achieved by

$$\begin{pmatrix} s^{(0)} & \cdots & s^{(8m-1)} \\ 0 & \cdots & 8m-1 \end{pmatrix} \Longrightarrow \begin{pmatrix} s^{(j_0)} & \cdots & s^{(j_{8m-1})} \\ j_0 & \cdots & j_{8m-1} \end{pmatrix}. \tag{23}$$

The rearranged secret frame e^A is then converted to its polynomial form for shadow generation. When decoding the secret, the reconstructed secret frame should be rearranged again using the inverse mapping.

The core idea to enhance visual performance is to allocate some desired secret bits to positions that remain unaffected during progressive decoding. Through the skillful design of mapping key K_m, superior performance can be ensured.

Example 4 is illustrated to capture the concept of bit reorganization for visual improvement.

In the proposed t-level technique, suitable irreducible polynomials are selected to enable progressive recovery. Specifically, polynomials that facilitate the correct decoding of some MSBs during low-level reconstruction should be chosen. However, this selection criterion changes with the introduction of bit rearrangement. When bit rearrangement is employed, the MSBs can be appropriately arranged, allowing for the utilization of more irreducible polynomials. This flexibility greatly enhances the progressive recovery.

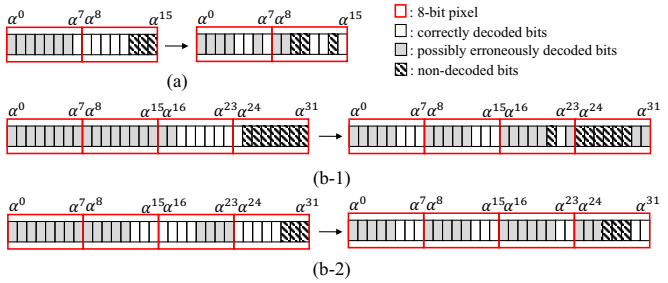

Fig. 4. Bits of the recovered $e(\alpha)$ by the proposed scheme (left: original, right: using bit arrangement). (a) The 1-st level of a scheme with 2 levels, (b) the 1-st and 2-nd levels of a scheme with 3 levels.

Example 4. *Consider the usage of bit rearrangement to improve visual performance. Suppose* $F_1(\alpha)$ *is employed (see Fig. 3(a)) to provide* 2*-level progressive recovery, the* 6 *bits in positions* $7 \sim 12$ *(starting from position* 0*) are correctly decoded for the* 1*-st level. Among them, only the MSB of the* 1*-st secret pixel is retrieved. Consequently, the recovered clarity of the* 1*-st level might be unsatisfactory. We hereby employ the following mapping key* $K_m = \{0, 1, 2, 3, 6, 8, 9, \underline{4}, \underline{5}, \underline{7}, \underline{12}, \underline{13}, \underline{15}, 10, 11, 14\}$ *to rearrange the secret bits, where the underlined element denotes the successful decoding in the* 1*-st level. According to* K_m*, the* 1*-st, 3-rd, and 4-th MSBs of the* 1*-st and* 2*-nd secret pixels can be correctly reconstructed in the* 1*-st level, as depicted in Fig. 4(a). Since more MSBs are decoded, the* 1*-st level clarity is greatly enhanced when compared with the result without bit rearrangement.*

Moreover, for the proposed scheme with 3 *levels given in Example 3,* 6 *MSBs of the* 3*-rd pixel and* 1 *LSB of the* 4*-th pixel are decrypted in the* 1*-st level. For the* 2*-nd level, another* 3 *LSBs of the* 1*-st pixels are recovered. Herein, one can use the following mapping key*

$$\begin{cases} K_m = \{0, 1, 2, 3, 4, 8, 9, 10, \overline{23}, \overline{30}, \overline{31}, 11, 12, 16, 17, 18, \\ 19, 20, \underline{5}, \underline{6}, \underline{7}, \underline{13}, \underline{14}, \underline{15}, \underline{22}, 21, 24, 25, 26, 27, 28, 29\} \end{cases} \tag{24}$$

for visual enhancement, where the underlined and overlined elements represent the correct decryption in the 1*-st and* 2*-nd levels, respectively. In this case, the* 1*-st level discloses* 3 *MSBs of the* 1*-st pixel,* 3 *MSBs of the* 2*-nd pixel, and the* 2*-nd MSB of the* 3*-rd pixel, as given in Fig. 4(b-1). For the* 2*-nd level, the decoded result reveals additional* 3 *bits, which are the MSB of the* 3*-rd pixel and* 2 *MSBs of the* 4*-th pixel, as shown in Fig. 4(b-2). The recovered clarity is improved.*

V. Experimental Result and Discussion

Experimental results of the proposed method are demonstrated and evaluated, as well as the visual performance of progressive recovery. Discussions and comparisons are provided to highlight the benefits of our technique. In terms of implementation, all experiments are performed using Python 3.9 in the following environment: an Intel(R) Core(TM) $i7-12650H$ processor, 16 GB RAM, and Windows 11.

