Reversible Data Hiding in Encrypted Images with Secret Sharing and Multivariate Linear Equation

Chunqiang Yu, Xianquan Zhang, Guoqiang Li, Peng Liu, Xinpeng Zhang, *Member, IEEE* and Zhenjun Tang, *Member, IEEE*

Abstract—Reversible data hiding in encrypted images (RDHEI) is an essential data security technique. Most RDHEI methods with secret sharing cannot perform well on the images with low redundancy, such as the complex texture images. To address this issue, we propose an RDHEI method with (n, k) threshold-secret sharing (SS), which is universal for the images with diverse content since it is unrelated with the image content. Specifically, the original image is shared by polynomials over Galois field GF(2^8) to generate n image shares. Two kinds of reference matrices are elaborated to guide data embedding and extraction in image shares, one for higher capacity and the other for less communication overhead. At the decoder stage, the marked pixel shares are viewed as the unknowns to construct the multivariate linear equation (MLE) and the original image can be recovered by solving MLE. Experiment results show that the proposed method outperforms some state-of-the-art SS-based RDHEI methods.

Index Terms—Reversible data hiding, encrypted images, secret sharing, high payload, multivariate linear equation.

1 Introduction

Data hiding technique [1, 2] imperceptibly embeds the additional data into a cover image so that the generated stego-image is similar to the cover image. Due to its imperceptibility, it can be applied in many security fields, such as secret transmission, copyright protection, privacy protection and cloud service. In general, the data embedding degrades image quality. However, permanent distortion is not acceptable in some sensitive fields, such as legal forensics, medical and military imagery. To deal with this issue, reversible data hiding (RDH) was proposed to extract the additional data meanwhile recover the cover image precisely. Due to its unique characteristic, many researchers pay close attention to RDH [3]. Nowadays, the existing RDH schemes can be mainly classified into five categories, namely, difference expansion (DE) [4], histogram shifting (HS) [5], lossless compression [6, 7], Dual-Image [8–11] and prediction error expansion (PEE) [12–18] . These techniques pursue an optimal trade-off between embedding rate and modification distortion.

With the development of cloud storage and cloud computing, more and more users' files such as images, videos are uploaded to the cloud for storage service. The uploaded files are usually encrypted for privacy protection. At the cloud server, the administrator embeds some additional data into the encrypted files for management and maintenance. Considering the requirements of the cloud user and administrator, reversible data hiding in encrypted images (RDHEI) was proposed.

Generally, there are three entities, namely, content-owner, data-hider and receiver in RDHEI. The content-owner performs image encryption to avoid image content leakage. The data-hider performs data embedding on the encrypted image without knowing the original image content. The receiver can perfectly perform data extraction and/or image recovery according to his/her authorized rights. RDHEI methods can be categorized into vacating room after encryption (VRAE) and reserving room before encryption (RRBE). Early VRAE methods [19–24] employing stream ciphers disrupted pixel correlation, resulting in low embedding capacity. To enhance capacity, recent VRAE approaches [25–32] use block-based encryption, such as block permutation and co-XOR,which preserves partial intra-block pixel correlations, enabling higher capacity. Nevertheless, these block-based techniques are vulnerable to known-plaintext attacks. In RRBE methods [33–39], the content owner takes full advantage of the redundancy of the original image prior to encryption, yielding a high capacity. These methods typically achieve higher embedding capacity than VRAE methods. However, RRBE requires the additional operations and introduces significant computational overhead.

Although RDHEI can provide both confidentiality for the image and secret data, most RDHEI methods are fragile in attacks, such as noise and cropping. To improve the security, some (n, k)-threshold secret sharing (SS) based RDHEI methods were proposed. Specifically, the original image is divided into several shares using the SS scheme. Each share can accommodate the additional data. The original image can be recovered by sufficient shares so that image security can be improved. SS based RDHEI can be applicable to various scenarios, such as remote medical consultation, as illustrated in Fig. 1. A medical image is outsourced to the cloud for storage and management. To ensure privacy, a

- *C. Yu, X. Q. Zhang, G. Li, P. Liu and Z. Tang are with the Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin 541004, China, with the Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin 541004, China, and also with Guangxi Engineering Research Center of Educational Intelligent Technology, Guangxi Normal University, Guiling, 541004, China. X. P. Zhang is with the School of Computer Science, Fudan University, Shanghai 200433, China. (Corresponding authors: Xianquan Zhang, Zhenjun Tang.) E-mails: yu_chunqiang@126.com; zxq6622@163.com; masterlgx@163.com; liupeng@gxnu.edu.cn;tangzj230@163.com; zhangxinpeng@fudan.edu.cn*

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3590551

medical center divides the image into n encrypted shares and then distribute them to n distinct cloud servers operated by competitive providers. Each server can embed additional data, such as timestamps, annotations, or copyright information into its share for authentication management and copyright protection. The authorized doctor can then reconstruct the original medical image by obtaining any k shares even if some servers are powered down. Furthermore, since the n servers belong to competing providers, they are inherently disinclined to perform collusion attack of the $k-1$ servers.

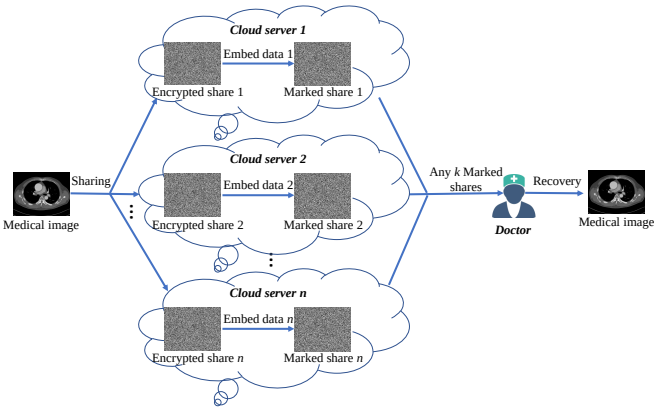

Fig. 1. Secure storage of medical image in remote medical consultation

Most SS based RDHEI methods are designed under the idea of exploiting original image block redundancy so that they achieve high embedding capacities. However, these methods have two inadequacies, namely, security concern and low embedding capacity for the complex textured images. Rising security concern is because the share blocks preserve pixel correlations within the plaintext blocks [40]. An attacker can exploit this vulnerability to potentially extract the content of the original plaintext image. In addition, it is difficult to vacate the embedding space from the complex textured image with weak pixel correlations, resulting in low embedding capacity.

To address the above issues, we propose a high capacity and secure RDHEI with secret sharing and multivariate linear equation. The content-owner divides the original image into several shares using polynomials without the preservation of pixel correlation within the plaintext block. Each data-hider embeds the additional data into the personal share by bit replacement according to the reference matrix. Finally, image recovery can be performed by multivariate linear equation. The main contributions of the proposed method are summarized as follows.

(1) We present a high embedding capacity reversible data hiding in encrypted images. An original image is shared by the polynomials over Galois field GF(2^8) to generate n image shares. Secret data is embedded into each image share using bit replacement technique. Our method is unrelated with the image content, which can achieve a high embedding capacity for the complex textured image.

(2) We design two kinds of 0-1 matrices to guide data embedding in image shares, one contributes to higher capacity and the other has less communication overhead. According to the matrices, the pixel shares corresponding to "1"s accommodate secret data to generate the marked shares and those corresponding to "0"s are unmodified.

(3) We construct multivariate linear equation to perform image recovery. For an original pixel, its marked shares are regarded as the unknowns. A multivariate linear equation is constructed by mapping the coefficients of the original polynomial and the coefficients of the polynomial reconstructed by Lagrange interpolation to recover the marked shares. The original pixel can be recovered by k shares including the recovered shares and unmodified shares.

(4) Experimental results demonstrate that the proposed method outperforms the existing state-of-the-art methods regarding embedding rate. Theoretical analysis and the experiments demonstrate that the proposed method can achieve an embedding rate as large as 6 bpp or 6.4 bpp when the sharing threshold is set to ($n = 4$, $k = 4$) or ($n = 5$, $k = 5$), respectively.

The rest of this paper is organized as follows. Section 2 reviews the related RDHEI methods. Section 3 presents the proposed method. Section 4 illustrates an example of the proposed method in details. Section 5 discusses the experimental results. Finally, Section 6 concludes this paper.

2 Related Work

Up to now, many excellent RDHEI methods have been proposed. RDHEI methods can be categorized into reserving room before encryption (RRBE), vacating room after encryption (VRAE) and secret sharing (SS) based methods.

2.1 VRAE based methods

In early RDHEI methods [19–24], the secret data is embedded into the encrypted image directly generated by standard image encryption algorithms such as stream cipher and the advanced encryption standard (AES). In Zhang's method [19], the encrypted image is divided into several blocks and one secret bit is embedded by flipping the three least significant bits (LSBs) of half pixels in each block. At the decoder side, a fluctuation function is designed to extract secret bits after image decryption. However, the fluctuation function does not work when the block size is small, which results in errors on data extraction and image recovery. Since then, the improvements [20, 21] have been made to improve accuracy. The methods can not perform data extraction without the encryption key. More flexibly, some separable methods [22, 23] were proposed so that data extraction is independent of image recovery. Wu et al. [24] randomly selected some pixels from the encrypted image and replaced high bit-planes of the selected pixels with secret bits. The prediction technique is used to recover the original image. Although stream cipher can provide good confidentiality, it disorganizes pixel spatial correlation and results in low payloads.

Some specific encryption based methods were proposed to provide redundancy and confidentiality. In [25–28], the original image is divided into several blocks. Then, block permutation and block based bit-XOR are used to perform encryption and each encrypted block has redundancy. As the redundancy, data hiding can be performed by difference histogram shifting (DHS) [25], adaptive block encoding

[26, 27], and difference compression [28]. Liu et al. [29] adopted bit plane disordering and block permutation to transfer the redundancy of the original image to the encrypted image. Then, the embedding room can be released from the encrypted image by efficient sparse code. Qin et al. [30] improved Liu et al.' method [29] by bit planes disordering with higher security and more efficient sparse code. In [31], block permutation and block based modulation are first used for encryption and meanwhile spatial correlations within image blocks are preserved. Then, the secret data are embedded into the encrypted image using parametric binary tree labeling (PBTL). Using the same encryption as [31], Yu et al. [32] proposed an RDHEI method with adaptive difference recovery (ADR) to achieve high embedding capacity. In these specific encryption based methods, the security of the image may be limited, such as known plaintext attack due to the redundancy within the encrypted images. Even so, the security of the encrypted image can be improved by a dynamic key.

