mathematics

MDPI

Article

A Hierarchical Authorization Reversible Data Hiding in Encrypted Image Based on Secret Sharing

Chao Jiang [1,2], Minqing Zhang [1,2,*], Yongjun Kong [1,2], Zongbao Jiang [1,2] and Fuqiang Di [1,2,*]

[1] College of Cryptography Engineering, Engineering University of PAP, Xi'an 710018, China; coo1@hhhrsc.com (C.J.); coo2@hhhrsc.com (Y.K.); coo3@hhhrsc.com (Z.J.)
[2] Key Laboratory of PAP for Cryptology and Information Security, Xi'an 710018, China
* Correspondence: h4052916@hhhrsc.com (M.Z.); coo4@hhhrsc.com (F.D.)

Abstract: In the current distributed environment, reversible data hiding in encrypted domain (RDH-ED) cannot grant corresponding privileges according to users' identity classes. To address this issue, this paper proposes a hierarchical authorization structure embedding scheme based on secret image sharing (SIS) and users' hierarchical identities. In the first embedding, the polynomial coefficient redundancy generated in the encryption process of the SIS is utilized by the image owner. For the second, the participants are categorized into two parts. One is core users with adaptive difference reservation embedding, and the other is ordinary users with pixel bit replacement embedding. At the time of reconstruction, more than one core user must provide pixel differences, which grants more privileges to core users. The experimental results demonstrate that the average embedding rate (ER) of the test images is 4.3333 bits per pixel (bpp) in the (3, 4) threshold scheme. Additionally, the reconstructed image achieves a PSNR of $+\infty$ and an SSIM of 1. Compared to existing high-performance RDH-ED schemes based on secret sharing, the proposed scheme with a larger ER maintains strong security and reversibility. Moreover, it is also suitable for multiple embeddings involving multilevel participant identities. In conclusion, the results underscore the efficacy of our technique in achieving both security and performance objectives within a complex distributed setting.

Keywords: reversible data hiding; encrypted image; secret sharing; multilevel identities; hierarchical authorization structure

MSC: 68U10

Citation: Jiang, C.; Zhang, M.; Kong, Y.; Jiang, Z.; Di, F. A Hierarchical Authorization Reversible Data Hiding in Encrypted Image Based on Secret Sharing. *Mathematics* **2024**, *12*, 2262. https://doi.org/10.3390/math12142262

Academic Editors: Yi-Hui Chen and Ya-Fen Chang

Received: 14 June 2024
Revised: 16 July 2024
Accepted: 17 July 2024
Published: 19 July 2024

1. Introduction

Reversible data hiding in encrypted domain [1–4] (RDH-ED) allows for the embedding of additional data into encrypted carriers. RDH-ED has a wide range of applications in covert communication [5], copyright management [6], etc., and thus has attracted the extensive attention of researchers. According to the different sources of redundancy, RDH-ED can be divided into three types: Vacating Room after Encryption (VRAE), Vacating Room before Encryption (VRBE), and Vacating Redundancy in Encryption (VRIE). The VRAE schemes [7–11] embed additional data by modifying encrypted pixels. Since the correlation of the ciphertext is weak, the Shannon entropy is maximized, and the ER is limited. VRBE schemes [12–16] usually embed additional data into the redundancy generated by the correlation of pixels after prediction, compression, and coding. The ER is larger, but the preprocessing is so complicated that the application scenarios are limited. To solve the above problems, Ke et al. first proposed VRIE schemes [17,18], which applied learning with errors (LWEs) by quantifying the encrypted space and utilizing redundancy in the encryption process. VRIE schemes can integrate data hiding and cryptography organically, so that better security, reversibility, and ER can be obtained. However, cryptographic schemes for VRIE are too difficult.

With the wide application of cloud environments [19], designing RDH-ED for distributed scenarios has become a popular research topic. The (k, n) threshold function of secret sharing [20] (SS) increases its suitability. In the (k, n) scheme, the secret owner splits secrets into multiple secret shares and distributes them to n different users for management. Then, the receiver collects at least k shares at random to reconstruct the secret. Most existing secret-sharing-class RDH-ED schemes [4,21–26] have been developed assuming that all participants hold equal weight and that every shared image possesses identical data hiding capabilities. Nevertheless, in real distributed settings, participants typically vary in their levels of authority, comprising senior executives and regular employees, each with distinct rights and duties. Consequently, the requirements for secret extraction should align with the corresponding privileges and necessities. For instance, within a bank environment, there exist senior managers and ordinary employees with varying privileges, where managers hold greater rights than regular employees. This rigid authorization structure proves inadequate for the dynamic and adaptable demands of multiparty computing scenarios. Therefore, there is a need to devise a hierarchical authorization framework that can cater to the flexible and evolving requirements of applications. Consequently, this paper proposes a hierarchical embedding scheme, which divides all participants into core users and ordinary users, both of which use different embedding methods. In the process of secret extraction and image reconstruction, a certain number of core users must be involved, so the core users are given greater privileges. The major contributions of the proposed scheme are as follows:

1. We propose a hierarchical authorization structure RDH-ED for secret sharing through difference preservation. This innovative approach involves classifying participants' identities into different levels, employing diverse embedding methods, necessitating core users' participation in secret reconstruction, and granting core users enhanced privileges. These enhancements effectively meet the hierarchical authorization demands of RDH-ED in distributed scenarios.
2. A polynomial embedding algorithm is developed for embedding copyright information into polynomial coefficients during secret distribution by the image owner. This functionality enables the authentication of the carrier image, thereby bolstering security and traceability.
3. We propose a hierarchical embedding algorithm that segregates participants into core users and ordinary users. Core users utilize adaptive difference reservation embedding, while ordinary users employ pixel bit replacement embedding. This strategy ensures that core users play a pivotal role in secret recovery during reconstruction, elevating their authorization levels.
4. Compared with the existing superior RDH-ED, our scheme boasts a higher embedding rate, lower expansion rate, enhanced security, and superior reversibility, showing significant performance enhancements compared to current solutions.

The remainder of this paper is as follows: In Section 2, we introduce the existing pixel difference preservation SIS scheme. Then, we describe the proposed RDHEI-ED scheme based on SIS in detail and provide a specific example for understanding. Section 4 demonstrates the superior performance of the many experiments. Finally, this paper concludes with a summary.

2. Related Works

2.1. Shamir's (k, n) Secret Sharing

Wu et al. first proposed an RDH-ED scheme [4] based on SS, which made full use of the threshold function. Priyanka et al. [21] used SS to embed extra data into color images. Soon after, Ke et al. [22] proposed an RDH-ED scheme based on the Chinese Remainder Theorem (CRT) and SIS, which obtains better separability, but its embedding rate is low. To solve the problem of the lack of diffusion characteristics in SISs, Hua et al. introduced two RDH-ED schemes [23,24] based on Cipher-Feedback Secret Sharing (CFSS), which solves the problem that the traditional secret-sharing-class RDH-ED cannot resist score reduction

attacks, thus improving security. Yu et al. [25] introduce an innovative approach utilizing secret sharing and hybrid encoding. Through the development of an iterative encryption method specific to block-based encryption, the spatial correlation of the original block is meticulously preserved within the encrypted block. This enhancement significantly boosts both the embedding rate and security levels of the system. Qin et al. [26] proposed the RDH-ED scheme on the Galois Field (GF) (p) and GF (2^8) based on SIS with difference preservation, which retained the same correlation between the original pixels and encrypted pixels. However, the participants needed to preselect thresholds, and abundant blocks were unembedded, so the ER and application scenarios were limited.

2.2. SIS Based on Pixel Difference Preservation

Shamir first proposed a (k, n) secret sharing scheme [20] based on Lagrange interpolating polynomials.