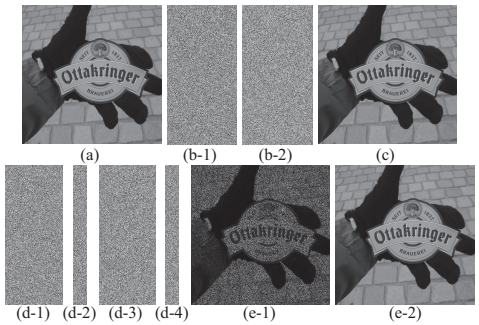

Fig. 5. Our $(2, 2)$ scheme with 2 recovery levels. (a) Secret image, (b) 2 generated shadows, (c) decoded result by direct recovery, (d) 4 modified shadows, and (e) decoded results with 2 levels by progressive recovery.

A. Perceptual Result

Our $(2, 2)$ scheme with 2 recovery levels is illustrated in Fig. 5, where $F_5(\alpha)$ is employed. Fig. 5(a) shows the secret image and Fig. 5(b) illustrates the 2 generated shadows. The recovered secret image is depicted in Fig. 5(c) when direct recovery is adopted. On the other hand, Fig. 5(d) demonstrates the modified shadows and the decoded results of the 1-st and 2-nd levels are given in Fig. 5(e). Our $(3, 3)$ scheme with 3-level recovery is depicted in Fig. 6, where $F_{10}(\alpha)$ with bit rearrangement is applied. The secret image and 3 constructed shadows are demonstrated in Figs. 6(a) and (b), respectively. Fig. 6(c) illustrates the reconstructed secret using direct decoding. Figs. 6(d) and (e) exhibit the modified shadows and corresponding 3 recovered images belonging to the 3 secret levels.

B. Visual Performance of Progressive Recovery

1) ***Quantitative Measurement****:* Visual performance of the proposed method, including the visual quality of the decoded image and progressive recovery property, is evaluated.

Two quantitative measurements, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), are adopted to evaluate the reconstructed image quality. Given two images X and Y, each with $W \times H$ pixels, PSNR is calculated by $PSNR = 10\log_{10}(\frac{255^2}{MSE})$ where MSE is the mean square error between X and Y, as computed by $MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(X_{i,j} - Y_{i,j})^2$. SSIM is computed as $SSIM = [l(X,Y)]^a[c(X,Y)]^\beta[s(X,Y)]^\gamma$ where $l(X,Y) = \frac{2\mu_X\mu_Y + C_1}{\mu_X^2 + \mu_Y^2 + C_1}$, $c(X,Y) = \frac{2\sigma_X\sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2}$, $s(X,Y) = \frac{2\sigma_{XY} + C_3}{\sigma_X\sigma_Y + C_3}$. In the above equation, (μ_X, μ_Y), (σ_X, σ_Y), σ_{XY} are the local means, standard deviations, and cross-covariance for images X and Y, respectively, and $C_1 = (K_1L)^2$, $C_2 = (K_2L)^2$,

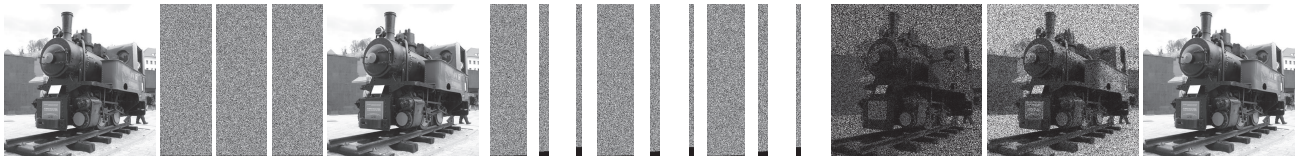

Fig. 6. Our $(3,3)$ scheme with 3 recovery levels. (a) Secret image, (b) 3 generated shadows, (c) decoded result by direct recovery, (d) 9 modified shadows, (e) decoded results with 3 levels by progressive recovery.

$C_3 = C_2/2$, $a = \beta = \gamma = 1$. Usually, $K_1 = 0.01$, $K_2 = 0.03$, and $L = 255$ for 8-bit gray-level images.

A PSIS with t progressive levels is said to achieve uniform progressive recovery if, for the l-th recovery level $(1 \le l \le t)$, the number of correctly decoded pixels equals $(\frac{l}{t} \times m)$ for a secret frame of m pixels. This definition provides a quantitative criterion: the correctly decoded information should grow linearly with the recovery level. A reference-based measurement serves to assess the progressive recovery characteristic. The l-th level reference image, which reveals exactly an l/t fraction of the original secret pixels, serves as the benchmark for evaluation. As a result of this, the visual quality of l-th progressive decoded image (e.g., PSNR and SSIM) of a scheme should be as close as possible to that of the l-th reference image. We further demonstrate an alternative method for generating the l-th reference image. Essentially, all the $W \times H$ pixels of the l-th reference image are divided into $(W \times H)/t$ non-overlapping groups. For each group having t pixels, l pixels are directly copied from the secret image at the same locations and the remaining $(t-l)$ pixels are assigned a default value (e.g., 0).