2.2 RRBE based methods

To impove the embedding capacity, some RRBE based methods [33–39] were proposed. These methods have high payloads. Ma et al. [33] released room before image encryption by using the traditional RDH method and image self-embedding. Cao et al. [34] used the patch-level sparse representation technique to release the room from the original image. In [35], a binary-block embedding (BBE) technique is first proposed and then applied to vacate the room before encryption. Chen and Chang [36] released the embedding room by rearranging the most significant bits (MSBs). Yin et al. [37] first labeled the same successive bit-planes from high to low bit-planes by comparing an original pixel and its prediction value and then compressed the generated labels with Huffman coding to vacate the room. Yu et al. [38] hierarchically divided the prediction error into three ranges. The pixel with a small or large range of prediction error can accommodate a bit in each layer, which achieves a high payload. Xu et al. [39] proposed an RDHEI with hierarchical block variable length coding. A bit-plane of the original image is first divided into some blocks with different hierarchical levels. Then, the embedded room is released by a variable length coding scheme.

2.3 SS based methods

In the aforementioned RDHEI methods, an encrypted image or a marked encrypted image is distributed to a cloud server. The image security is decided by one party. Once the cloud server is attacked or it is untrusted. The security of the image within it will suffer from threat. To improve image security, some secret sharing (SS) based methods were proposed [41–49]. The content owner divides an original image into several image shares using SS techniques. The data-hiders embed the secret data into the image shares. At the receiver side, the original image can be recovered by collecting sufficient shares or marked shares even if the other shares are corrupted or missing. The image is safeguarded by multiparty. Wu et al. [41] designed a pairwise Shamir's SS to preserve the difference of each pixel pair. Then, data embedding can be performed on each share by DE or DHS technique due to difference preservation. In [42], a pair of pixels is first transformed by DE and then encrypted by a 3-degree polynomial. The encrypted pixel pair can accommodate one secret bit. In [41, 42], one data-hider performs the whole data hiding.

To further improve security, some multiple data-hiders based methods were proposed [43–47, 49], in which different data-hiders perform data hiding on personal shares independently. In method [43], two SS based RDHEI schemes over Galois fields GF(p) and GF(2^8) were proposed. These two schemes both preserve the pixel correlations for each share so that the embedding room can be released in each share. Chen et al. [44] used a specific Shamir's SS [50] to generate multiple shares and distribute them to multiple data-hiders. Secret data is embedded by bit replacement. The original image can be recovered by solving the polynomial coefficients. In [45], the image shares are generated by Chinese remainder theorem-based secret sharing (CRTSS) [51]. According to the additive homomorphism of CRT, secret data can be embedded in each share using the DE technique. The embedding rate of this method is close to 0.5 bpp. In [46], a cipher-feedback secret sharing (CFSS) technique is introduced to perform image sharing. A multi-MSB prediction method is used to release the embedding room for each share. In [47], a secure matrix-based SS scheme is designed to preserve the pixel correlations of the original image for each share and then block error mixture encoding is applied to release the embedding room in each share. Hua et al. [48] first designed a secure preprocessing-free matrix secret sharing (PFMSS) technique to preserve block correlations for data embedding. Xiong et al. [49] proposed an RDH in shared images (RDHSI) based on syndrome decoding and homomorphism. An original image and secret data are first both preprocessed with Hamming code and then data hiding is performed by an addition operation in GF(2^8) between the shares of the preprocessed image and secret data. In the method [45, 49], the image and secret data are both fault-tolerant. In [52], a CRTSS scheme with constraints is designed to preserve redundancy for each share and meanwhile provide high-level security. A hybrid coding is used to vacate embedding room from each share.

3 Proposed Method

In this paper, we propose a high payload reversible data hiding in encrypted images with secret sharing and multivariate linear equation. Fig. 2 exhibits the framework of the proposed method, including image sharing, data embedding, and data exaction and image recovery. The content owner divides an original image into n image shares using polynomial over GF(2^8) and then these n shares are distributed to n data-hiders. To perform RDH, we elaborate two kinds of matrices. One can contribute to a higher capacity and it is essential for image recovery. Whereas the other is not essential and has less communication overhead. Each data-hider independently performs data embedding by bit replacement according to any kind of matrix. After receiving any k marked shares, the original image can be recovered losslessly using multivariate linear equation.

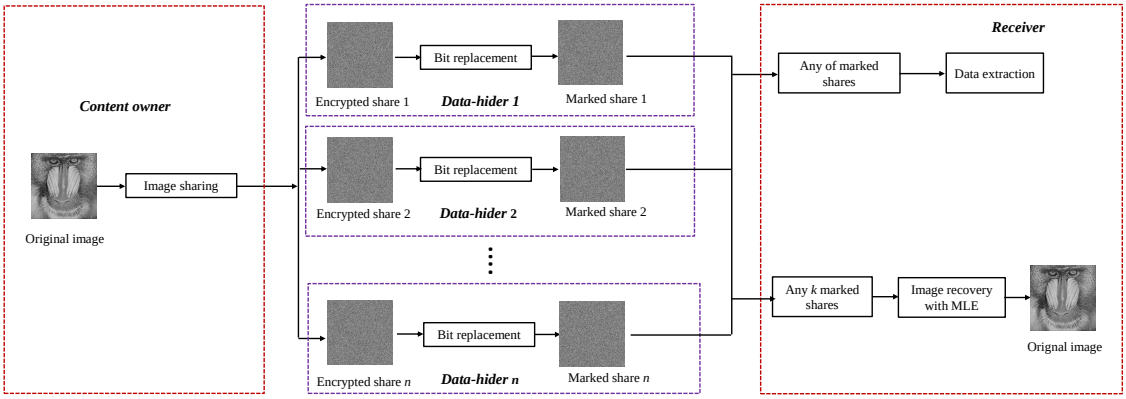

Fig. 2. Framework of the proposed method

3.1 Image sharing

Suppose that the original image $\mathbf{I}$ is an 8-bits grayscale image with $H \times W$ size and $I_{i,j}$ is the pixel value located at (i,j) of $\mathbf{I}$, where $0 \leq I_{i,j} \leq 255$, $1 \leq i \leq H$ and $1 \leq j \leq W$. In Shamir's sharing for images, $I_{i,j} > 251$ cannot be shared and needs to be shrunk for lossless recovery since 251 is the greatest prime within all 8-bits values. To address this issue, we construct a polynomial over GF(2^8) for each pixel to perform sharing. Specifically, $\mathbf{I}$ is transformed into a one-dimensional pixel sequence denoted by $\boldsymbol{P} = [p_1, p_2, \cdots, p_u]$, where $u = HW$ in Hilbert scanning order. The following polynomial is constructed for each pixel p_i as

$$\begin{aligned} f_i(x) &= (p_i + \sum_{j=1}^{k-1} a_{i,j}x^j) \bmod g(x) \\ &= (p_i + a_{i,1}x + a_{i,2}x^2 +, \cdots, +a_{i,k-1}x^{k-1}) \bmod g(x), \end{aligned} \tag{1}$$

where $1 \leq i \leq u$, p_i is regarded as the constant term which will be shared, $g(x) = x^8 + x^4 + x^3 + x + 1$ is an irreducible polynomial, and $a_{i,j} \in GF(2^8)$ is the polynomial coefficient which is a randomly chosen integer generated by an encryption key. Note that the polynomial coefficient $a_{i,j}$ is used to recover the constant term, namely, p_i. Then, a non-zero value $x_{i,t}(1 \leq t \leq n)$ is generated for the t-th share of p_i and each value in $[x_{i,1}, x_{i,2}, \cdots, x_{i,n}]$ must be distinct from any other value. The t-th share $f_i(x_{i,t})$ can be derived by substituting $x_{i,t}$ into Eq.(1). Clearly, $f_i(x_{i,t}) \in GF(2^8)$. We can obtain n shares $[f_i(x_{i,1}), f_i(x_{i,2}), \cdots, f_i(x_{i,n})]$ for p_i. Finally, n image shares $\{\mathbf{E}^{(1)}, \mathbf{E}^{(2)}, \cdots, \mathbf{E}^{(n)}\}$ are distributed to n different data-hiders.

3.2 Data embedding

After receiving the associated image share $\mathbf{E}^{(t)}$, the t-th data-hider can embed the secret data into $\mathbf{E}^{(t)}$ by bit replacement. The embedding details are illustrated as follows.

First, $\mathbf{E}^{(t)}$ is transformed into a one-dimensional pixel sequence $[f_1(x_{1,t}), f_2(x_{2,t}), \cdots, f_u(x_{u,t})]$, where $1 \leq t \leq n$. The one-dimensional pixel sequences of all image shares are denoted by $\mathbf{Y} = \begin{pmatrix} f_1(x_{1,1}) & f_2(x_{2,1}) & \cdots & f_u(x_{u,1}) \\ f_1(x_{1,2}) & f_2(x_{2,2}) & \cdots & f_u(x_{u,2}) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_{1,n}) & f_2(x_{2,n}) & \cdots & f_u(x_{u,n}) \end{pmatrix}$, where each column denotes n shares of a pixel and each row denotes u pixels of one image share. Further, the pixels of each row can be exactly divided into h groups denoted by $[f_{g,1}(x_{1,t}), f_{g,2}(x_{2,t}), \cdots, f_{g,n}(x_{n,t})]_{g=1}^{h}$ for simplicity and each group contains n pixels, where g denotes the index of the group. Then, we use the gth groups of n sequence shares to construct a square matrix with $n \times n$ size as $\mathbf{Y}_g = \begin{pmatrix} f_{g,1}(x_{1,1}) & f_{g,2}(x_{2,1}) & \cdots & f_{g,n}(x_{n,1}) \\ f_{g,1}(x_{1,2}) & f_{g,2}(x_{2,2}) & \cdots & f_{g,n}(x_{n,2}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{g,1}(x_{1,n}) & f_{g,2}(x_{2,n}) & \cdots & f_{g,n}(x_{n,n}) \end{pmatrix}$. Evidently, $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \cdots, \mathbf{Y}_h]$. For $\mathbf{Y}_g$, we construct a 0-1 cyclic square matrix with $n \times n$ size to select embedding positions. The 0-1 cyclic square matrix is constructed as