Theorem 1. *For any k different* $(x_i, F(x_i))$, $i = 1, 2, \ldots, n$, *can be used to reconstruct a* $k-1$ *degree polynomial uniquely via Equation (1):*

$$F(x) = \sum_{i=1}^{k} F(x_i) \prod_{j=1, j\neq i}^{k} \frac{x - x_j}{x_i - x_j} \tag{1}$$

The secret owner constructs Equation (2):

$$F(x) = s + C_1 x + C_2 x^2 + \cdots + C_{k-1} x^{k-1} \tag{2}$$

where s is the secret, and $C_1, C_2, \ldots, C_{k-1}$ are random numbers. The owner calculates shares $f_i = F(i)$, and distributes them to n different users P_i. By collecting at least k shares at random, we can reconstruct $F(x)$. Thus, all the coefficients of $F(x)$ can be reconstructed.

In order to keep the differences the same as possible between the original and encrypted pixels, Qin et al. proposed an SIS scheme [26] based on pixel difference preservation. Let the size of the image block be 2×2, the pixels be p_1, p_2, p_3, and p_4, and the threshold be (k, n). Equation (3) can be constructed as follows:

$$\begin{cases} g_1(x) = p_1 + h_1 x + h_2 x^2 + \ldots + h_{k-1} x^{k-1} \bmod 251 \\ g_z(x) = p_z + h_1 x + h_2 x^2 + \ldots + h_{k-1} x^{k-1} \bmod 251 \end{cases} \tag{3}$$

where z = 2, 3, 4, $h_1, h_2, \ldots, h_{k-1}$ are random numbers. Equation (4) can be obtained as follows:

$$\begin{cases} g_1(x) + 251 r_1 = p_1 + h_1 x + h_2 x^2 + \ldots + h_{k-1} x^{k-1} \\ g_z(x) + 251 r_z = p_z + h_1 x + h_2 x^2 + \ldots + h_{k-1} x^{k-1} \end{cases} \tag{4}$$

where r_1 and r_z are nonnegative integers; thus,

$$\Delta = g_1(x) - g_z(x) = (g_1(x) + 251 r_1) \bmod 251 - (g_z(x) + 251 r_z) \bmod 251 \tag{5}$$

Mostly because of the high similarity between pixels in an image block, thus $|g_1(x) - g_z(x)| < 128$, $r_1 = r_z$, and $p_1 - p_z = g_1(x) + 251 r_1 - g_z(x) - 251 r_z = g_1(x) - g_z(x) = \Delta$. However, when $|g_1(x) - g_z(x)| \geq 128$, $r_1 - r_z = \pm 1$. When $r_1 - r_z = 1$, $p_1 - p_z = g_1(x) + 251 r_1 - g_z(x) + 251 r_z = \Delta + 251$; when $r_1 - r_z = -1$, $p_1 - p_z = g_1(x) + 251 r_1 - g_z(x) + 251 r_z = \Delta - 251$. Thus, Equation (6) can be obtained as follows:

$$p_1 - p_z = \begin{cases} g_1(x) - g_z(x), \; if \; |g_1(x) - g_z(x)| < 128, \\ g_1(x) - g_z(x) + 251, \; if \; g_1(x) - g_z(x) \leq -128, \\ g_1(x) - g_z(x) - 251, \; if \; g_1(x) - g_z(x) \geq 128. \end{cases} \tag{6}$$

Therefore, it can be determined whether "$g_1(x) - g_z(x)$" is equal to "$p_1 - p_z$" based on $|g_1(x) - g_z(x)|$. This scheme ensures complete consistency in pixel correlation between the encrypted image and the original image. By leveraging this scheme to implement RDH-ED, it can effectively address the limitations of pixel correlation destruction in the ciphertext, which typically results in low embedding rates. Moreover, this scheme exhibits minimal construction complexity and offers robust operability.

3. Proposed RDHEI-ED Scheme

3.1. The Procedures of Hierarchical Authorization Structure RDH-ED

Data transmission in cloud environments is fraught with security risks, increasing the susceptibility to attacks. It is imperative to safeguard both the data and transmission processes to enhance security measures. Consequently, data concealment within carriers is essential for secure transmission. Furthermore, image owners seek to ensure the integrity of carrier images, necessitating the effective protection of these images through embedded authentication mechanisms. The procedures of the proposed scheme are shown in Figure 1. The content owner distributes the carrier image to multiple participants via SIS based on pixel difference preservation. At the time of secret sharing, the copyright information of the image can be embedded into the polynomial coefficients. The participants are categorized into core users and ordinary users based on their status level. Core users usually play more important roles than ordinary users, so they need to be given more rights. Therefore, this section designs a hierarchical authorization structure based on identity levels. As illustrated in Figure 1, core users employ the difference preservation embedding algorithm, while ordinary users utilize the pixel replacement embedding algorithm. During the image reconstruction and secret extraction phase, core users are indispensable, granting them greater authority within the process.

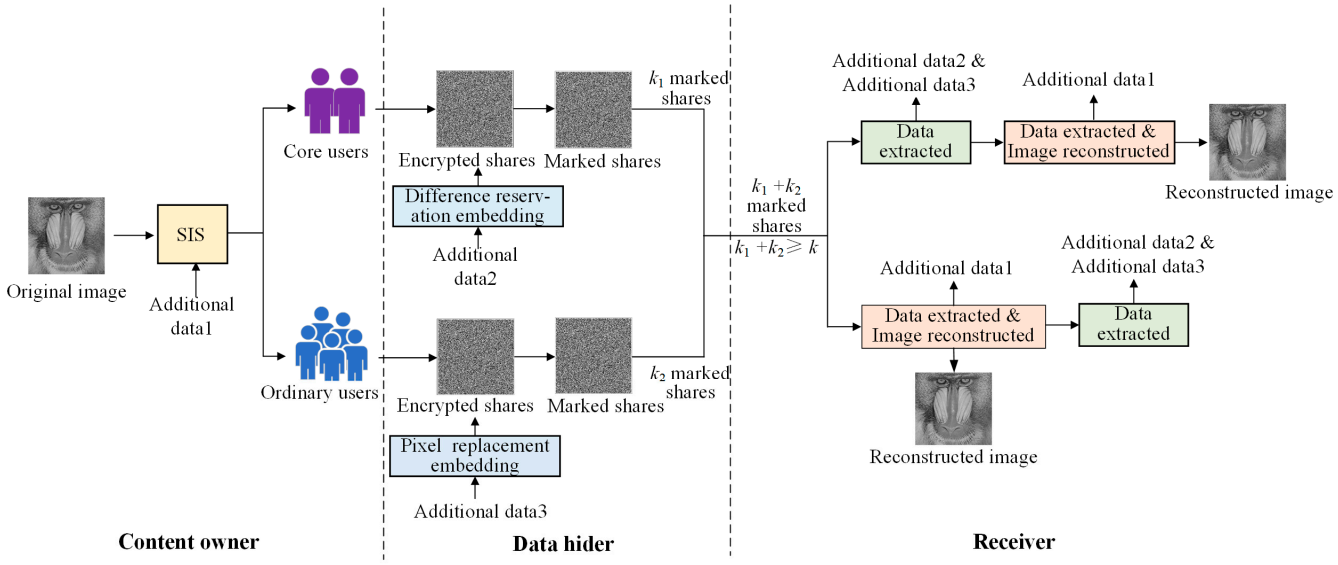

Figure 1. The procedures of the proposed RDH-ED schemes.

In simple terms, an image block is chosen for description. As shown in Figure 2, core users must retain the differences in the original pixels to perform accurate reconstruction. To improve the ER, adaptive difference coding embedding is adopted to increase the number of embeddable blocks. Ordinary users replace the zth ($z = 2, 3, 4$) pixel with additional data directly. After extracting the secrets, the first original pixel can be reconstructed from the k shares. Then, the zth pixel can be obtained based on the differences provided by the core users. The process of secret sharing and data embedding is shown in Algorithm 1.