$$\begin{cases} K_m^{(1)} = \{0,1,8,9,10,11,12,\underline{2},\underline{3},\underline{4},\underline{5},\underline{6},\underline{7},13,14,15\}, \\ K_m^{(2)} = \{0,1,2,3,4,\underline{7},\underline{8},\underline{9},\underline{10},\underline{11},\underline{12},\underline{13},\underline{14},\underline{15},\underline{17}, \\ \quad \underline{18},\underline{19},\underline{20},\underline{21},\underline{22},\underline{23},5,6,16\}, \\ K_m^{(3)} = \{0,1,2,3,4,\underline{7},\underline{9},\underline{10},\underline{11},\underline{12},\underline{13},\underline{14},\underline{15},\underline{16},\underline{17}, \\ \quad \underline{18},\underline{19},\underline{20},\underline{21},\underline{22},\underline{23},\underline{24},\underline{25},\underline{26}, \\ \quad \underline{27},\underline{28},\underline{29},\underline{30},\underline{31},5,6,8\}, \\ K_m^{(4)} = \{0,1,2,3,4,\underline{16},\underline{17},\underline{18},\underline{19},\underline{20},\underline{21},5,6,8,\underline{22}, \\ \quad \underline{23},\underline{24},\underline{25},\underline{26},\underline{27},9,10,11,\underline{28},\underline{29},\underline{30},\underline{31},\underline{7},\underline{15}, \\ \quad 12,13,14\}, \\ K_m^{(5)} = \{0,1,2,3,6,8,9,\underline{4},10,11,14,16,17,18,\underline{5},\underline{7}, \\ \quad 19,20,22,\underline{12},24,25,26,\underline{13},\underline{15}, \\ \quad \underline{21},\underline{23},\underline{29},\underline{31},27,28,30\} \\ K_m^{(6)} = \{0,1,2,3,4,5,6,\underline{7},\underline{15},8,9,10,11,12,13,14\}, \\ K_m^{(7)} = \{0,1,2,3,4,5,6,\underline{7},8,9,10,11,12,13,14,15\}, \\ K_m^{(8)} = \{0,1,8,9,10,11,12,\underline{2},\underline{3},\underline{4},\underline{5},\underline{6},\underline{7},\underline{15},\underline{18},\underline{19}, \\ \quad \underline{20},\underline{21},\underline{22},\underline{23},13,14,16,17\}, \\ K_m^{(9)} = \{8,9,10,11,12,13,14,\underline{0},\underline{1},\underline{2},\underline{3},\underline{4},\underline{5},\underline{6},\underline{7},\underline{15}, \\ \quad \underline{17},\underline{18},\underline{19},\underline{20},\underline{21},\underline{22},\underline{23},\underline{27},\underline{28},\underline{29},\underline{30},\underline{31}, \\ \quad 16,24,25,26\}. \end{cases} \tag{25}$$

*2) **Performance Evaluation**:* For evaluation, 49 test images are randomly selected from the USC-SIPI and CVG-UGR image datasets to serve as secret images for assessing visual performance. The average values of PSNR and SSIM of the reference images (RI) and decoded images by our scheme are listed in Tables III-V where our approach with 2, 3, and 4 progressive levels is implemented.

$$\begin{cases} K_m^{(10)} = \{0,1,2,3,4,8,10,11,\overline{15},\overline{29},\overline{5},20,12,13,14, \\ \quad 16,17,18,\underline{30},\underline{6},\underline{7},\underline{21},\underline{22},\underline{23},\underline{31},25,26,27,28, \\ \quad 9,19,24\}, \\ K_m^{(11)} = \{18,8,9,10,\overline{19},11,12,13,\overline{20},\overline{21},\overline{23},14,15, \\ \quad 16,\underline{0},\underline{1},\underline{2},\underline{3},\underline{4},\underline{5},\underline{6},\underline{7},\underline{22},17,24,25,26,\overline{29},\overline{31}, \\ \quad 27,28,30\}, \\ K_m^{(12)} = \{0,1,2,3,8,\overline{12},\overline{14},9,10,11,16,\underline{4},\underline{5},\underline{6},\underline{7},\underline{13}, \\ \quad \underline{15},\underline{20},\underline{21},\underline{22},\underline{23},\underline{31},17,18,19,24,\overline{28},\overline{29},\overline{30},25, \\ \quad 26,27\}, \\ K_m^{(13)} = \{0,1,2,3,\overline{6},\overline{14},\overline{30},4,8,9,10,11,12,\underline{5},\underline{7},\underline{13}, \\ \quad \underline{15},\underline{22},\underline{23},\underline{29},\underline{31},16,17,18,19,20,21,24,25, \\ \quad 26,27,28\}. \end{cases} \tag{26}$$

$$\begin{cases} K_m^{(14)} = \{0,1,2,3,4,8,\overline{14},\overline{22},\overline{30},9,10,\hat{5},\hat{13},11,12, \\ \quad 16,\underline{6},\underline{7},\underline{15},\underline{23},\underline{31},17,18,19,20,24,25,\hat{21}, \\ \quad \hat{29},26,27,28\}, \\ K_m^{(15)} = \{0,1,2,3,4,5,8,9,10,\overline{6},11,12,\hat{14},13,16, \\ \quad 17,\hat{30},18,19,\underline{7},\underline{23},20,21,\hat{22},24,25,\overline{15},\overline{31}, \\ \quad 26,27,28,29\}, \\ K_m^{(16)} = \{0,8,9,10,\overline{12},\overline{15},\overline{22},11,13,14,16,17,18, \\ \quad \underline{1},\underline{2},\underline{3},\underline{4},\underline{5},\underline{6},\underline{7},\underline{19},20,21,23,24,25,26,27, \\ \quad \hat{30},\hat{31},28,29\}. \end{cases} \tag{27}$$