$$\mathbf{A}_g^{(1)} = \begin{pmatrix} A_1 & A_2 & \cdots & A_n \\ A_n & A_1 & \cdots & A_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_2 & A_3 & \cdots & A_1 \end{pmatrix}, \tag{2}$$

where each column of $\mathbf{A}_g^{(1)}$ corresponds to one column of $\mathbf{Y}_g$, namely, n shares of a pixel. In this paper, $k-1$ shares of p_i can be modified since there are $k-1$ known coefficients as shown in Eq.(1). Thus, we randomly select $k-1$ elements from $[A_1, A_2, \cdots, A_n]$ according to a key and let them be "1". The remaining $n-(k-1)$ elements are set to "0". Note that the selection key is shared among all data-hiders. According to the properties of the cyclic matrix, we have that the number of "1"s in each row is $k-1$ for $\mathbf{A}_g^{(1)}$. According to the property of the cyclic matrix, each column of $\mathbf{A}_g^{(1)}$ also has $k-1$ "1"s . For example, set ($n = 5$, $k = 4$) and $[A_1 = 1, A_2 = 0, A_3 = 1, A_4 = 1, A_5 = 0]$ are randomly generated, in which the number of "1"s is $k - 1 = 3$. Then, we have $\mathbf{A}_g^{(1)} = \begin{pmatrix} A_1=1 & A_2=0 & A_3=1 & A_4=1 & A_5=0 \\ A_5=0 & A_1=1 & A_2=0 & A_3=1 & A_4=1 \\ A_4=1 & A_5=0 & A_1=1 & A_2=0 & A_3=1 \\ A_3=1 & A_4=1 & A_5=0 & A_1=1 & A_2=0 \\ A_2=0 & A_3=1 & A_4=1 & A_5=0 & A_1=1 \end{pmatrix}$. It can be seen that each row or each column both has $k-1=3$ "1"s.

Then, a reference matrix for all image shares is generated as $\mathbf{A}^{(1)} = [\mathbf{A}_1^{(1)}, \mathbf{A}_2^{(1)}, \cdots, \mathbf{A}_h^{(1)}]$, where each row of $\mathbf{A}^{(1)}$ corresponds to all pixels of an image share and each column

TABLE 1
Share selections for (5,4) threshold in a matrix

5 shares	index				
1^{st} share	**1, 1**	2, 1	**3, 1**	**4, 1**	5, 1
2^{nd} share	1, 2	**2, 2**	3, 2	**4, 2**	**5, 2**
3^{rd} share	**1, 3**	2, 3	**3, 3**	4, 3	**5, 3**
4^{th} share	**1, 4**	**2, 4**	3, 4	**4, 4**	5, 4
5^{th} share	1, 5	**2, 5**	**3, 5**	4, 5	**5, 5**

of $\mathbf{A}^{(1)}$ corresponds to n shares of an original pixel p_i. Moreover, each row or each column of $\mathbf{A}^{(1)}$ has $(k-1)$ "1"s.

According to $\mathbf{A}^{(1)}$, data hiding can be performed as follows. If $\mathbf{A}^{(1)}_{i,t} = 1$, 8 bits of secret data are embedded into $f_i(x_{i,t})$ by bit replacement technique to generate a marked pixel share, where $1 \leq i \leq u$ and $1 \leq t \leq n$. If $\mathbf{A}^{(1)}_{i,t} = 0$, $f_i(x_{i,t})$ keeps unchanged. Finally, the n shares of p_i are generated as $[f_i^{'}(x_{i,1}), f_i^{'}(x_{i,2}), \cdots, f_i^{'}(x_{i,n})]$ after data embedding. By the aforementioned selection mechanism, just $k-1$ shares of p_i accommodate the secret data and there are unmodified $n-(k-1)$ shares. In addition, the possibility of correctly extracting data from each group is $\frac{1}{\binom{n}{k-1}}$. Thus, the possibility of correctly extracting whole data is $\frac{1}{\binom{n}{k-1}^h}$, where h is the number of groups. For example, suppose that $n=4$, $k=3$ and the image size is 512×512. We have $h = 512 \times 512/4 = 65536$. In this scenario, the possibility of correctly extracting whole data from each image share is $\frac{1}{\binom{4}{2}^{65536}}$, which is extremely small.

Next, we take Tab.1 to illustrate the share selection for ($n=5$, $k=4$) threshold in a matrix. In Tab. 1, the selected shares are marked by bold text. It can be seen that $k-1=3$ shares are used to embed data for each p_i.

By the above operation, there are $k-1$ marked shares in n shares of a pixel. Evidently, at least one remains unchanged in any k shares after data embedding, which is helpful for image recovery. Meanwhile, each image share can accommodate the same payload. Finally, the t-th sequence share $[f_1^{'}(x_{1,t}), f_2^{'}(x_{2,t}), \cdots, f_u^{'}(x_{u,t})]$ is transformed into a marked image share. At the decoder stage, p_i can be recovered losslessly since no more than $k-1$ shares are modified. The recovery will be discussed in Section 3.3.2. According to the above embedding method, the embedding rate (ER) can be calculated by

$$ER^{(1)} = \frac{8 \times (k-1)}{n}. \tag{3}$$

As Eq.(10) shows, our ER is only related to n and k, not related to the original image content. Consequently, our method can also contribute to the large ERs for the complex texture images. The ER is larger when $k-1$ is closer to n in the proposed method. Our theoretical maximal ER is close to 8 bpp when k and n are toward positive infinity. However, $n <= 6$ is applied in most scenarios. Tab. 2 gives different theoretical ERs $\mathbf{A}^{(1)}$ for different combinations of (n, k). For example, $ER^{(1)} = \frac{8\times4}{5} = 6.4$ bpp when $n=5$ and $k=5$, which is a very high embedding capacity.

TABLE 2
Theoretical ERs with $\mathbf{A}^{(1)}$

(n, k)	(2, 2)	(3, 3)	(4, 3)	(5, 3)	(5, 4)	(5, 5)
ER(bpp)	4.00	5.33	4.00	3.20	4.80	6.40

3.3 Data extraction and image recovery

3.3.1 Data extraction

When receiving the t-th marked image share, secret data can be extracted without errors. First, the marked image share is transformed into a one-dimensional pixel sequence $[f_1^{'}(x_{1,t}), f_2^{'}(x_{2,t}), \cdots, f_u^{'}(x_{u,t})]$. Secret data can be extracted from the marked share with $\mathbf{A}^{(1)}_{i,t} = 1$, where $1 \leq i \leq u$. Finally, the complete secret data can be obtained.

3.3.2 Image recovery with reference matrix

When any k marked image shares are collected, the original image can be recovered losslessly. Next, we take a pixel p_i to illustrate the recovery with a polynomial, where $1 \leq i \leq u$. Suppose that $[f_i^{'}(x_{i,t_1}), f_i^{'}(x_{i,t_2}), \cdots, f_i^{'}(x_{i,t_k})]$ are randomly selected from n marked shares to recover p_i. It is clear that some values among $[f_i(x_{i,t_1}), f_i(x_{i,t_2}), \cdots, f_i(x_{i,t_k})]$ are changed during data embedding, but no more than $k-1$ according to the matrix $\mathbf{A}^{(1)}$. In addition, at least one share is not changed. According to the inverse process of Shamir's secret sharing, the $(k-1)$-degree polynomial f using the Lagrange interpolation can be reconstructed as

$$f_i(x) = \left(\sum_{\beta=1}^{k} f_i(x_{i,t_\beta}) \prod_{1 \leq \alpha \leq k, \alpha \neq \beta} \frac{x - x_{i,t_\alpha}}{x_{i,t_\alpha} - x_{i,t_\beta}} \right) \bmod g(x), \tag{4}$$

where $t_\beta, t_\alpha \in 1, 2, \cdots, n$. Clearly, Eq.(4) is a function. Then, we extend Eq.(4) as

$$\begin{aligned} f_i(x) &= \left(\sum_{\beta=1}^{k} f_i(x_{i,t_\beta}) \prod_{1 \leq \alpha \leq k, \alpha \neq \beta} \frac{x - x_{i,t_\alpha}}{x_{i,t_\alpha} - x_{i,t_\beta}} \right) \bmod g(x) \\ &= \Big(\underbrace{b_{1,1} f_i(x_{i,t_1}) + b_{1,2} f_i(x_{i,t_2}) +, \cdots, b_{1,k} f_i(x_{i,t_k})}_{p_i} \\ &\quad + \underbrace{(b_{2,1} f_i(x_{i,t_1}) + b_{2,2} f_i(x_{i,t_2}) +, \cdots, b_{2,k} f_i(x_{i,t_k}))}_{a_{i,1}} x \\ &\quad +, \ldots, + \\ &\quad \underbrace{(b_{k,1} f_i(x_{i,t_1}) + b_{k,2} f_i(x_{i,t_2}) +, \cdots, b_{k,k} f_i(x_{i,t_k})}_{a_{i,k-1}} \Big) x^{k-1} \\ &\quad \bmod g(x), \end{aligned} \tag{5}$$

where $\begin{pmatrix} b_{1,1} & \cdots & b_{1,k} \\ \vdots & \ddots & \vdots \\ b_{k,1} & \cdots & b_{k,k} \end{pmatrix}$ can be obtained by the known values x_{i,t_β} and x_{i,t_α}.