Algorithm 1: Secret sharing and data embedding

Input: original image, k_1, k, n

1: Split the image into blocks of size 2 × 2;
2: Scramble image blocks using scrambled key;
3: Generate session key using users' identity numbers and pseudorandom number generator;
4: Scan the image blocks to obtain the size M;
5: $l \leftarrow 1$;
6: **while** $l < M$ do
7: Scan image block to obtain pixels a, b, c, d;
8: Encrypt extra secrets using AES;
9: Construct the polynomial $f(x)$;
10: Calculate $f(x)$ by substituting session key as x to generate then distribute shares to participant P;
11: **if** P == core users
12: Implementing difference preserving embedding;
13: **else**
14: Implementing pixel bit replacement embedding;
15: **end if**
16: $l \leftarrow l + 1$;
17: **end while**

Output: Share images with extra secrets embedded

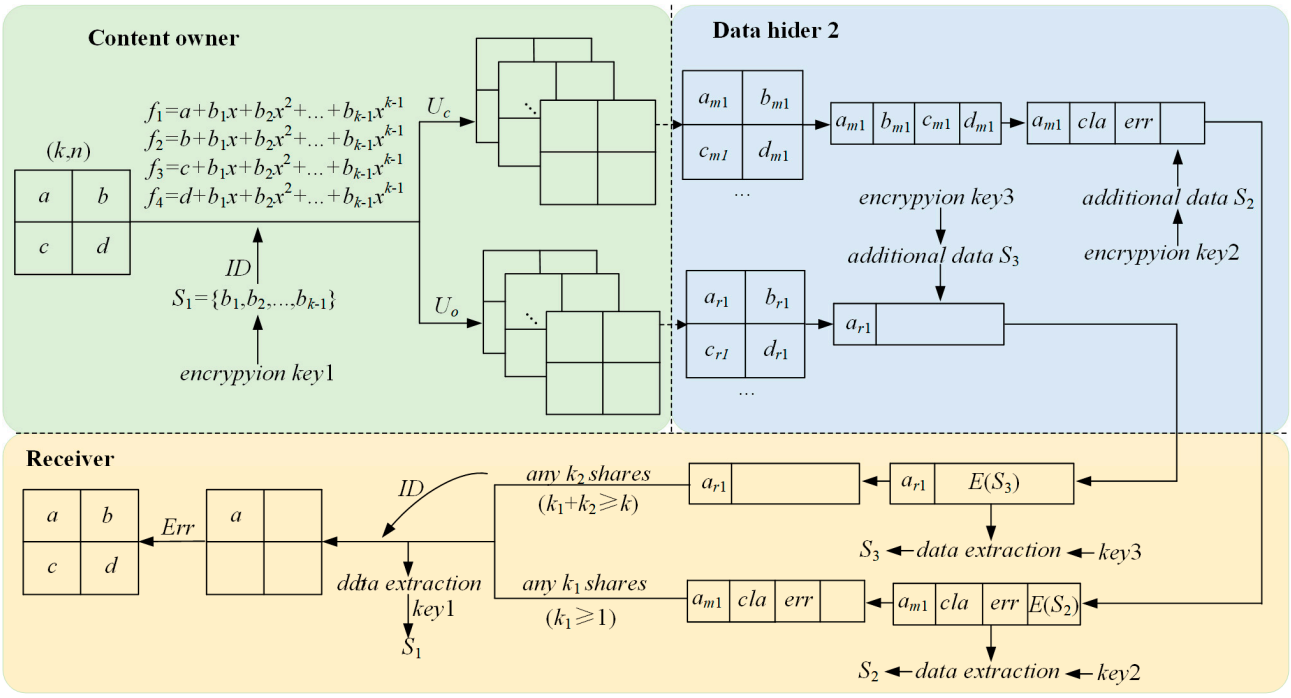

Figure 2. Framework description of an image block.

3.2. Image Preprocessing

To maintain high security, firstly, 512 × 512 grayscale images are divided into nonoverlapping blocks with a size of 2 × 2. Then, all the blocks and pixels in each block are scrambled. All the blocks are scrambled with the key "Scrkey1", and the pixels in each block are scrambled with the key "Scrkey2".

3.3. Session Key Generation

According to the n_1 core users' (U_c) identity numbers $i_1\{i_1 = 1, 2, \ldots, n_1\}$, the n_2 ordinary users' (U_g) identity numbers $i_2\{i_2 = n_1 + 1, n_1 + 2, \ldots, n_1 + n_2\}$, and the seed key h_i, a

pseudorandom number generator is utilized to generate n_1 flag numbers $ID_{cor} = \{id_1, id_2, \ldots, id_{n_1}\}$ and n_2 flag numbers $ID_{gen} = \{id_{n_1+1}, id_{n_1+2}, \ldots, id_{n_1+n_2}\}$ as session keys, where $id_i \neq id_j$.

3.4. Encrypted Image Generation

SIS is used to embed the authentication to carriers. Let the image block pixels be a, b, c, d after block and pixel scrambling, the first embedded secret $S_1 = \{0, 1\}^N$.

3.4.1. Polynomial Construction

S_1 are encrypted with AES by the encryption key "key1" and then every 8 bits of encrypted S_1 are converted to a decimal to generate $E(S_1) = \{b_1, b_2, \ldots, b_{k-1}\}$. Then, Equation (7) is constructed on GF (251).

$$\begin{cases} f_1(x) = a + b_1 x + b_2 x^2 + \ldots + b_{k-1} x^{k-1} \bmod 251 \\ f_2(x) = b + b_1 x + b_2 x^2 + \ldots + b_{k-1} x^{k-1} \bmod 251 \\ f_3(x) = c + b_1 x + b_2 x^2 + \ldots + b_{k-1} x^{k-1} \bmod 251 \\ f_4(x) = d + b_1 x + b_2 x^2 + \ldots + b_{k-1} x^{k-1} \bmod 251 \end{cases} \tag{7}$$

3.4.2. Secret Share Generation

For core users, substitute $x_i \in ID_{cor} = \{id_1, id_2, \ldots, id_{n_1}\}$ into Equation (7) to compute n_1 shares, and then distribute them to corresponding users.

For ordinary users, substitute $x_i \in ID_{gen} = \{id_{n_1+1}, id_{n_1+2}, \ldots, id_{n_1+n_2}\}$ into Equation (7) to compute n_2 shares and distribute them to corresponding users.

Since this process is carried out on GF (251), original pixels greater than 250 are directly replaced with 250. To accurately reconstruct the original image, pixels exceeding 250 need to be labeled. Given the rarity of pixels above 250 in grayscale images, we assign values of 1, 2, 3, 4, and 5 to represent the positions of pixels 251, 252, 253, 254, and 255, respectively, while using 0 to denote the positions of other pixels. This approach generates a location map of the carrier image. With the majority of the location map populated by 0 values, we employ an arithmetic coding algorithm to compress it. The compressed location map and length can be embedded as payload_1 into the polynomial coefficients.

3.5. Marked Image Generation after Data Embedding

3.5.1. Difference Reservation Embedding

The embedding rules of the core users are shown in Figure 3 and Table 1. d_z is the difference between the zth pixel and the first pixel, and $|d|_{max}$ is the maximum of the absolute value of d_z. According to Equation (6), due to the strong correlation in the 2×2 image blocks, $|d|_{max}$ is less than 64 in most of the blocks. Therefore, $|d|_{max}$ can be classified into seven different levels. For the blocks of $|d|_{max} \leq 63$, $|d|_{max}$ can be encoded by 3 bits. dc_{max} is the corresponding encoding of $|d|_{max}$. Since d_z may be positive or negative, "1" is used to represent a positive value, and "0" is negative. According to Table 1, when $|d|_{max} \in [32, 63]$, the block is unembeddable, and can be labeled according to Table 1 without risking pixel overflow. When $|d|_{max} \geq 64$, pixel overflow will occur. Consequently, for these pixel blocks of $|d|_{max} \geq 64$, we preserve the pixels unchanged. Additionally, a location map is employed to distinguish blocks using 1 for embeddable blocks and 0 for others. This location map undergoes compression through arithmetic coding, with the compressed positional map and its length embedded into the embeddable pixels as payload_2.