For the proposed technique with 2 progressive levels, alternative main polynomials $F_1(\alpha) \sim F_9(\alpha)$ given in Eq. (16) are considered. Meantime, the mapping keys $K_m^{(1)} \sim K_m^{(9)}$ correlated to $F_1(\alpha) \sim F_9(\alpha)$, respectively, for bit rearrangement are presented in Eq. (25). As shown in Eq. (18), the main polynomials $F_{10}(\alpha) \sim F_{13}(\alpha)$ (with mapping keys $K_m^{(10)} \sim K_m^{(13)}$ in Eq. (26)) and $F_{14}(\alpha) \sim F_{16}(\alpha)$ (with mapping keys $K_m^{(14)} \sim K_m^{(16)}$ in Eq. (27)) are employed for constituting the scheme with 3 and 4 levels, respectively.

Additionally, the results with and without bit rearrangement (denoted as w BR and w/o BR) are illustrated as well.

According to the results, different main polynomials lead to variations in visual performance. Bit rearrangement with an appropriate mapping key can substantially enhance the performance. For our scheme with 2 levels, $F_1(\alpha)$ combined with bit rearrangement yields improved recovery clarity. Meanwhile, the decoded results are closer to the reference images, indicating that uniform progressive recovery is achieved. The same conclusion also holds when $F_{11}(\alpha)$ and $F_{16}(\alpha)$ with bit rearrangement are utilized for the 3 and 4 levels.

TABLE III
VISUAL PERFORMANCE OF OUR SCHEME FOR 2 PROGRESSIVE LEVELS

Main Polynomial	l-th	PSNR		SSIM	
		w/o BR	w BR	w/o BR	w BR
$F_1(\alpha)$	1	9.27	**10.11**	0.1050	**0.1301**
	2	∞	∞	1	**1**
$F_2(\alpha)$	1	12.87	15.78	0.2461	0.3236
	2	∞	∞	1	1
$F_3(\alpha)$	1	14.11	17.03	0.2866	0.3685
	2	∞	∞	1	1
$F_4(\alpha)$	1	12.11	14.02	0.2109	0.2634
	2	∞	∞	1	1
$F_5(\alpha)$	1	10.78	13.61	0.1489	0.3786
	2	∞	∞	1	1
$F_6(\alpha)$	1	8.54	9.63	0.0490	0.0938
	2	∞	∞	1	1
$F_7(\alpha)$	1	8.51	8.51	0.0459	0.0459
	2	∞	∞	1	1
$F_8(\alpha)$	1	10.64	15.78	0.1417	0.3268
	2	∞	∞	1	1
$F_9(\alpha)$	1	11.89	17.02	0.1832	0.3714
	2	∞	∞	1	1
RI	1	**10.11**		**0.1316**	
	2	∞		**1**	

TABLE IV
VISUAL PERFORMANCE OF OUR SCHEME FOR 3 PROGRESSIVE LEVELS

Main Polynomial	l-th	PSNR		SSIM	
		w/o BR	w BR	w/o BR	w BR
$F_{10}(\alpha)$	1	8.36	9.00	0.0638	0.0855
	2	8.44	12.21	0.0696	0.1904
	3	∞	∞	1	1
$F_{11}(\alpha)$	1	9.22	**9.00**	0.0892	**0.0858**
	2	9.86	**12.03**	0.1263	**0.1917**
	3	∞	∞	1	**1**
$F_{12}(\alpha)$	1	8.99	14.84	0.0889	0.3147
	2	10.57	29.26	0.1691	0.9172
	3	∞	∞	1	1
$F_{13}(\alpha)$	1	8.65	13.70	0.0795	0.3113
	2	9.75	20.45	0.1260	0.6180
	3	∞	∞	1	1
RI	1	**8.86**		**0.0836**	
	2	**11.87**		**0.1871**	
	3	∞		**1**	

C. Discussion and comparison

1) ***Comparison of Perceptual Result****:* Perceptual comparisons of 2, 3, and 4 progressive levels are shown in Figs. 7, 8, and 9, respectively. Herein, state-of-the-art works, including MP-SIS [13], CP-PSIS [12], MCP-PSIS [12], and PSIS-BP [21], are used. For our technique, the main polynomials $F_1(\alpha)$, $F_{11}(\alpha)$, $F_{16}(\alpha)$ with bit rearrangement are utilized. According to the perceptual results, progressive recovery is uneven for MP-SIS [13], CP-PSIS [12], MCP-PSIS [12], and PSIS-BP [21]. Take the 3 levels for example. For MP-SIS, the 1-st and 2-nd recovery levels (i.e., Figs. 8(b-1) and (b-2)) give no visual clue about the secret. For CP-PSIS, it is not easy to perceive the secret information through the 1-st recovery level (i.e., Figs. 8(c-1)). The decoded results of the 1-st and 2-nd levels (i.e., Figs. 8(e-1) and (e-2)) by PSIS-BP are barely the same and the visual quality is unsatisfactory. For MCP-PSIS, the recovery is not progressive since the 1-st level is approximately the same as the 3-rd level. In contrast, the proposed scheme provides uniform progressive recovery and better recovered image quality of the low-levels.