Clearly, Eq.(5) is equivalent to Eq.(1). The constant term of Eq.(5) is mapped to that of Eq.(1), namely,

$$p_i = b_{1,1} f_i(x_{i,t_1}) + b_{1,2} f_i(x_{i,t_2}) +, \cdots, + b_{1,k} f_i(x_{i,t_k}). \tag{6}$$

Evidently, some values among $[f_i(x_{i,t_1}), f_i(x_{i,t_2}), \cdots, f_i(x_{i,t_k})]$ are the marked

shares according to the reference matrix $\mathbf{A}^{(1)}$. Thus, these marked shares should be recovered prior to calculating p_i. We assume that $[\mathbf{A}^{(1)}_{i,t_1} = 1, \mathbf{A}^{(1)}_{i,t_2} = 1, \cdots, \mathbf{A}^{(1)}_{i,t_{q-1}} = 1, \mathbf{A}^{(1)}_{i,t_q} = 0, \mathbf{A}^{(1)}_{i,t_{q+1}} = 1, \cdots, \mathbf{A}^{(1)}_{i,t_k} = 1]$ for simplicity. Under this assumption, it is clear that $f_i(x_{i,t_q}) = f_i^{'}(x_{i,t_q})$ is not modified due to $\mathbf{A}^{(1)}_{i,t_q} = 0$ and $[f_i(x_{i,t_1}), f_i(x_{i,t_2}), \cdots, f_i(x_{i,t_{q-1}}), f_i(x_{i,t_{q+1}}), \cdots, f_i(x_{i,t_k})]$ are unknown in Eq. (6) and will be solved as following. Since Eq.(5) and Eq.(1) are equivalent, a $(k-1)$-variable linear equation can be constructed by mapping the coefficients between Eq.(5) and Eq.(1), which are described by Eq. (7).

$$\begin{cases} b_{2,1}f_i(x_{i,t_1}) + .. + b_{2,q}f_i(x_{i,t_q}) + .. + b_{2,k}f_i(x_{i,t_k}) = a_{i,1} \\ b_{3,1}f_i(x_{i,t_1}) + .. + b_{3,q}f_i(x_{i,t_q}) + .. + b_{3,k}f_i(x_{i,t_k}) = a_{i,2} \\ \vdots \\ b_{k,1}f_i(x_{i,t_1}) + .. + b_{k,q}f_i(x_{i,t_q}) + .. + b_{k,k}f_i(x_{i,t_k}) = a_{i,k-1}, \end{cases} \tag{7}$$

where $a_{i,1}, a_{i,2}, \cdots, a_{i,k-1}$ and $f_i(x_{i,t_q})$ are known, and $[f_i(x_{i,t_1}), f_i(x_{i,t_2}), \cdots, f_i(x_{i,t_{q-1}}), f_i(x_{i,t_{q+1}}), \cdots, f_i(x_{i,t_k})]$ are unknown. Thus, Eq.(7) is revised as

$$\begin{cases} \sum_{1\le j\le k, j\neq q} b_{2,j}f_i(x_{i,t_j}) = a_{i,1} - b_{2,q}f_i(x_{i,t_q}) \\ \sum_{1\le j\le k, j\neq q} b_{3,j}f_i(x_{i,t_j}) = a_{i,2} - b_{3,q}f_i(x_{i,t_q}) \\ \vdots \\ \sum_{1\le j\le k, j\neq q} b_{k,j}f_i(x_{i,t_j}) = a_{i,k-1} - b_{k,q}f_i(x_{i,t_q}). \end{cases} \tag{8}$$

It is clear that there are $k-1$ equations for $k-1$ unknowns. Thus, $\left[f_i(x_{i,t_1}), f_i(x_{i,t_2}), \cdots, f_i(x_{i,t_{q-1}}), f_i(x_{i,t_{q+1}}), \cdots, f_i(x_{i,t_k})\right]$ can be obtained by solving Eq.(8) and then they are substituted into Eq.(6) to recover p_i.

3.4 Image recovery without reference matrix

Clearly, $\mathbf{A}^{(1)}$ indicates the data embedding positions in each image share and it must be public to the receiver for image recovery in the above scheme. It may suffer from the additional communication overhead.

Next, we design a new reference matrix $\mathbf{A}^{(2)}$ for data embedding. Without the knowledge of $\mathbf{A}^{(2)}$, the receiver can also recover the original image. Different from the previous scheme, $\lceil k/2 \rceil - 1$ elements are randomly chosen to set "1" in $[A_1, A_2, \cdots, A_n]$ and the remaining elements are "0", where $k \geq 3$. Then, $[A_1, A_2, \cdots, A_n]$ is used to construct a matrix $\mathbf{A}^{(2)}_g$ assisting data embedding the same as the previous scheme. For example, there are $\lceil k/2 \rceil - 1 = 2$ "1"s in $[A_1, A_2, A_3, A_4, A_5]$ when $n = 5$ and $k = 5$. Suppose that $[A_1 = 0, A_2 = 1, A_3 = 1, A_4 = 0, A_5 = 0]$, then we have

$$\mathbf{A}^{(2)}_g = \begin{pmatrix} A_1=0 & A_2=1 & A_3=1 & A_4=0 & A_5=0 \\ A_5=0 & A_1=0 & A_2=1 & A_3=1 & A_4=0 \\ A_4=0 & A_5=0 & A_1=0 & A_2=1 & A_3=1 \\ A_3=1 & A_4=0 & A_5=0 & A_1=0 & A_2=1 \\ A_2=1 & A_3=1 & A_4=0 & A_5=0 & A_1=0 \end{pmatrix}$$

Further, a reference matrix for all image shares is generated as $\mathbf{A}^{(2)} = [\mathbf{A}^{(2)}_1, \mathbf{A}^{(2)}_2, \cdots, \mathbf{A}^{(2)}_h]$. The same as previous scheme, secret data is embedded into each image share according to $\mathbf{A}^{(2)}$.

In this case, $\lceil k/2 \rceil - 1$ shares of each pixel p_i accommodate secret data and the remaining $n - \lceil k/2 \rceil - 1$ shares remain unchanged, where $\lceil\rceil$ denotes a round up function. Suppose that k_1 and k_2 are the numbers of the unmodified shares and marked shares among any k shares of p_i, respectively, where $k_1 + k_2 = k$. It is clear that

$$\begin{cases} k_1 > k_2 \\ k_1 > \lceil k/2 \rceil - 1 \\ k_1 + k_2 = k \end{cases} \tag{9}$$

due to $k - (\lceil k/2 \rceil - 1) > \lceil k/2 \rceil - 1$. In short, the unmodified shares are more than the marked shares among any k shares.

Based on $k_1 > k_2$, the original pixel p_i can be recovered without the knowledge of $\mathbf{A}^{(2)}$. The detailed recovery is illustrated as follows. Although it cannot determine which share is unmodified without $\mathbf{A}^{(2)}$, the number of the unmodified shares is greater than that of the marked shares. Then, we assume that each share $f_i(x_{i,t_q})$ is unmodified in turn and $[f_i(x_{i,t_1}), f_i(x_{i,t_2}), \cdots, f_i(x_{i,t_{q-1}}), f_i(x_{i,t_{q+1}}), \cdots, f_i(x_{i,t_k})]$ are the marked shares, where $q = 1, 2, \cdots, k$. Similarly to the previous scheme, we can construct a polynomial $f_i(x)^{(q)}$ to obtain a recovered value $p_i^{(q)}$ for an unmodified pixel $f_i(x_{i,t_q})$ by solving the MLE. Actually, each unmodified share $f_i(x_{i,t_q})$ can rightly recover p_i. While, the marked share $f_i(x_{i,t_q})$ obtains a wrongly recovered value for p_i once $f_i(x_{i,t_q})$ is modified. Note that the number of the unmodified shares is greater than that of the marked shares in this scheme. Thus, the number of the correctly recovered values is greater than that of the wrongly recovered values. Among $[p_i^{(1)}, p_i^{(2)}, \cdots, p_i^{(k)}]$, the same values with the maximum number are the original value of p_i since k_1 unmodified shares generate k_1 real values of p_i and $\begin{cases} k_1 > k_2 \\ k_1 + k_2 = k \end{cases}$.

In this case, the embedding rate (ER) is

$$ER^{(2)} = \frac{8 \times (\lceil k/2 \rceil - 1)}{n} < ER^{(1)} \quad \text{s.t. } k \geq 3. \tag{10}$$

Tab. 3 gives different theoretical ERs with $\mathbf{A}^{(2)}$ for different combinations of (n, k).

TABLE 3
Theoretical ERs with $\mathbf{A}^{(2)}$

(n, k)	(3, 3)	(4, 3)	(5, 3)	(5, 4)	(5, 5)
ER(bpp)	2.67	2.00	1.60	1.60	3.20

Although this scheme has lower embedding capacity than the previous scheme, it has less communication overhead since the image recover does require $\mathbf{A}^{(2)}$.

4 A DETAILED EXAMPLE OF PROPOSED METHOD

In this section, to better illustrate the proposed method, an example is given in Fig.3, in which (4, 3)-threshold secret sharing is used. Section 4.1 illustrates the sharing procedure. Sections 4.2 and 4.3 construct two kinds of matrices, respectively. The former recovers the original image with the matrix, whereas the latter does not require the matrix.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3590551