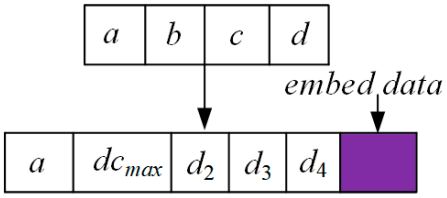

Figure 3. Embedding rules of core users.

Table 1. Coding rules of core users.

$\|d\|_{max}$	dc_{max}	$\|d\|$ Coding Bits	Data Hiding Bits
[0, 1]	000	1	15
[2, 3]	001	2	12
[4, 7]	010	3	9
[8, 15]	011	4	6
[16, 31]	100	5	3
[32, 63]	101	6	0

3.5.2. Pixel Bit Replacement Embedding

Every 8 bits of the secret $S_3 = \{0,1\}^N$ are converted to a decimal with AES encrypted by the encryption key "key3" to generate E(S_3). For each block share, the first pixel is kept as the label pixel, and the other three pixels are replaced with E(S_3).

3.6. Data Extraction and Image Recovery

The process of data extraction and image recovery is shown in Algorithm 2. The receiver collects any more than or equal to k_1 core shares and k_2 ordinary shares, where $k_1 + k_2 \geq k$, and $k_1 \geq 1$.

For core users, payload_2 should be extracted first. For the blocks of $|d|_{max} \leq 63$, the four pixels are converted to binary values; then, the corresponding differences can be extracted according to the difference classes, and additional data can be obtained. After decryption, the secrets can be obtained.

For ordinary users, the last three pixels of each block can be extracted directly, and the secrets can be obtained after decryption.

For the image owner, the label pixel could be recovered by the Lagrange interpolation formula since the $k_1 + k_2$ users could provide k label pixel shares, and E(S_1) could be extracted. After decryption, S_1 can be obtained. Then, all the original pixels could be recovered by the pixel differences provided by the core users and the payload_1.

Algorithm 2: Secret extraction and image reconstruction

Input: Share images with extra secrets embedded, k_1, k, n
1: $l \leftarrow 1$;
2: **while** $l < M$ do
3: **if** P == core users
4: Obtain the first pixel and the three differences;
5: Extract the extra secret;
6: **else**
7: Obtain the first pixel;
8: Extract the extra secret;
9: Recovering the first original pixel using the first share pixel by Theorem 1;
10: Use the three differences and the first original pixel to obtain the original three pixels.
11: **end if**
12: $l \leftarrow l + 1$;
13: **end while**
Output: Extra secret, reconstructed image

3.7. Instance of the Proposed Scheme

For clarity, an example of a core user is shown in Figure 4. Let the marked pixels be $\{65, 70, 66, 63\}$. The differences between the zth and the first pixel are $+5$, $+1$, and -2, respectively. Because the maximum absolute value is 5, the corresponding class code is 010, and the corresponding difference codes are 1101, 1001, and 0010. Therefore, the embeddable space is 9 bits. After embedding "101010101", the marked pixels are $\{65, 91, 37, 85\}$. The receiver first converts the data to binary values and subsequently calculates the three differences according to the class. Therefore, the secrets, labeled pixels, and differences can be obtained.

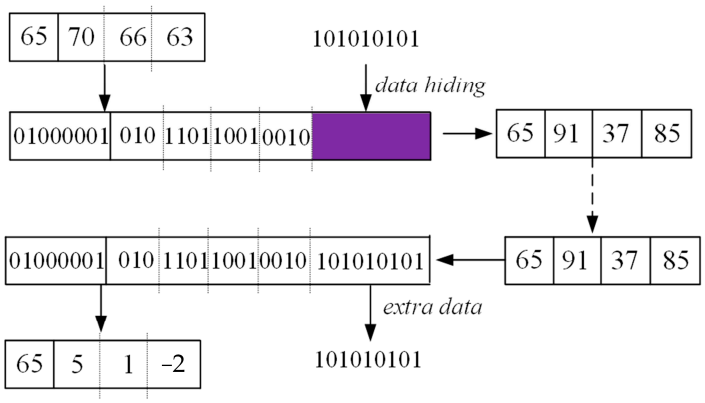

Figure 4. An instance of the proposed scheme.

4. Experimental Results and Comparisons

The experiments were implemented in MATLAB R2021b with a CPU (i7-11800H) @ 2.30 GHz. The test data are shown in Figure 5, and all the images are 512×512 8-bit grayscale images named Jetplane, Peppers, Goldhill, Baboon, Boats, and Airplane.

Figure 5. Test images.

4.1. Security Analysis

The parameter settings are $k = 3$, $n = 4$, and $k_1 = 1$. The encrypted image mentioned in this section is the first secret share, and the marked image is the first secret share after embedding the extra data again.

4.1.1. Keyspace

When the key space is large enough, it can effectively resist attacks such as brute force cracking. In the proposed scheme, the original image is 512×512, and each block is 2×2. Thus, the total number of blocks is 65,536, and possible situations for block scrambling are 65536!. Because each image block has four pixels, there are 4! cases of pixel scrambling. Therefore, the whole key space of block and pixel scrambling is $65536! \times 4!^{65536}$, which is much larger than 2^{100}, so it can efficiently resist brute force attacks.

4.1.2. Histogram

A more uniform histogram corresponds to a better security scheme and can resist statistical analysis attacks more effectively. Figure 6a–d show the original image, encrypted image, marked image, and reconstructed image of Baboon. (e)–(h), (i)–(l), and (m)–(p)

are the corresponding plane histogram, scatter histogram, and 3D histogram, respectively. As shown in Figure 6, the histograms of the encrypted image and marked image are uniform and gentle, and the distributions of the original image and reconstructed image are the same. Our experimental verification reveals that the histograms of labeled images display lower homogeneity than those of encrypted images. This disparity is attributed to the embedded secrets not being true random numbers, influencing the distribution of histograms to a certain degree. However, the histograms of the marked images exhibit significant homogeneity. This signifies that while the embedding process does have an impact on security, the extent of this impact remains within acceptable limits. Due to the encryption of the coefficients used in constructing the polynomials, the encrypted data exhibit a high level of uncorrelation. Furthermore, these encrypted data are distributed to participants through secret sharing, providing an additional layer of double encryption protection. These measures ultimately contribute to achieving a uniform histogram distribution. Therefore, attackers cannot obtain any relevant information from the pixel distributions.