TABLE V
VISUAL PERFORMANCE OF OUR SCHEME FOR 4 PROGRESSIVE LEVELS

Main Polynomial	l-th	PSNR		SSIM	
		w/o BR	w BR	w/o BR	w BR
$F_{14}(\alpha)$	1	7.36	11.88	0.0350	0.2387
	2	8.53	16.85	0.0710	0.5540
	3	9.13	22.93	0.1148	0.7913
	4	∞	∞	1	1
$F_{15}(\alpha)$	1	7.30	8.51	0.0305	0.0459
	2	7.43	11.88	0.0457	0.2387
	3	8.22	16.85	0.0742	0.5540
	4	∞	∞	1	1
$F_{16}(\alpha)$	1	8.65	**8.35**	0.0795	**0.0634**
	2	9.75	**10.21**	0.1260	**0.1281**
	3	10.48	**12.84**	0.1707	**0.2151**
	4	∞	∞	1	**1**
RI	1	**8.35**		**0.0643**	
	2	**10.11**		**0.1316**	
	3	**13.12**		**0.2252**	
	4	∞		**1**	

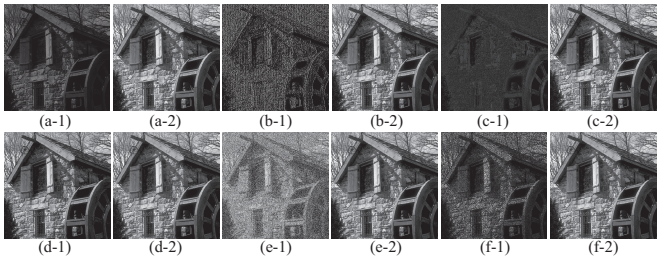

Fig. 7. Visual comparison with 2 progressive recovery levels. (a) Reference images (b) MP-SIS, (c) CP-PSIS, (d) MCP-PSIS, (e) PSIS-BP, and (f) the proposed technique.

TABLE VI
COMPARISON OF AVERAGE PSNR OF DECODED IMAGE

t	l	RI	MP-SIS	CP-PSIS	MCP-PSIS	PSIS-BP	Our
2	1	10.11	7.11	8.77	29.26	10.65	**10.11**
	2	∞	33.45(11)	∞	∞	∞	∞
3	1	8.86	7.44	7.45	22.93	9.35	**9.00**
	2	11.87	7.93	13.54	21.84	11.11	**12.03**
	3	∞	37.85(11)	∞	∞	∞	∞
4	1	8.35	6.30	7.45	11.01	9.69	**8.35**
	2	10.11	6.07	9.04	14.52	10.56	**10.21**
	3	13.12	6.91	15.50	15.26	12.87	**12.84**
	4	∞	38.39(10)	∞	∞	∞	∞

TABLE IX
FEATURE COMPARISON AMONG RELATED SCHEMES

Method	Feature					
	Key Technique	Decoding Operation	Lossless Decryption	Minimum Number for Lossless Recovery	Progressive Recovery	
					Uniform Recovery	Clarity
VSS-MC [25]	LP, BM	XOR-ing	✗	-	✗	Inferior
VSS-EAS [26]	SA, BM	Stacking, XOR-ing	✗	-	✗	Inferior
JPSIS [23]	LI, Masking	PMR	✗	-	✗	Inferior
PSIS-DR [22]	DL, RDHEI	PMR	✓	n	✗	Good
PSIS-BP [21]	BO, LI	Boolean, PMR	✓	n	✗	Inferior
CP-PSIS [12]	IRC, LI	IMR	✓	k	✗	Inferior
MCP-PSIS [12]	IRC, LI	IMR	✓	k	✗	Good
MP-SIS [13]	IRC, LI	IMR	✗	k	✗	Inferior
Our	MPIP, PRC, LI, BR	PMR	✓	k	✓	Good

LP: Linear Programming, BM: Base Matrix, SA: Simulated Annealing, LI: Lagrange Interpolation, DL: Deep Learning, RDHEI: Reversible Data Hiding in Encrypted Images, BO: Boolean Operation, IRC: Integer Ring CRT, PRC: Polynomial Ring CRT, IMR: Integer Modular Arithmetic, PMR: Polynomial Modular Arithmetic, MPIP: Multiplicative Primitive Irreducible Polynomial, BR: Bit Rearrangement

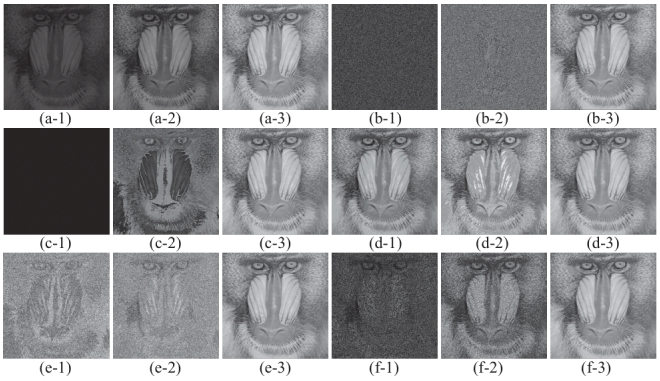

Fig. 8. Visual comparison with 3 progressive recovery levels. (a) Reference images (b) MP-SIS, (c) CP-PSIS, (d) MCP-PSIS, (e) PSIS-BP, and (f) the proposed technique.