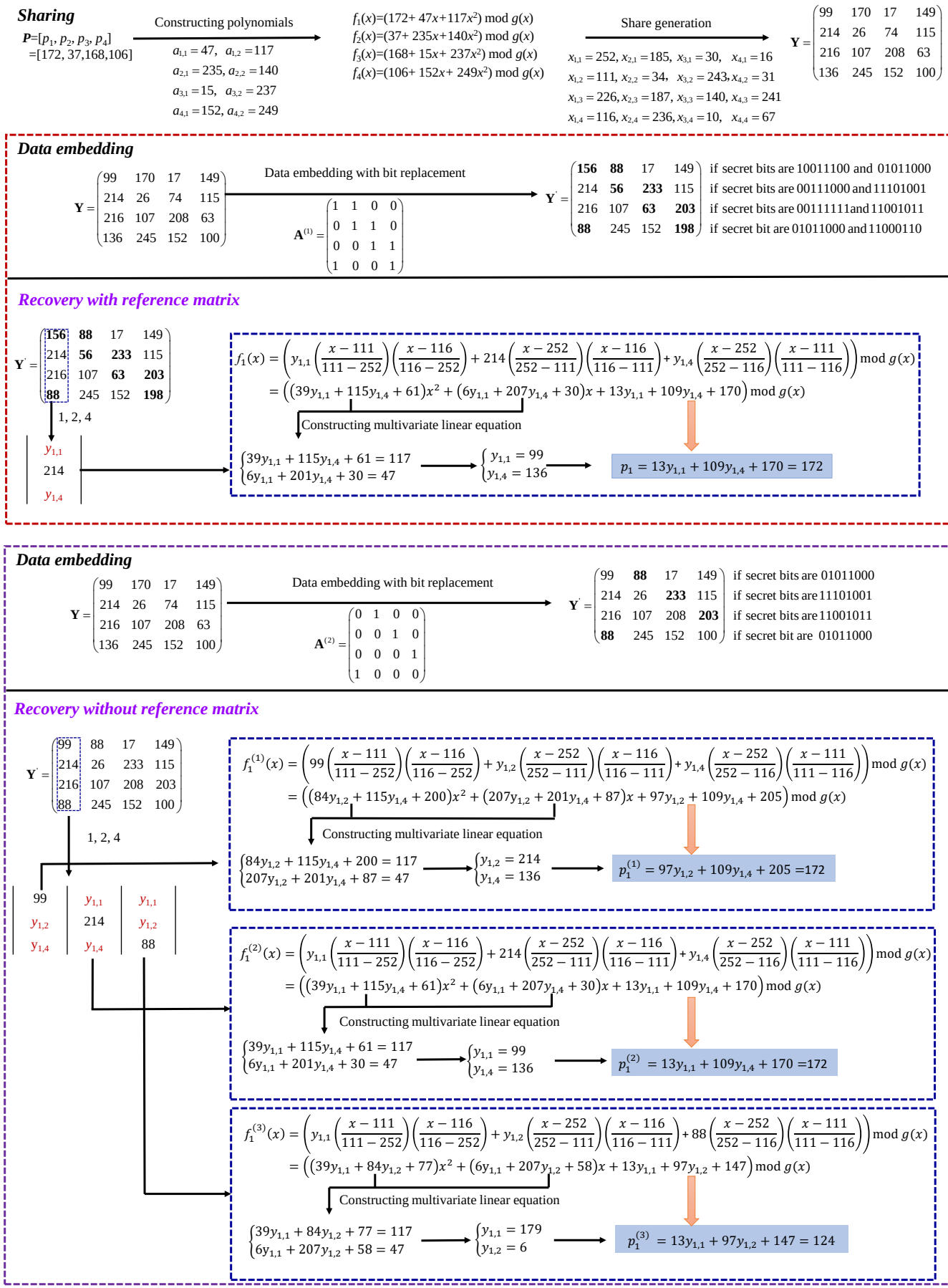

Fig. 3. A detailed example

4.1 Sharing

Suppose that a group of pixels as well as the constant terms are $\boldsymbol{P} = [p_1, p_2, p_3, p_4] = [172, 37, 168, 106]$, which will be shared. According to the encryption key, the polynomial coefficients for $\boldsymbol{P}$ are randomly generated as

$$\begin{cases} \boldsymbol{a}_1 = [a_{1,1}, a_{1,2}] = [47, 117] \\ \boldsymbol{a}_2 = [a_{2,1}, a_{2,2}] = [235, 140] \\ \boldsymbol{a}_3 = [a_{3,1}, a_{3,2}] = [15, 237] \\ \boldsymbol{a}_4 = [a_{4,1}, a_{4,2}] = [152, 249]. \end{cases} \tag{11}$$

With the constant terms p_1, p_2, p_3, p_4 and polynomial coefficients $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \boldsymbol{a}_4$, four 2-degree polynomials can be constructed over Galois field GF(2^8) as

$$\begin{cases} f_1(x) = (172 + 47x + 117x^2) \bmod g(x) \\ f_2(x) = (37 + 235x + 140x^2) \bmod g(x) \\ f_3(x) = (168 + 15x + 237x^2) \bmod g(x) \\ f_4(x) = (106 + 152x + 249x^2) \bmod g(x). \end{cases} \tag{12}$$

For each polynomial $f_i(x)$, four values are also randomly as

$$\begin{cases} \boldsymbol{x}_1 = [x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}] = [252, 111, 22, 116] \\ \boldsymbol{x}_2 = [x_{2,1}, x_{2,2}, a_{2,3}, x_{2,4}] = [185, 34, 187, 236] \\ \boldsymbol{x}_3 = [x_{3,1}, x_{3,2}, a_{3,4}, x_{3,4}] = [30, 243, 140, 10] \\ \boldsymbol{x}_4 = [x_{4,1}, x_{4,2}, a_{4,3}, x_{4,5}] = [16, 31, 241, 67]. \end{cases} \tag{13}$$

By substituting $\boldsymbol{x}_i$ into $f_i(x)$, the shares of the pixels can be obtained as

$$\mathbf{Y} = \begin{cases} [f_1(x_{1,1}), f_2(x_{2,1}), f_3(x_{3,1}), f_4(x_{4,1})] \\ = [99, 170, 17, 149] \\ [f_1(x_{1,2}), f_2(x_{2,2}), f_3(x_{3,2}), f_4(x_{4,2})] \\ = [214, 26, 74, 115] \\ [f_1(x_{1,3}), f_2(x_{2,3}), f_3(x_{3,3}), f_4(x_{4,3})] \\ = [216, 107, 208, 63] \\ [f_1(x_{1,4}), f_2(x_{2,4}), f_3(x_{3,4}), f_4(x_{4,4})] \\ = [136, 345, 152, 100]. \end{cases} \tag{14}$$

where each row denotes one share of $\boldsymbol{R}$ and each column denotes four shares of one pixel in $\boldsymbol{R}$.

4.2 Recovery with reference matrix

In this section, we use a 0-1 square cyclic matrix $\mathbf{A}^{(1)} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$ as a reference matrix to select embedding positions for $\mathbf{Y}$, where each row or each column has $k-1 = 2$ "1"s. And, $\mathbf{A}^{(1)}$ is essential for recovery

In the data embedding phase, secret bits can be embedded into a pixel share $f_i(x_{i,t})$ with $\mathbf{A}^{(1)}_{i,t} = 1$ by bit replacement. For example, the 1-st data-hider embeds '10011100' and '01011000' into $f_1(x_{1,1})$ and $f_1(x_{1,2})$ to generate the marked values $f_1^{'}(x_{1,1}) = 156$ and $f_1^{'}(x_{1,2}) = 88$ with bit replacement since $\mathbf{A}^{(1)}_{1,1} = 1$ and $\mathbf{A}^{(1)}_{1,2} = 1$. Consequently, we can obtain all marked shares as

$$\mathbf{Y}' = \begin{cases} [f_1^{'}(x_{1,1}), f_2^{'}(x_{2,1}), f_3^{'}(x_{3,1}), f_4^{'}(x_{4,1})] \\ = [\mathbf{156}, \mathbf{88}, 17, 149] \\ [f_1^{'}(x_{1,2}), f_2^{'}(x_{2,2}), f_3^{'}(x_{3,2}), f_4^{'}(x_{4,2})] \\ = [214, \mathbf{56}, \mathbf{233}, 115] \\ [f_1^{'}(x_{1,3}), f_2^{'}(x_{2,3}), f_3^{'}(x_{3,3}), f_4^{'}(x_{4,3})] \\ = [216, 107, \mathbf{63}, \mathbf{203}] \\ [f_1^{'}(x_{1,4}), f_2^{'}(x_{2,4}), f_3^{'}(x_{3,4}), f_4^{'}(x_{4,4})] \\ = [\mathbf{88}, 345, 152, \mathbf{198}]. \end{cases} \tag{15}$$

In the data extraction and image recovery phase, we randomly select the 1st, 2nd and 4th marked shares as $\begin{cases} [f_1^{'}(x_{1,1}), f_2^{'}(x_{2,1}), f_3^{'}(x_{3,1}), f_4^{'}(x_{4,1})] \\ = [\mathbf{156}, \mathbf{88}, 17, 149] \\ [f_1^{'}(x_{1,2}), f_2^{'}(x_{2,2}), f_3^{'}(x_{3,2}), f_4^{'}(x_{4,2})] \\ = [214, \mathbf{56}, \mathbf{233}, 115] \\ [f_1^{'}(x_{1,4}), f_2^{'}(x_{2,4}), f_3^{'}(x_{3,4}), f_4^{'}(x_{4,4})] \\ = [\mathbf{88}, 345, 152, \mathbf{198}] \end{cases}$. Then, secret bits can be directly extracted from the bits of the marked share with $\mathbf{A}^{(1)}_{i,t} = 1$. For example, '10011100' and '01011000' can be extracted from $f_1^{'}(x_{1,1}) = 156$ and $f_1^{'}(x_{1,2}) = 88$ due to $\mathbf{A}^{(1)}_{1,1} = 1$ and $\mathbf{A}^{(1)}_{1,2} = 1$.

Next, we use multivariate linear equation to recover the original values. Taking p_1 as an example, $f_1(x_{1,1})$ and $f_1(x_{1,4})$ are modified during data embedding since $\mathbf{A}^{(1)}_{1,1} = 1$ and $\mathbf{A}^{(1)}_{1,2} = 1$ and these two values are regarded as two unknowns $y_{1,1}$ and $y_{1,4}$. The share $f_1(x_{1,2}) = 214$ is not modified since $\mathbf{A}^{(1)}_{1,2} = 0$. Thus, the 2-degree polynomial f_1 using the Lagrange interpolation can be reconstructed as

$$\begin{aligned} &f_1(x) = \\ &\Big(y_{1,1} \frac{(x-111)(x-116)}{(111-252)(116-252)} + 214 \frac{(x-252)(x-116)}{(252-111)(116-111)} \\ &+ y_{1,4} \frac{(x-252)(x-111)}{(252-116)(111-116)} \Big) \bmod g(x) \\ &= \Big(\underbrace{(39y_{1,1} + 115y_{1,4} + 61)}_{117} x^2 + \underbrace{(6y_{1,1} + 201y_{1,4} + 30)}_{47} x + \\ &\underbrace{13y_{1,1} + 109y_{1,4} + 170}_{p_1} \Big) \bmod g(x). \end{aligned} \tag{16}$$

Clearly, $39y_{1,1} + 115y_{1,4} + 61$ and $6y_{1,1} + 201y_{1,4} + 30$ are two polynomial coefficients of $f_1(x)$, respectively. $13y_{1,1} + 109y_{1,4} + 170$ is the constant term of $f_1(x)$ as well as original pixel p_i. It can be seen that $y_{1,1}$ and $y_{1,4}$ must be calculated before recovering p_i.