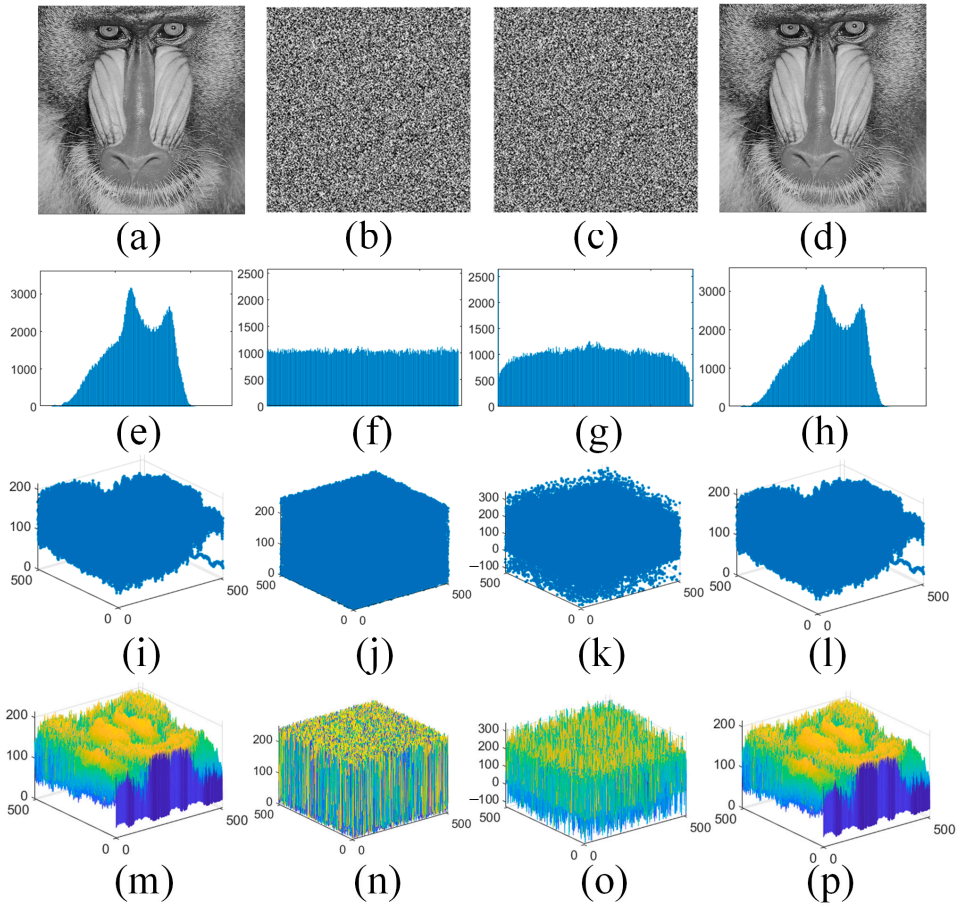

Figure 6. Histograms of Baboon. (**a**) Original image; (**b**) encrypted image; (**c**) marked image; (**d**) reconstructed image; (**e**–**h**) plane histogram; (**i**–**l**) scatter plot histogram; (**m**–**p**) 3D histogram.

4.1.3. Relevant Parameters

1. Shannon Entropy

The more uniformly the pixels are distributed, the closer to 8 the entropy is. The calculation is shown in Equation (8).

$$H(X) = -\Sigma\, P(x)\, log_2 P(x) \tag{8}$$

where $H(X)$ denotes the entropy of random X, and $P(x)$ is the probability of x. If each pixel in the image has equal probability, the entropy could reach a maximum of 8. The entropies of the different images are shown in Figure 7, from which it can be seen that the entropies of encrypted shares and marked shares are both close to 8, and the entropy of the proposed scheme is much closer to 8 than that of Qin et al.'s scheme [26], which indicates that the proposed scheme can resist entropy analysis attacks more effectively.

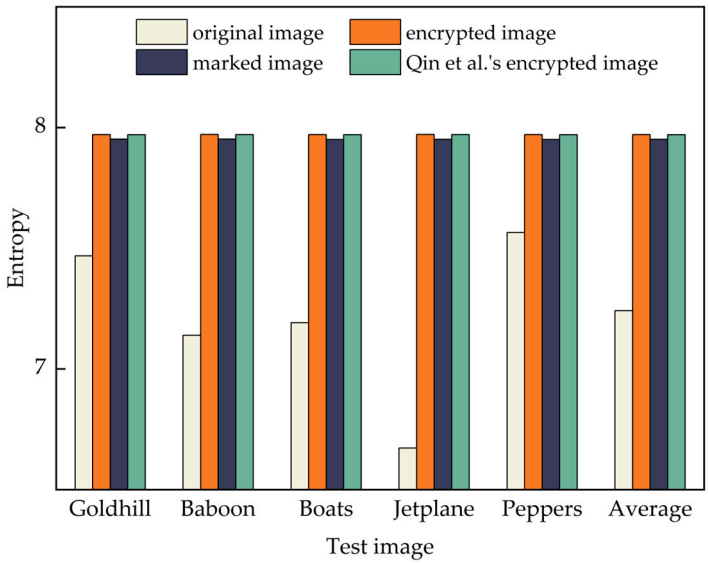

Figure 7. Shannon entropy of different images [26].

2. PSNR and SSIM

The larger the peak signal-to-noise ratio (PSNR) is, the less distorted the image will be. When $PSNR > 35$ dB, we cannot detect any distortion; when $PSNR < 10$ dB, there is a large difference between the contrast images. The closer to 1 the structural similarity (SSIM) is, the more similar the contrastive images are. An SSIM value of 1 indicates that two images are the same. The PSNR and SSIM can be calculated as follows:

$$MSE = \frac{1}{X \times Y} \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} (P(i,j) - P'(i,j)) \tag{9}$$

$$PSNR = 10 \log_{10} \frac{255}{MSE} \tag{10}$$

$$SSIM(X,Y) = \frac{(2M_X M_Y + Cons1)(2C_{XY} + Cons2)}{(M_X^2 + M_Y^2 + Cons1)(C_X^2 + C_Y^2 + Cons2)} \tag{11}$$

In the context provided, the original image X is compared with image Y using specific pixels $P(i, j)$ and $P'(i, j)$. The mean values of X and Y are denoted as M_X and M_Y, respectively, while the standard deviations of X and Y are represented by C_X and C_Y, respectively. The covariance of X and Y is indicated as C_{XY}. *Cons1* and *Cons2* are two small numbers to avoid division by 0. The six images in Figure 4 were tested, and the results are shown in Figure 8. The average PSNR and SSIM values of the encrypted image and the original image are close to 4.7988 and 0.0154, respectively, and the average values of the marked image and original image are close to 7.5858 and 0.0243, respectively. Smaller PSNR and SSIM values indicate large differences between the encrypted image, marked image, and original image. Thus, the proposed scheme has strong security.

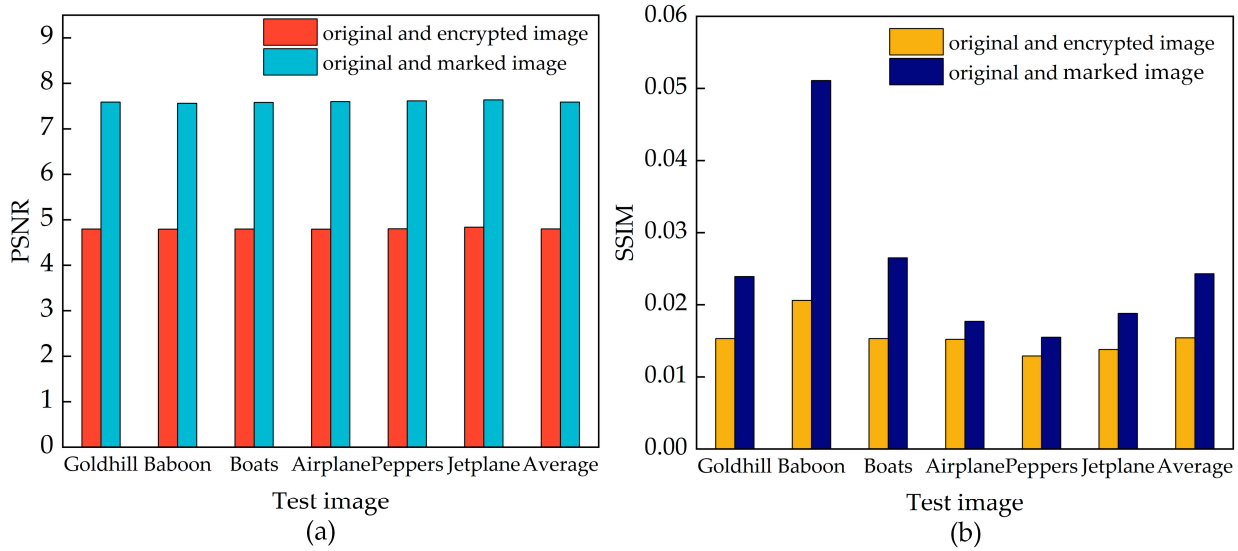

Figure 8. PSNR and SSIM of different images. (**a**) PSNR of different images; (**b**) SSIM of different images.