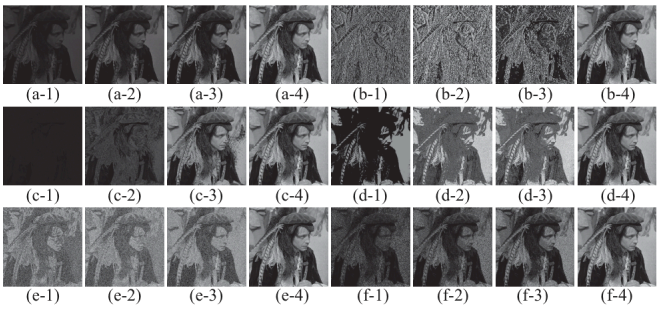

Fig. 9. Visual comparison with 4 progressive recovery levels. (a) Reference images (b) MP-SIS, (c) CP-PSIS, (d) MCP-PSIS, (e) PSIS-BP, and (f) the proposed technique.

TABLE VII
COMPARISON OF AVERAGE SSIM OF DECODED IMAGE

t	l	RI	MP-SIS	CP-PSIS	MCP-PSIS	PSIS-BP	Our
2	1	0.1316	0.0089	0.1838	0.9205	0.0762	**0.1301**
	2	1	0.9757	1	1	1	1
3	1	0.0836	0.0115	0.0570	0.7977	0.0379	**0.0858**
	2	0.1871	0.0124	0.6043	0.6345	0.0901	**0.1917**
	3	1	0.9800	1	1	1	1
4	1	0.0643	0.0300	0.0570	0.2452	0.0427	**0.0634**
	2	0.1316	0.0445	0.2009	0.2293	0.0716	**0.1281**
	3	0.2252	0.0410	0.6942	0.2968	0.1477	**0.2151**
	4	1	0.9800	1	1	1	1

TABLE VIII
COMPARISON OF EXECUTION TIME FOR A SHARING-RECOVERY CYCLE

Total level	MP-SIS	CP-PSIS	MCP-PSIS	PSIS-BP	Our
2	1.63	0.57	0.71	243.08	14.41
3	2.93	0.86	1.07	359.12	22.18
4	2.88	1.21	1.42	389.75	28.34

*2) **Quantitative Evaluation of Progressive Recovery**:* Comparisons of average PSNR and SSIM across various schemes for $t = 2$, 3, and 4 progressive levels are given in Tables VI and VII, where 49 test images randomly selected from USC-SIPI and CVG-UGR image databases are adopted. The average values of PSNR and SSIM of the reference images (RI) are provided. For our scheme, the main polynomials $F_1(\alpha)$, $F_{11}(\alpha)$, $F_{16}(\alpha)$ with bit rearrangement are considered.

The values of PSNR and SSIM of the reference images are used as indicators to evaluate the progressive recovery property. Uniformity is guaranteed as similar values are provided by a PSIS. Herein, we highlight those results which are the nearest to those of the reference images. According to Tables VI and VII, the best values are provided by our scheme, ensuring the uniform progressive recovery. In addition, when compared with MP-SIS [13], CP-PSIS [12], and PSIS-BP [21], our method also achieves higher values of PSNR and SSIM for most of the low-level decoded results. Better image quality of low-level recovered images is provided. Note that, MP-SIS [13] is not able to offer lossless decoding. The PSNR and SSIM cannot be ∞ and 1, respectively. Herein, the average values of those distorted recovered results are calculated and the quantities of lossy decoded images among the 49 test images are also indicated in Table VI (i.e., 11, 11, and 10 for 2, 3, and 4 levels, respectively).

*3) **Insight into Uniform Progressive Recovery**:* **Ideal Uniformity Condition.** To achieve ideal uniform progressive recovery, the polynomials $f_1(\alpha)$, $\cdots$, $f_t(\alpha)$ used in our scheme must be selected such that, at each recovery level l $(1 \leq l \leq t)$, the number of correctly decoded coefficients in the recovered $e(\alpha)$ is exactly $(\frac{l}{t} \times 8m)$ where $8m$ is the total number of coefficients.

However, as demonstrated in the theoretical analysis of

Examples 2-3, the number and positions of correctly decoded coefficients at each level are determined by the nonzero-term structure of the involved irreducible polynomials. Since each $f_i(\alpha)$ must be a primitive irreducible polynomial, its nonzero-term distribution is subject to strict algebraic constraints imposed by the irreducibility and primitivity conditions. We cannot freely design the term structure of $f_i(\alpha)$ to precisely control which coefficients of $e(\alpha)$ are correctly decoded. Consequently, finding a combination of primitive irreducible polynomials that yields exactly $(\frac{l}{t} \times 8m)$ correctly decoded coefficients at every level l is generally infeasible.