Then, $y_{1,1}$ and $y_{1,4}$ can be solved by mapping the polynomial coefficients between $f_1(x)$ in Eq.(12) and $f_1(x)$ in Eq.(16) and we can construct a multivariate linear equation as

$$\begin{cases} 39y_{1,1} + 115y_{1,4} + 61 = 117 \\ 6y_{1,1} + 201y_{1,4} + 30 = 47. \end{cases} \tag{17}$$

It can be obtained that $\begin{cases} y_{1,1} = 99 \\ y_{1,4} = 136 \end{cases}$ by directly solving Eq.(17). The original value p_1 can be obtained by substituting $y_{1,1}$ and $y_{4,1}$ into the following equation.

$$p_1 = 13y_{1,1} + 109y_{1,4} + 170 = 172. \tag{18}$$

Similarly, the remaining pixels can be recovered as $p_2 = 37, p_3 = 168, p_4 = 106$.

Note $\mathbf{A}^{(1)}$ is essential for the recovery of the original pixels.

4.3 Recovery without reference matrix

In this section, we illustrate pixel recovery without reference matrix. First, the reference matrix is generated as $\mathbf{A}^{(2)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$, where each column or each row has $\lceil k/2 \rceil - 1 = 1$ "1". The same as the previous scheme, data embedding and extraction are conducted according to $\mathbf{A}^{(2)}$ as Fig.3.

Next, we also chose the 1st, 2nd and 4th shares $\begin{cases} [f_1^{'}(x_{1,1}), f_2^{'}(x_{2,1}), f_3^{'}(x_{3,1}), f_4^{'}(x_{4,1})] \\ = [99, \mathbf{88}, 17, 149] \\ [f_1^{'}(x_{1,2}), f_2^{'}(x_{2,2}), f_3^{'}(x_{3,2}), f_4^{'}(x_{4,2})] \\ = [214, 26, \mathbf{233}, 115] \\ [f_1^{'}(x_{1,4}), f_2^{'}(x_{2,4}), f_3^{'}(x_{3,4}), f_4^{'}(x_{4,4})] \\ = [\mathbf{88}, 345, 152, 100] \end{cases}$ to recover the original pixels without $\mathbf{A}^{(2)}$. We take the recovery of p_1 as an example. For simplicity, the 1st, 2nd and 4th shares of p_1 are denoted as $y_{1,1}, y_{1,2}, y_{1,4}$. It is evident that the number of the unmodified shares is greater than that of the marked shares. But, it cannot identify which one is unmodified. Then, we assume one of $y_{1,1}, y_{1,2}, y_{1,4}$ is unmodified in turn and the remaining ones are the marked shares.

First, assume that $y_{1,1}$ is unmodified and we have $y_{1,1} = 99$. The two shares $y_{1,2}$ and $y_{1,4}$ are the unknowns. The 2-degree polynomial $f_1^{(1)}$ using the Lagrange interpolation can be reconstructed as

$$\begin{aligned} &f_1^{(1)}(x) = \\ &\Big(99\frac{(x-111)(x-116)}{(111-252)(116-252)} + y_{1,2}\frac{(x-252)(x-116)}{(252-111)(116-111)} \\ &+ y_{1,4}\frac{(x-252)(x-111)}{(252-116)(111-116)}\Big) \bmod g(x) \\ &= \Big(\underbrace{(84y_{1,2} + 115y_{1,4} + 200)}_{117}x^2 + \underbrace{(207y_{1,2} + 201y_{1,4} + 87)}_{47}x \\ &+ \underbrace{97y_{1,2} + 109y_{1,4} + 205}_{p_1^{(1)}}\Big) \bmod g(x). \end{aligned} \tag{19}$$

Then, it can be derived that

$$\begin{cases} 84y_{1,2} + 115y_{1,4} + 200 = 117 \\ 207y_{1,2} + 201y_{1,4} + 84 = 47 \end{cases} \tag{20}$$

By solving the above linear equation, we have $\begin{cases} y_{1,2} = 214 \\ y_{1,4} = 136 \end{cases}$. Under the assumption of $y_{1,1} = 99$, p_1 is recovered as

$$p_1^{(1)} = 97y_{1,2} + 109y_{1,4} + 205 = 172. \tag{21}$$

Similarly, under the assumptions of unmodified shares $y_{1,2} = 214$ and $y_{1,4} = 88$, we can construct $f_1^{(2)}$ and $f_1^{(3)}$ to derive $p_1^{(2)} = 172$ and $p_1^{(3)} = 124$ for $y_{1,2} = 214$ and $y_{1,4} = 88$, respectively. Among these three results, two values are the same, namely, $p_1^{(1)} = p_1^{(2)} = 172$. Then, we can recover $p_1 = 172$ without $\mathbf{A}^{(2)}$. By the same way, the remaining pixels can be recovered.

5 Experimental results

In this section, we conduct some experiments using the proposed method with the matrix $\mathbf{A}^{(1)}$ and compare it with some existing state-of-the-art methods, where $\mathbf{A}^{(1)}$ can contribute to higher embedding capacity. Six classical images and two image datasets BOSSBase[53], BOWS2[54] are used to perform experiments. Six images consists of Lena, Jetplane, Peppers, Boat, Goldhill and Baboon. BOSSbase and BOWS2 both contain 10000 images, respectively, which provide the diverse content and the experiments conducted on these datasets are persuasive. All test images have 512×512 sizes.

5.1 Simulation results

In this part, some simulation experiments are conducted on Baboon image based on (4, 3)-threshold SS. The experimental results are shown in Fig. 4. Fig. 4(a) is the original image and its four shares are generated by the proposed method, which are shown in Figs. 4(b-e) respectively. One can see that its shares are all noise-like. Then, 4 bpp of secret data is embedded into each share to generate four marked shares as shown in Figs. 4(f-i), respectively. Finally, a recovered image can be obtained with any three of four marked shares according to MLQ. Figs. 4(j-m) are the four images recovered by three different marked shares. Compared with the original image, the PSNRs of the recovered images are all toward positive infinity, which demonstrates the original image can be recovered losslessly and verifies that our method is reversible.

5.2 Security analysis

Since the secret data and original image can both be encrypted by any secure encryption, they can be protected well and our method can provide higher security than most existing RDHEI methods by exploiting image redundancy. Next, we evaluate the security of the shared images from aspects of information entropy and correlation analysis.

Information entropy is a metric to evaluate encryption schemes. The ideal information entropy value is 8 for an 8-bit gray image $\mathbf{I}$. Clearly, the encryption scheme is more secure when the information entropy is closer to 8. The following formula illustrates the calculation of information entropy.

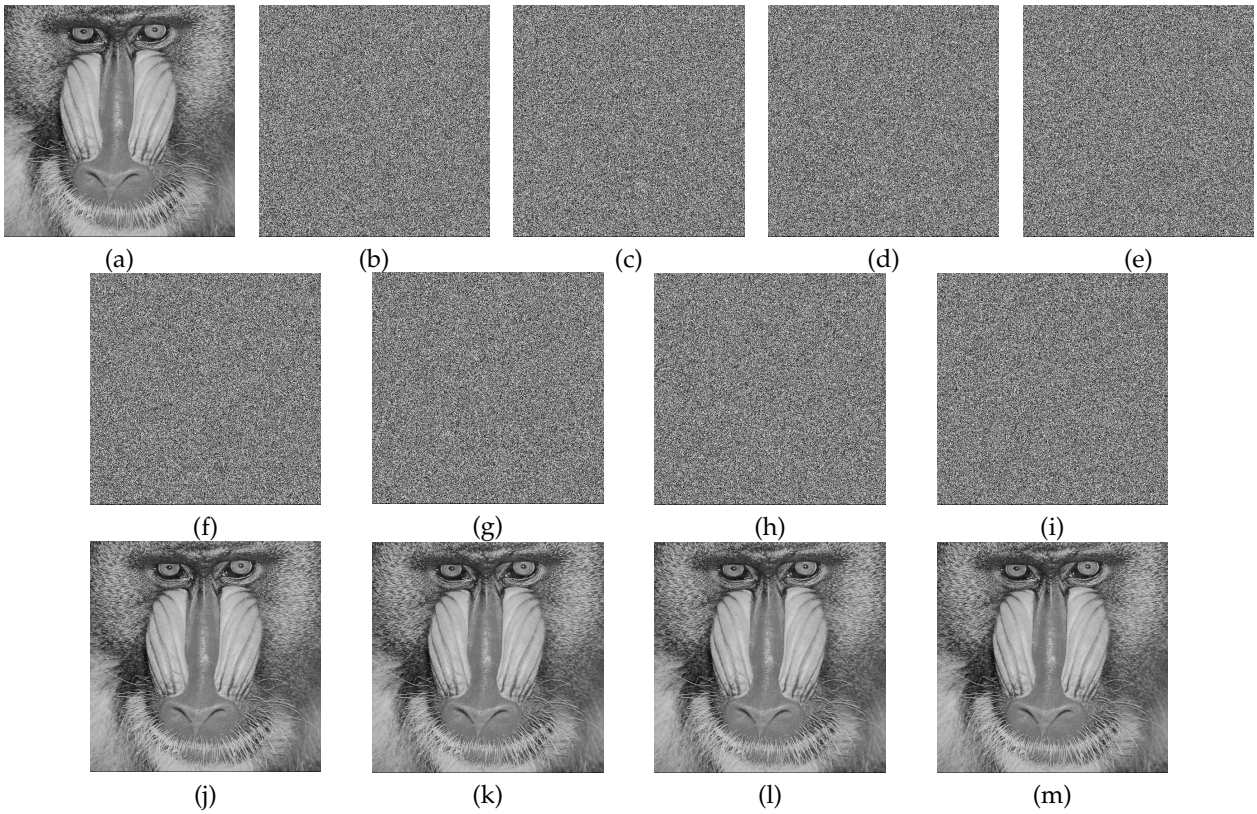

Fig. 4. Simulation experiments with (4,3)-threshold. (a) Original image; (b-e) four shares; (f-i) four marked shares with 4 bpp; (j) Recovered image (PSNR $\rightarrow +\infty$) from (f),(g) and (h); (k) Recovered image (PSNR $\rightarrow +\infty$) from (g),(h) and (i) ; (l) Recovered image (PSNR $\rightarrow +\infty$) from (f),(g) and (h) ; (m) Recovered image (PSNR $\rightarrow +\infty$) from (f),(g) and (i);

$$H(\mathbf{I}) = \frac{1}{N}\sum_{i=1}^{N} p(m_i)\log\frac{1}{p(m_i)} \tag{22}$$

where $p(m_i)$ denotes the probability of the gray level m_i and N is the number of gray levels. Tab. 4 shows the information entropy comparison among different SS based methods for (4, 3) threshold. In Tab. 4, the second column illustrates the entropies of original images and the last four columns are the average entropies of 4 shares generated by the methods [43, 48] and proposed method, respectively. One can see that our method achieves the largest entropies, which are closest to 8. Thus, our method has higher security regarding information entropy.