4.1.4. Pixel Correlation

The smaller the pixel correlation of the encrypted image is, the more secure the scheme. Experiments tested the correlation of Baboon's first encrypted image and marked image from the horizontal, vertical, 45°, and 135° perspectives. Figure 9 shows the pixel correlation of the first encrypted image and Figure 10 shows the pixel correlation of the first marked image. As shown in Figures 7 and 8, the pixel correlation of both the encrypted and marked images is close to 0, which indicates that the proposed scheme has high security.

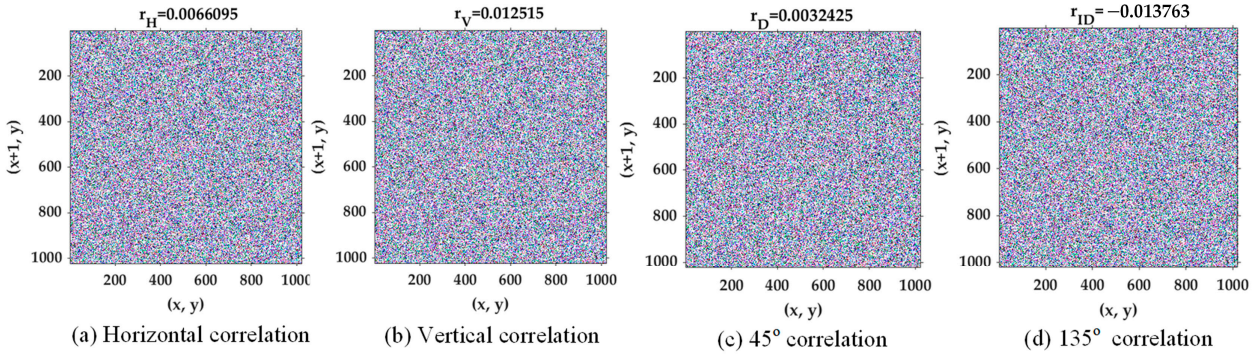

Figure 9. The pixel correlation of the first encrypted image.

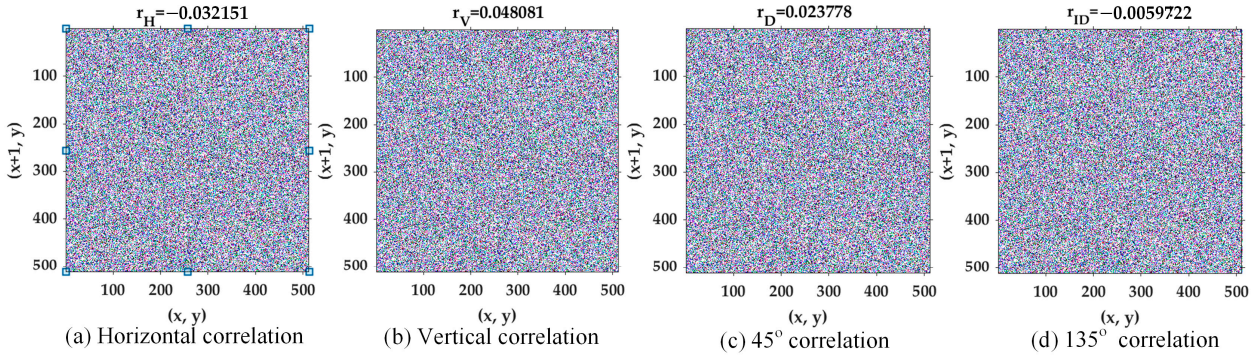

Figure 10. The pixel correlation of the first marked image.

4.2. Embedding Rate

The embedding capacity (EC) is the total bits of additional data embedded in the carrier. The embedding rate (ER) is the average number of bits embedded per pixel. The ER of the proposed scheme consists of the polynomial embedding rate (ER_1), core user embedding rate (ER_2), and ordinary user embedding rate (ER_3). The payload_1 and payload_2 must be embedded into the carrier to determine the reversibility. Therefore, the total payload denoted as payload_total can be calculated as $payload\_total = payload\_1 + payload\_2$. If we denote the embedded capacity of the core user as EC_2, as the number of encrypted pixels increases to $512 \times 512 \times n$ after SIS, the ER can be calculated as follows:

$$ER = ER_1 + ER_2 + ER_3 = \frac{EC - payload\_total}{512 \times 512 \times n} \tag{12}$$

$$ER_1 = \frac{8 \times (k-1) \times 512 \times 512 - payload\_1}{2 \times 2 \times 512 \times 512 \times n} \tag{13}$$

$$ER_2 = \frac{EC_2 - payload\_2}{2 \times 2 \times 512 \times 512 \times n} \tag{14}$$

$$ER_3 = \frac{24 \times (k - k_1) \times 512 \times 512}{2 \times 2 \times 512 \times 512 \times n} \tag{15}$$

Firstly, the six images in Figure 4 are tested at different thresholds, and the results are shown in Table 2. For $k = n$, the ER is larger, but the threshold function is not used. For common thresholds (3, 4) and (4, 5), the proposed scheme also has an excellent embedding performance. To illustrate the superiority of the proposed scheme, it is compared with current similar perfect schemes, and the results are shown in Figure 11 and Table 3. The average ER of the proposed scheme is maximized at six different thresholds. The comparison schemes are all based on SIS. Wu et al. proposed two schemes [4], and the one with the larger ER was chosen for comparison. Chen et al.'s scheme [27] has an ER of $7/n$, which decreases with the increasing n. Hua et al. proposed the CFSS RDH-ED [23] and MSS RDH-ED [24], which embed extra data by predicting encrypted image pixels. Because the correlation of encrypted pixels is small, the ER is limited. Qin et al.'s scheme [26] utilized difference protection SIS to maintain the same correlation between encrypted pixels and original pixels. The threshold needs to be preselected, and many unembeddable blocks exist, which leads to a good embedding performance.

Table 2. ER of different images at different thresholds.

Maximum Embedding Rate (bpp)	$k = 2$		$k = 3$		$k = 4$	
	$n = 2$	$n = 3$	$n = 3$	$n = 4$	$n = 4$	$n = 5$
Goldhill	4.2532	2.9202	5.5865	4.2534	6.2531	5.0532
Baboon	4.1226	2.7897	5.4561	4.1228	6.1227	4.9227
Boats	4.1956	2.8664	5.5288	4.1954	6.1954	4.9951
Airplane	4.3332	3.0003	5.6666	4.3331	6.3334	5.1334
Peppers	4.3450	3.0121	5.6783	4.3450	6.3448	5.1450
Jetplane	4.3329	2.9996	5.6664	4.3331	6.3331	5.1330

Table 3. Percentage increase in the ER of the proposed program in the (3, 4) scheme.