Visually Equivalent Uniform Recovery Guaranteed by Bit Rearrangement. As established above, the ideal uniformity condition cannot be exactly satisfied through polynomial selection due to the inherent algebraic constraints of primitive irreducible polynomials. However, once $f_1(\alpha)$, $\cdots$, $f_t(\alpha)$ are determined, the positions of correctly decoded coefficients in $e(\alpha)$ are fixed and predictable (see Fig. 3 and Examples 2-3). This enables an alternative strategy: instead of seeking polynomials that yield the exact number of correctly decoded coefficients, we design a bit rearrangement that optimally utilizes the available correctly decoded positions. The mapping key K_m allocates the desired bits to the positions in $e(\alpha)$ that are known to be correctly decoded. The selection of desired bits and the construction of K_m are guided by the goal of making the decoded image at each progressive level as close as possible to the corresponding reference image.

The experimental results confirm the effectiveness of this approach. As shown in Table III, for the 2-level scheme using $F_1(\alpha)$ with $K_m^{(1)}$, the 1-st level PSNR is 10.11 dB, identical to the reference value of 10.11 dB. As shown in Table V, for the 4-level scheme using $F_{16}(\alpha)$ with $K_m^{(16)}$, the PSNR values at levels 1 through 3 are $(8.35, 10.21, 12.84)$, closely approximating the reference values of $(8.35, 10.11, 13.12)$. These results demonstrate that, with the aid of bit rearrangement, the proposed scheme achieves progressive recovery that is visually equivalent to uniform recovery.

Influence of Image Content on Uniformity. The bit rearrangement procedure is not entirely independent of the image content. The mapping key K_m is designed based on both the correctly decoded positions determined by $f_1(\alpha)$, $\cdots$, $f_t(\alpha)$ and the goal of making the decoded image as close as feasible to the reference image at each progressive level. Since the reference image is derived from the original secret, the design of K_m and the resulting recovery quality inevitably bear some dependence on the image content.

However, this dependence is effectively mitigated by the random permutation applied to the secret image prior to sharing. The permutation disrupts the spatial correlation among pixels, so that each frame is composed of pixels from diverse, uncorrelated spatial locations. This homogenizes the statistical characteristics across frames, regardless of the original image content. Consequently, the bit rearrangement operates on statistically similar frames for different images, which significantly reduces the content sensitivity of the progressive recovery quality. This is empirically confirmed by the experiments on 49 test images with diverse characteristics from the USC-SIPI and CVG-UGR datasets (Tables III-VII). Across all test images, the PSNR and SSIM values at each progressive level consistently approximate those of the reference images, validating that the proposed scheme maintains stable progressive recovery performance for arbitrary image content.

*4) **Computational Efficiency**:* Computational efficiency critically influences the practical applicability of a technique. As the calculations are based on polynomial modular arithmetic, our approach is inherently more complex than those using integer modular arithmetic. The average execution time for a sharing-recovery cycle across various progressive levels is compared in Table VIII, using 49 test images from the USC-SIPI and CVG-UGR datasets. This cycle comprises the processes of shadow generation and progressive recovery from the first to the t-th level.

For MP-SIS [13], CP-PSIS [12], MCP-PSIS [12], and our approach, the $(2, 2)$ threshold is used for evaluation. Meantime, the $(2, 3)$, $(2, 4)$, $(2, 5)$ thresholds based on PSIS-BP [21] are employed for the 2, 3, 4 levels, respectively. As shown in Table VIII, the proposed method is slightly less computationally efficient than MP-SIS [13], CP-PSIS [12], MCP-PSIS [12], all of which employ integer modular arithmetic. However, it performs noticeably better than PSIS-BP [21], which relies on polynomial modular arithmetic.

*5) **Feature comparison**:* A comprehensive comparison of features between MP-SIS [13], CP-PSIS [12], MCP-SIS [12], PSIS-BP [21], VSS with maximum contrast (VSS-MC) [25], VSS for Evolving access structure (VSS-EAS) [26], JPEG PSIS (JPSIS) [23], PSIS based on deep learning and RDHEI (PSIS-DR) [22], and our technique is presented in Table IX. Among them, the methods in [25] [26] are visual secret sharing such that they can provide easy decoding. The remaining approaches employ modular operation for secret reconstruction. Benefits of our scheme are listed below.

- **Uniform progressive recovery.** One paramount consideration of previous PSIS methods is the non-uniform progressive recovery. The secret information is not gradually decoded when more shared data are utilized. On the contrary, our method can provide uniform progressive recovery.
- **Enhanced image quality for lower-level decoded result.** Existing methods, such as MP-SIS, CP-PSIS, and PSIS-BP, suffer from the defect of inferior recovery clarity. The decoded images from low-levels might be random-looking, which is challenging to identify the secret information. Conversely, our technique offers superior recovery clarity.
- **Distortion-free secret reconstruction.** Conventional SIS techniques using integer modular operation and VSS methods reconstruct the secret in a lossy manner. However, our approach enables lossless secret recovery.