TABLE 4
Entropy comparison among different SS based methods for (4,3) threshold

Image	Original entropy	Average entropy		
		Qin et al. [43]	Hua et al. [48]	Proposed
Lena	7.4474	7.9707	7.9978	**7.9994**
Baboon	7.1391	7.9708	7.9966	**7.9993**
Boat	7.1238	7.9709	7.9965	**7.9993**
Peppers	7.5715	7.9707	7.9981	**7.9994**

Correlation between any two adjacent pixels is one of the most important characteristics of image encryption. In general, adjacent pixels in horizontal (H), vertical (V) and diagonal (D) direction always have high correlations. To reduce the risk of image leakage, the correlation between adjacent original pixels must be disrupted by encryption and the correlation of an encrypted one should be highly close to zero. The correlation coefficient of a group of adjacent pixels can be derived by

$$R(X,Y) = \frac{cov(X,Y)}{\delta(X)\delta(Y)} \tag{23}$$

where X denotes a series of pixels, Y denote a series of adjacent pixels of X in specific direction, $\delta(.)$ is the standard deviation function, and $cov(.)$ is the covariance function. 10000 pairs of adjacent pixels in three directions are randomly selected for the test. Tab. 5 lists the correlation coefficients of four shares generated by different methods in three directions. The methods [43] and [48] are both designed under block correlation preservation, where the block sizes are set to 2×2 and 4×4 in [43] and [48], respectively. Thus, the blocks in each share have high correlations in [43] and [48]. Compared with these two methods, the correlation coefficients of our method are smaller in three directions and close to 0.

In addition, the image is often contaminated by interchannel noise or is lost during image acquisition and transmission. Since the conventional RDHEI methods are vulnerable to attacks, the receiver cannot perfectly recover the original image once the marked image is corrupted. The proposed method uses the SS technique to generate n encrypted shares. Image recovery can be performed if at least k encrypted or marked shares are uncorrupted. Thus, the proposed method can improve the ability to resist attacks. The original image cannot be cracked even if the cracker collects $k-1$ shares.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2025.3590551

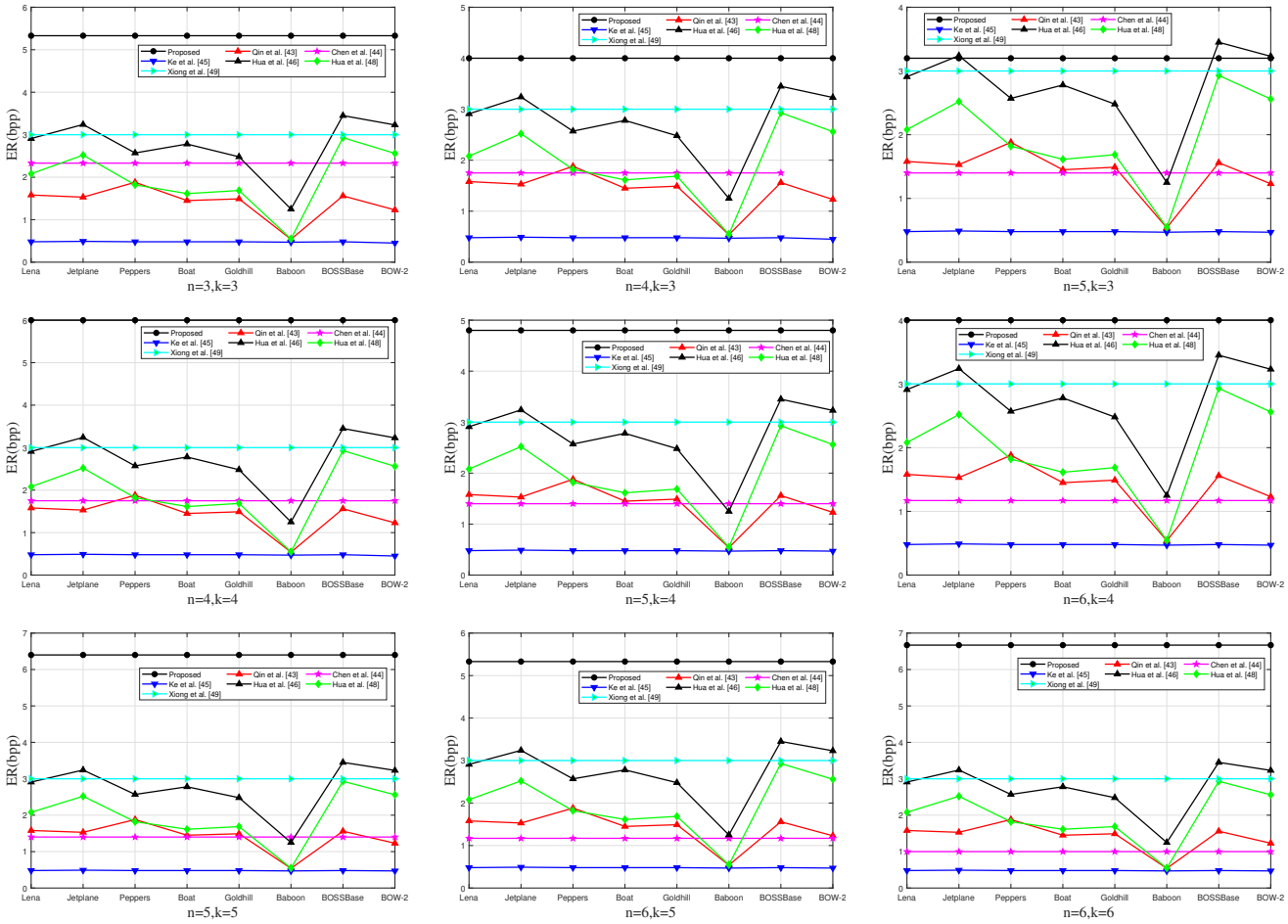

Fig. 5. ER comparison with different (n, k)-thresholds

TABLE 5
Correlation coefficient comparison among different SS based methods for (4,3) threshold

Shares	Directions	Qin et al. [43]	Hua et al. [48]	Proposed
1st share	V	0.5032	0.7470	**−0.0111**
	H	0.4922	0.7315	**−0.0086**
	D	0.2482	0.5486	**0.00590**
2nd share	V	0.4871	0.7512	**−0.0036**
	H	0.4782	0.7331	**−0.0027**
	D	0.2258	0.5556	**−0.0070**
3rd share	V	0.4965	0.7365	**−0.0088**
	H	0.4916	0.7347	**−0.0048**
	D	0.2370	0.5490	**0.00930**
4th share	V	0.4824	0.7370	**−0.0032**
	H	0.4833	0.7335	**−0.0101**
	D	0.2523	0.5405	**−0.0065**

5.3 Embedding capacity comparison

To demonstrate the superiority of the proposed method in terms of embedding performance, the proposed method is compared with seven state-of-the-art SS based methods including Qin et al. [43], Chen et al. [44] , Ke et al. [45], Hua et al. [46] , Hua et al. [48] and Xiong et al. [49]. As for the methods [43, 49], each of them contains two schemes. For fair comparison, the scheme with the larger ER is selected as the competing one in [43, 49]. For example, the Modified RDHSI in [49] is selected for comparison and its ER can reach 3 bpp for each share. In [44], the ER is decided by two parameters,namely, l and n, where l is the number of replaced bit-planes in each group with n pixels. Thus, each share can accommodate the maximum secret data with $\frac{7}{n}$ bpp when $l = 7$. If $n > 7$, the method [44] does not work. In method [45], the ER is determined by the fidelity parameter h_{fid} and h_{fid} is set to 64. In [46], the ER is determined by an optimal level l for l-MSB prediction and different images have different optimal l values. For example, the maximal ER of image Lena can reach 2.91 bpp when $l = 5$. In [48], the block size is set as 8. The methods [46, 49] require an amount of preprocessing operations and may have application limitations.

Fig. 5 shows ERs of different SS based methods under different thresholds (n, k) on different images and two datasets, where the measured ER is the average ER of n shares and (n, k) is set to (3, 3), (4, 3), (5, 3), (4, 4), (5, 4), (6, 4), (5, 5), (6, 5) or (6, 6). One can see that the proposed method has better embedding performance than all compared methods under these designated (n, k) thresholds and our ER can reach 6 bpp or 6.4 bpp when ($n = 4$, $k = 4$) or ($n = 5$, $k = 5$). In Hua et al.' method [46] , the redundancy is vacated from the original image before SS so that each share can accommodate secret data. Qin et al.'s method [43] and Hua et al.'s method [48] both designed the specific SS schemes so that each share inherits the redundancy of the original image. Then, the redundancy

of each share can be obtained by encoding techniques. In [43, 46, 48], the ER is related to the content of the original image and decided by encoding efficiency. Thus, different images have different ERs for these methods. For these methods, the image Baboon has the lowest embedding rate since Baboon has the small redundancy, which demonstrates these methods cannot perform well on the complex texture images. While, our method has the stable embedding rates for different images. As for the method [44], the ER is related to n and independent of the image content. The ER of each share is $\frac{7}{n}$ bpp in [44]. For example, ER=1.75 bpp when $n = 4$. The method [45] uses Chinese Remainder Theorem (CRT) to share the original image and secret data. Due to the additive homomorphism of CRT, data embedding on each share can be performed by DE. Thus, the ER of this method is close to 0.5 bpp. In Modified RDHSI of [49], the ER is fixed to 3 bpp. However, secret data cannot be directly extracted from the marked shares in [49]. Compared with these SS based methods, the proposed method has outstanding performance in terms of embedding capacity.