Test Image	Scheme [4]	Scheme [24]	Scheme [26]	Scheme [27]
Goldhill	140%	74%	186%	144%
Baboon	162%	312%	667%	136%
Boats	129%	57%	185%	140%
Airplane	75%	20%	86%	148%
Jetplane	109%	38%	190%	148%
Peppers	138%	73%	175%	149%

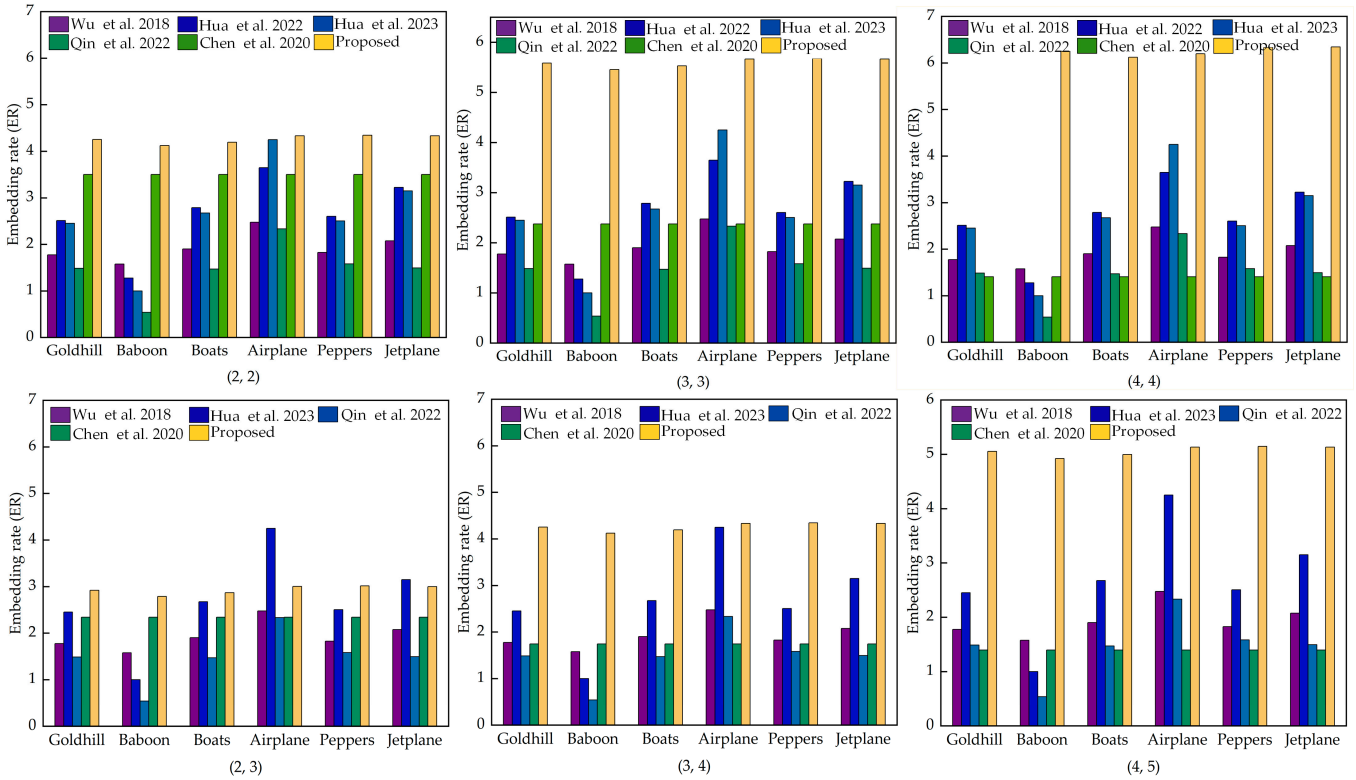

Figure 11. Comparison of the ER for different images at different thresholds [4,23,24,26,27].

4.3. Reversibility

4.3.1. Relevant Parameters

The reversibility of the procedure can be classified into two main aspects: (1) The original image can be recovered completely. (2) Extra data could be extracted losslessly. All test images were tested, and the results are shown in Table 4. The mean squared errors (MSEs) are all 0, the PSNRs are all $+\infty$, and the SSIMs are all 1, which indicates that there is no difference between the reconstructed images and the original images. Since the polynomial coefficients can be recovered correctly based on Theorem 1, the labeled pixels can be recovered accurately. In addition, core users can determine the difference between the zth pixel and the label pixel; thus, the original image can be recovered completely, which indicates that the proposed scheme is completely reversible.

Table 4. MSEs, PSNRs, and SSIMs for different test images.

Test Image	MSE	PSNR	SSIM
Goldhill	0	$+\infty$	1
Baboon	0	$+\infty$	1
Boats	0	$+\infty$	1
Airplane	0	$+\infty$	1
Peppers	0	$+\infty$	1
Jetplane	0	$+\infty$	1

4.3.2. Error Map for Secret Extraction

The error map shows the results of the bit-by-bit comparisons between the processed and original data. Figure 12a–c show the error maps of the extracted S_1, S_2, and S_3 and the original S_1, S_2, and S_3 at threshold (3, 4). The values are all 0, which indicates that the proposed scheme can extract secrets accurately.

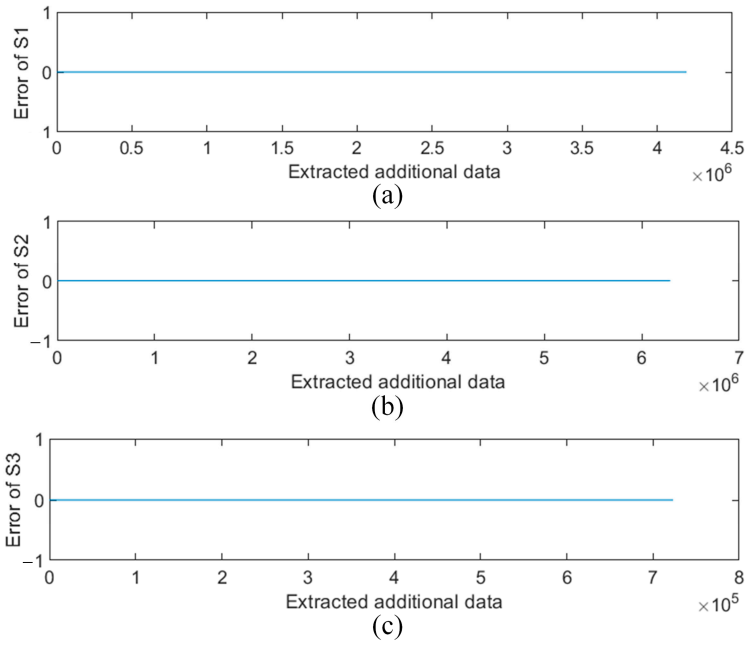

Figure 12. Error maps of S_1, S_2, and S_3. (**a**) Error maps of S_1; (**b**) Error maps of S_2; (**c**) Error maps of S_3.

4.4. Data Extension

The data extension rate is the ratio between the encrypted image size and the original image size, and is calculated as follows:

$$Expansion\ rate = \frac{Total\ bits\ of\ encrypted\ image}{Total\ bits\ of\ original\ image} \tag{16}$$

In this paper, the carrier image is encrypted into n shares by SIS, with each user holding one share. Thus, the relative data expansion rate is 1 for a single embedder, but the total expansion rate depends on the threshold. When n is not too large, the total expansion rate is acceptable. Table 5 shows the comparison results of the different schemes. Schemes [22,28] generate only one encrypted image, and their relative expansion rate is equal to the total expansion rate. However, the scheme [22] uses homomorphic encryption, which generates a large data expansion. Scheme [28] utilizes multi-secret share and lightweight encryption methods, and the data expansion rate is 1. Schemes [23,26] and the proposed scheme generate secret shares via SIS, and all the relative expansion rates are 1, but the total expansion rate depends on the threshold n. Therefore, the data expansion of the proposed scheme could be within an acceptable range by controlling the threshold.

Table 5. Comparison of the data expansion rates.

Scheme	Type	Encryption	Relative Expansion Rate	Total Expansion Rate
Ke et al. [22]	VRIE	Homomorphic encryption	256	256
Hua et al. [23]	VRAE	Secret sharing	$1/(r-1)$	$n/(r-1)$
Qin et al. [26]	VRAE	Secret sharing	1	n
Chen et al. [28]	VRBE	Multi-secret sharing	1	1
Proposed	VRIE + VRAE	Secret sharing	1	n

5. Conclusions

In the current distributed environment, most existing RDH-ED schemes cannot grant corresponding privileges to multilevel identity participants. However, the identities of participants usually have different classes in real distributed environments, and they

need to be given corresponding permission. Accordingly, we propose a multiple embedding scheme based on SIS and users' multilevel identities. Specifically, the participants are divided into core users and ordinary users. For the core users, adaptive difference reservation embedding based on SIS is utilized, which grants more important permission to core users. For ordinary users, pixel bit replacement embedding is used to obtain a better ER. Moreover, the image copyrights are embedded in the polynomial coefficient redundancy generated during the SIS process, which ensures the integrity of the carrier images. The experimental findings demonstrate that our proposed scheme excels in terms of the ER, security, and reversibility, accommodating multiple embeddings across diverse participant identities.