VI. Conclusion

In this research, multiplicative primitive irreducible polynomials were integrated with polynomial ring CRT to constitute a PSIS. The approach with 2 progressive levels was first investigated, where 2 suitable polynomials $f_1(\alpha)$ and $f_2(\alpha)$ determine a main polynomial $F(\alpha) = f_1(\alpha) \times f_2(\alpha)$. A $(k -$

1)-degree polynomial $g(x)$ modulo $F(\alpha)$ is constructed for shadow generation, and the feasibility of the 2-level technique was analyzed theoretically. The scheme was further extended to a general t-level progressive framework. Bit rearrangement was designed to enhance the recovery clarity and to achieve visually equivalent uniform progressive recovery. Extensive experiments and comparisons illustrate the effectiveness and benefits of the proposed methods.

REFERENCES

[1] C. Yu, X. Zhang, C.-N. Yang, X. Zhang, and Z. Tang, "Reversible data hiding in shared images using overlapped coefficients in polynomials," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 36, no. 1, pp. 521–536, 2026.

[2] C. Yu, X. Zhang, C. Qin, and Z. Tang, "Reversible data hiding in encrypted images with secret sharing and hybrid coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 11, pp. 6443–6458, 2023.

[3] Y. Ma, X. Chai, G. Long, Z. Gan, and Y. Zhang, "Tpe for jpeg images with dynamic m-ary decomposition and adaptive threshold constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 9, pp. 8864–8879, 2025.

[4] L. Xiong, R. Ding, C.-N. Yang, and Z. Fu, "Robust secret image sharing scheme based on polynomial k-consistency," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 9, pp. 8880–8892, 2025.

[5] 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.

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

[7] 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.

[8] X. Wu and C.-N. Yang, "Invertible secret image sharing with steganography and authentication for ambtc compressed images," *Signal Processing: Image Communication*, vol. 78, pp. 437–447, 2019.

[9] 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.

[10] P. Li, Z. Liu, and C.-N. Yang, "A construction method of (t, k, n)-essential secret image sharing scheme," *Signal Processing: Image Communication*, vol. 65, pp. 210–220, 2018.

[11] 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.

[12] L. Xiong, X. Han, and C.-N. Yang, "Cp-psis: Crt and polynomial-based progressive secret image sharing," *Signal Processing*, vol. 185, p. 108064, 2021.

[13] C.-N. Yang, C.-E. Zheng, M.-C. Lu, and X. Wu, "Secret image sharing by using multi-prime modular arithmetic," *Signal Processing*, vol. 205, p. 108882, 2023.

[14] 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.

[15] 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.

[16] R.-Z. Wang and S.-J. Shyu, "Scalable secret image sharing," *Signal processing: Image communication*, vol. 22, no. 4, pp. 363–373, 2007.

[17] C.-N. Yang and S.-M. Huang, "Constructions and properties of k out of n scalable secret image sharing," *Optics Communications*, vol. 283, no. 9, pp. 1750–1762, 2010.

[18] C.-N. Yang and Y.-Y. Chu, "A general (k, n) scalable secret image sharing scheme with the smooth scalability," *Journal of Systems and Software*, vol. 84, no. 10, pp. 1726–1733, 2011.

[19] Y.-X. Liu, C.-N. Yang, and P.-H. Yeh, "Reducing shadow size in smooth scalable secret image sharing," *Security and Communication Networks*, vol. 7, no. 12, pp. 2237–2244, 2014.

[20] Y. Liu and C. Yang, "Scalable secret image sharing scheme with essential shadows," *Signal Processing: Image Communication*, vol. 58, pp. 49–55, 2017.

[21] H. Chen, L. Xiong, and C.-N. Yang, "Progressive secret image sharing based on boolean operations and polynomial interpolations," *Multimedia Systems*, vol. 30, no. 4, p. 189, 2024.

[22] M. He, X. Xie, X. Wang, and L. Zhang, "Progressive secret sharing through the integration of deep learning and reversible data hiding in encrypted images," *IEEE Internet of Things Journal*, vol. 13, no. 3, pp. 4564–4577, 2026.

[23] Y. Guan and Z. Wu, "Progressive recovery method for jpeg images based on hybrid threshold secret sharing," in *2025 IEEE International Conference on Blockchain Technology and Information Security*. IEEE, 2025, pp. 1–5.

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

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

[26] 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.

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.

Xinyue Wang is currently pursuing the master's degree with the Department of Computer Science, Jinan University, China. Her research interests are secret image sharing and data hiding.

Bing Chen received the Ph.D. degree in computer science and technology from the Sun Yat-sen University, Guangzhou, China, in 2020. He is an associate professor with the School of Cyber Security, Guangdong Polytechnic Normal University, Guangzhou, China. His research interests include multimedia security, data hiding, and secret sharing. He is a member of the IEEE.

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 is an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology, ACM Transactions on Multimedia Computing, Communications and Applications, a Senior Area Editor of IEEE Signal Processing Letters. He currently holds the position of Chair of ACM Multimedia Chapter of New Zealand, a Full Board Member of AI Researchers Association (NZ), he is a Fellow of Engineering New Zealand Te Ao Rangahau (FEngNZ).