6 Conclusion

In this paper, we have proposed a novel universal RDHEI with secret sharing. Unlike the existing methods, the proposed method can perform data hiding on the diverse images, which are more suitable for cloud applications. We have designed two kinds of matrices. During data embedding, the shares of each pixel can be modified for data embedding according to the matrix. The marked shares are regarded as the unknowns. The multivariate linear equation can be constructed by mapping the coefficients between the original polynomial and reconstructed polynomial. The original shares can be recovered by solving multivariate linear equation. With the recovered shares, the original image can be recovered losslessly. Experiment results demonstrate that the proposed method outperforms most state-of-the-art secret sharing based methods in security and payload.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62162006), the Guangxi Natural Science Foundation (2025GXNSFAA069425).

References

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

[2] Y. Du, Z. Yin, and X. P. Zhang, "High capacity lossless data hiding in jpeg bitstream based on general vlc mapping," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1420–1433, 2022.

[3] M. Jana, B. Jana, and S. Joardar, " Reversible data hiding strategy exploiting circular distance interpolation utilizing optimal pixel adjustment with error substitution," *Multimedia Tools and Applications*, vol. 83, no. 16, pp. 48949–48986, 2024.

[4] J. Tian, "Reversible data embedding using a difference expansion," *IEEE transactions on circuits and systems for video technology*, vol. 13, no. 8, pp. 890–896, 2003.

[5] Z. Ni, Y. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on circuits and systems for video technology*, vol. 16, no. 3, pp. 354–362, 2006.

[6] W. Zhang, X. Hu, X. Li, and N. Yu, "Recursive histogram modification: establishing equivalency between reversible data hiding and lossless data compression," *IEEE transactions on image processing*, vol. 22, no. 7, pp. 2775–2785, 2013.

[7] W. Zhang, X. Hu, X. Li, and Y. Nenghai, "Optimal transition probability of reversible data hiding for general distortion metrics and its applications," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 294–304, 2014.

[8] K. Datta, B. Jana, and M. D. Chakraborty, " Dual image based secured reversible data hiding scheme exploiting huffman compression tree combining bit-reversal permutation technique," *Multimedia Tools and Applications*, pp. 1–46, 2024.

[9] K. Datta, B. Jana, and M. Dalui, " A Weighted Matrix-Based Reversible Data Hiding Scheme with Dual-Image by Exploiting BWT Encoding Technique," in *International Conference on Network Security and Blockchain Technology*, pp. 3–14, Springer, 2024.

[10] K. Datta, S. Banerjee, B. Jana, and M. Dalui, " A Dual-Image Based Secured Reversible Data Hiding Scheme Exploiting Weighted Matrix and Cellular Automata," in *Asian Symposium on Cellular Automata Technology*, pp. 123–136, Springer, 2024.

[11] K. Datta, B. Jana, and M. D. Chakraborty, " A cellular automata based secured reversible data hiding scheme for dual images using bit-reversal permutation technique," *Computer Standards & Interfaces*, vol. 92, p. 103919, 2025.

[12] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Transactions on image processing*, vol. 22, no. 12, pp. 5010–5021, 2013.

[13] C. Yu, X. Q. Zhang, D. Wang, and Z. Tang, "Reversible data hiding with pairwise PEE and 2D-PEH decomposition," *Signal Processing*, vol. 196, p. 108527, 2022.

[14] D. Wang, X. Q. Zhang, C. Yu, and Z. Tang, "Reversible data hiding by using adaptive pixel value prediction and adaptive embedding bin selection," *IEEE Signal Processing Letters*, vol. 26, no. 11, pp. 1713–1717, 2019.

[15] S. Weng, Y. Zhou, T. Zhang, M. Xiao, and Y. Zhao, "Reversible data hiding for jpeg images with adaptive multiple two-dimensional histogram and mapping generation," *IEEE Transactions on Multimedia*, 2023.

[16] X. Li, W. Zhang, X. Gui, and B. Yang, "Efficient reversible data hiding based on multiple histograms modification," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 2016–2027, 2015.

[17] S. Meikap and B. Jana, " Reference pixel-based reversible data hiding scheme using multi-level pixel value ordering," *Multimedia Tools and Applications*, vol. 83, no. 6, pp. 16895–16928, 2024.

[18] S. Meikap, B. Jana, and T.-C. Lu, " Context pixel-based reversible data hiding scheme using pixel value

ordering," *The Visual Computer*, vol. 40, no. 5, pp. 3529–3552, 2024.

[19] X. P. Zhang, "Reversible data hiding in encrypted image," *IEEE signal processing letters*, vol. 18, no. 4, pp. 255–258, 2011.

[20] W. Hong, T. Chen, and H. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[21] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication and Image Representation*, vol. 28, pp. 21–27, 2015.

[22] X. P. Zhang, "Separable reversible data hiding in encrypted image," *IEEE transactions on information forensics and security*, vol. 7, no. 2, pp. 826–832, 2011.

[23] Z. Qian and X. P. Zhang, "Reversible data hiding in encrypted images with distributed source encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636–646, 2015.

[24] X. Wu and W. Sun, "High-capacity reversible data hiding in encrypted images by prediction error," *Signal processing*, vol. 104, pp. 387–400, 2014.

[25] F. Huang, J. Huang, and Y. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE transactions on information forensics and security*, vol. 11, no. 12, pp. 2777–2789, 2016.

[26] Y. Fu, P. Kong, H. Yao, Z. Tang, and C. Qin, "Effective reversible data hiding in encrypted image with adaptive encoding strategy," *Information Sciences*, vol. 494, pp. 21–36, 2019.

[27] X. Wang, C. Chang, and C. Lin, "Reversible data hiding in encrypted images with block-based adaptive msb encoding," *Information Sciences*, vol. 567, pp. 375–394, 2021.

[28] Z. Tang, S. Xu, H. Yao, C. Qin, and X. Q. Zhang, "Reversible data hiding with differential compression in encrypted image," *Multimedia Tools and Applications*, vol. 78, no. 8, pp. 9691–9715, 2019.

[29] Z. Liu and C. Pun, "Reversible data-hiding in encrypted images by redundant space transfer," *Information Sciences*, vol. 433, pp. 188–203, 2018.

[30] C. Qin, X. Qian, W. Hong, and X. P. Zhang, "An efficient coding scheme for reversible data hiding in encrypted image with redundancy transfer," *Information Sciences*, vol. 487, pp. 176–192, 2019.

[31] S. Yi and Y. Zhou, "Separable and reversible data hiding in encrypted images using parametric binary tree labeling," *IEEE Transactions on Multimedia*, vol. 21, no. 1, pp. 51–64, 2019.

[32] C. Yu, X. Q. Zhang, G. Li, S. Zhan, and Z. Tang, "Reversible data hiding with adaptive difference recovery for encrypted images," *Information Sciences*, vol. 584, pp. 89–110, 2022.

[33] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on information forensics and security*, vol. 8, no. 3, pp. 553–562, 2013.

[34] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, "High capacity reversible data hiding in encrypted images by patch-level sparse representation," *IEEE transactions on cybernetics*, vol. 46, no. 5, pp. 1132–1143, 2016.

[35] S. Yi and Y. Zhou, "Binary-block embedding for reversible data hiding in encrypted images," *Signal Processing*, vol. 133, pp. 40–51, 2017.

[36] K. Chen and C. Chang, "High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 334–344, 2019.

[37] Z. Yin, Y. Xiang, and X. P. Zhang, "Reversible data hiding in encrypted images based on multi-MSB prediction and huffman coding," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 874–884, 2020.

[38] C. Yu, X. Q. Zhang, X. P. Zhang, G. Li, and Z. Tang, "Reversible data hiding with hierarchical embedding for encrypted images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 2, pp. 451–466, 2022.

[39] S. Xu, J. Horng, C. Chang, and C. Chang, "Reversible data hiding with hierarchical block variable length coding for cloud security," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, 2022.

[40] C.-C. Chang, J.-C. Liu, and K. Gao, "Cryptanalysis of iterative encryption and image sharing scheme based on the vq attack," *Journal of Visual Communication and Image Representation*, vol. 97, p. 103973, 2023.

[41] X. Wu, J. Weng, and W. Yan, "Adopting secret sharing for reversible data hiding in encrypted images," *Signal Processing*, vol. 143, pp. 269–281, 2018.

[42] Y. Chen, T. Hung, S. Hsieh, and C. Shiu, "A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3332–3343, 2019.

[43] C. Qin, C. Jiang, Q. Mo, H. Yao, and C. Chang, "Reversible data hiding in encrypted image via secret sharing based on GF (p) and GF (2^8)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 1928–1941, 2022.

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

[45] Y. Ke, M. Zhang, X. P. Zhang, J. Liu, T. Su, and X. Yang, "A reversible data hiding scheme in encrypted domain for secret image sharing based on Chinese Remainder Theorem," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2469–2481, 2022.

[46] Z. Hua, Y. Wang, S. Yi, Y. Zhou, and X. Jia, "Reversible data hiding in encrypted images using cipher-feedback secret sharing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 4968–4982, 2022.

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

[48] Z. Hua, X. Liu, Y. Zheng, S. Yi, and Y. Zhang,

"Reversible data hiding over encrypted images via preprocessing-free matrix secret sharing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 3, pp. 1799–1814, 2024.

[49] L. Xiong, X. Han, C.-N. Yang, and X. Zhang, "Reversible data hiding in shared images based on syndrome decoding and homomorphism," *IEEE Transactions on Cloud Computing*, vol. 11, no. 3, pp. 3085–3098, 2023.

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

[51] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE transactions on information theory*, vol. 29, no. 2, pp. 208–210, 1983.

[52] 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.

[53] P. Bas, T. Filler, and T. Pevnỳ, "Break our steganographic system: the ins and outs of organizing boss," in *Information Hiding: 13th International Conference, IH 2011, Prague, Czech Republic, May 18-20, 2011, Revised Selected Papers 13*, pp. 59–70, Springer, 2011.

[54] A. G. NCU, "Bows2," *Mendeley Data*, vol. 1, 2023.