However, our scheme has certain limitations. Maintaining an exact correlation between ciphertext and plaintext pixels through secret sharing makes it less resilient to selective ciphertext attacks. Moreover, cases where a participant defects under the threshold condition can lead to secret leakage. In future research endeavors, we recommend exploring strategies such as attribute encryption and digital signatures to bolster security measures and mitigate these identified shortcomings.

Author Contributions: The authors confirm their contribution to the paper as follows: conceptualization, C.J., M.Z. and F.D.; methodology, C.J., Y.K. and M.Z.; validation, C.J., Y.K. and Z.J.; formal analysis, C.J., M.Z., Y.K. and F.D.; investigation, C.J. and Z.J.; data curation, C.J.; writing—original draft preparation, C.J.; writing—review and editing, C.J., M.Z. and F.D.; visualization, M.Z.; supervision, M.Z.; project administration, M.Z. and F.D.; funding acquisition, M.Z. and F.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant numbers 62272478, 62102450, 61872384, 62102451, 62202496 and the National Social Science Funds of China, grant number 2022-SKJJ-C-091.

Data Availability Statement: The data that support the findings of this study are available from the first author [Chao Jiang] upon reasonable request.

Acknowledgments: The authors would like to thank the editor and anonymous reviewers for their valuable comments that helped to improve this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Puech, W.; Chaumont, M.; Strauss, O. A reversible data hiding method for encrypted images. In Proceedings of the SPIE—The International Society for Optical Engineering, San Jose, CA, USA, 18 March 2008; Volume 6819. [CrossRef]
2. Zhang, X.P. Reversible Data hiding in encrypted image. *IEEE Signal Process. Lett.* **2011**, *18*, 255–258. [CrossRef]
3. Zhou, J.T.; Sun, W.W.; Dong, L.; Liu, X.M.; Au, O.C.; Tang, Y.Y. Secure reversible image data hiding over encrypted domain via key modulation. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 441–452. [CrossRef]
4. Wu, X.T.; Weng, J.; Yang, W.Q. Adopting secret sharing for reversible data hiding in encrypted images. *Signal Process. Off. Publ. Eur. Assoc. Signal Process.* **2018**, *143*, 269–281. [CrossRef]
5. Wang, Y.; Yan, S.; Yang, W.; Cai, Y. Covert communications with constrained age of information. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 368–372. [CrossRef]
6. Ansari, M.R.R.; Allwinnaldo; Alief, R.N.; Igboanusi, I.S.; Lee, J.M.; Kim, D.S. HADES: Hash-Based audio copy detection system for copyright protection in decentralized music sharing. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 2845–2853. [CrossRef]
7. Qian, Z.; Zhang, X.; Wang, S. Reversible data hiding in encrypted JPEG bitstream. *IEEE Trans. Multimed.* **2014**, *16*, 1486–1491. [CrossRef]
8. Huang, F.; Huang, J.; Shi, Y.Q. New framework for reversible data hiding in encrypted domain. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2777–2789. [CrossRef]
9. Xiang, S.; Luo, X. Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 3099–3110. [CrossRef]
10. Wang, X.; Chang, C.C.; Lin, C.C. Reversible data hiding in encrypted images with block-based adaptive MSB encoding. *Inf. Sci.* **2021**, *567*, 375–394. [CrossRef]
11. Yang, Y.; He, H.; Chen, F.; Yuan, Y.; Mao, N. Reversible data hiding in encrypted images based on time-Varying huffman coding table. *IEEE Trans. Multimed.* **2023**, *25*, 8607–8619. [CrossRef]

12. Chen, K.M.; Chang, C.C. High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement. *J. Vis. Commun. Image Represent.* **2019**, *58*, 334–344. [CrossRef]
13. Yin, Z.X.; Xiang, Y.Z.; Zhang, X.P. Reversible data hiding in encrypted images based on multi-MSB prediction and huffman Coding. *IEEE Trans. Multimed.* **2020**, *22*, 874–884. [CrossRef]
14. Puteaux, P.; Puech, W. A Recursive Reversible data hiding in encrypted images method with a very high payload. *IEEE Trans. Multimed.* **2021**, *23*, 636–650. [CrossRef]
15. Gao, G.; Zhang, L.; Lin, Y.; Tong, S.; Yuan, C. High-performance reversible data hiding in encrypted images with adaptive Huffman code. *Digit. Signal Process.* **2023**, *133*, 103870. [CrossRef]
16. Zou, H.; Chen, G.H. Reversible data hiding in encrypted image with local-correlation-based classification and adaptive encoding strategy. *Signal Process.* **2023**, *205*, 1108847. [CrossRef]
17. Ke, Y.; Zhang, M.Q.; Liu, J. A multilevel reversible data hiding scheme in encrypted domain based on R-LWE. *Comput. Res. Dev.* **2016**, *53*, 2307–2322. [CrossRef]
18. Ke, Y.; Zhang, M.Q.; Liu, J.; Su, T.T.; Yang, X.Y. Fully homomorphic encryption encapsulated difference expansion for reversible data hiding in encrypted domain. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 2353–2365. [CrossRef]
19. Chinnasamy, P.; Deepalakshmi, P.; Dutta, A.K.; You, J.; Joshi, G.P. Ciphertext-Policy Attribute-Based Encryption for Cloud Storage: Toward Data Privacy and Authentication in AI-Enabled IoT System. *Mathematics* **2022**, *10*, 68. [CrossRef]
20. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
21. Priyanka, S.; Raman, B. Reversible data hiding based on Shamir's secret sharing for color images over cloud. *Inf. Sci.* **2018**, *422*, 77–97. [CrossRef]
22. Ke, Y.; Zhang, M.Q.; Zhang, X.P.; Liu, J.; Su, T.T.; Yang, X.Y. A reversible data hiding scheme in encrypted domain for secret image sharing based on Chinese Remainder Theorem. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 2469–2481. [CrossRef]
23. Hua, Z.Y.; Wang, Y.; Yi, S.; Zhou, Y.; Jia, X. Reversible data hiding in encrypted images using cipher-feedback secret sharing. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 4968–4982. [CrossRef]
24. Hua, Z.Y.; Wang, Y.; Yi, S.; Zheng, Y.F.; Liu, X.Y.; Chen, Y.Y.; Zhang, X.P. Matrix-based secret sharing for reversible data hiding in encrypted images. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 3669–3686. [CrossRef]
25. Yu, C.; Zhang, X.; Qin, C.; Tang, Z. Reversible Data Hiding in Encrypted Images with Secret Sharing and Hybrid Coding. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 6443–6458. [CrossRef]
26. Qin, C.; Jiang, C.Y.; Mo, Q.; Yao, H.; Chang, C.C. Reversible data hiding in encrypted image via secret sharing based on GF(p) and GF(2^8). *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 1928–1941. [CrossRef]
27. Chen, B.; Lu, W.; Huang, J.W.; Weng, J.; Zhou, Y.C. Secret sharing based reversible data hiding in encrypted images with multiple data-hiders. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 978–991. [CrossRef]
28. Chen, Y.C.; Hung, T.; Hsieh, S.; Shiu, C. A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic schemes. *IEEE Trans. Data Forensics Secur.* **2019**, *14*, 3332–3343. [CrossRef]