applied sciences

MDPI

Article

Reversible Data Hiding in Encrypted Images Based on the Mixed Multi-Bit Layer Embedding Strategy

Ruihua Liu and Quan Zhou *

Xi'an Institute of Space Radio Technology, Xi'an 710100, China; lrh5042019@163.com
* Correspondence: zhouq97@cast504.com

Abstract: With the increasing requirements for the security of medical data, military data, and other data transmission, data hiding technology has gradually developed from only protecting the security of secret data to all transmission data. As a necessary technical means, reversible data hiding in encrypted images (RDH-EIs) provides superior performance in terms of security. To simultaneously improve the effectiveness of RDH-EIs, this work proposes a mixed multi-bit layer embedding strategy in encrypted images. The cover image is processed into two categories: available hidden blocks (AHBs) and unavailable hidden blocks (UHBs) at the sender. Then, all data are embedded in the multi-bit layer of the encrypted pixels in AHBs through two embedding strategies to obtain the transmission image. At the receiver, the user can extract the needed data separably according to different keys to achieve error-free extraction of the secret data and lossless recovery of the cover image. The experimental results show that the proposed scheme has the advantages of superior embedding capacity and high decryption quality over the current state-of-the-art works.

Keywords: reversible data hiding in encrypted images; high capacity; separability; data transfer safety; adaptive multi-bit layer embedding strategy; fixed multi-bit layer embedding strategy

Citation: Liu, R.; Zhou, Q. Reversible Data Hiding in Encrypted Images Based on the Mixed Multi-Bit Layer Embedding Strategy. *Appl. Sci.* **2023**, *13*, 5696. https://doi.org/10.3390/app13095696

Academic Editor: Mostafa Fouda

Received: 17 March 2023
Revised: 27 April 2023
Accepted: 2 May 2023
Published: 5 May 2023

1. Introduction

Data hiding technology is an essential technology in the field of data security. At the early stage of development, it can only guarantee the security of the secret data. Still, it cannot fully recover the cover image at the receiver, such as the least significant bit (LSB) algorithm [1]. Research focuses on reversible data hiding (RDH) at the second stage of technology development to achieve lossless recovery of the cover image. It has been widely applied in military, medical, and information security applications. In the last two decades, RDH technology has flourished [2]. General RDH algorithms are based on three types of techniques: lossless compression [3–5], extended transformations [6–11], and histogram shifting (HS) [12,13]. With these techniques, many algorithms can achieve better performance. For example, in [12,13], a general framework for constructing HS-based RDH is used to effectively achieve high capacity and low distortion by employing specific shifting and embedding functions.

The above traditional RDH methods are used in plaintext cover images. During data transmission, the visual quality of the transmission image is close to that of the original cover image because of the technology demand, so it is not easy for the secret data to be discovered. However, in this case, the content of the cover image is constantly exposed. Data security has become increasingly important in recent years with the continuous development of data privacy protection technology. On the one hand, the data sender does not trust the transferred server because the data are easily stolen. On the other hand, the plaintext cover image is more likely to be cracked. Then, RDH in encrypted images (RDH-EIs) was developed to meet the needs of privacy and security. RDH-EI technology embeds data in encrypted images so that the contents of secret data and the cover image are not visible during transmission. This benefit has significant implications for military

communications, private medical data, trade secret data, and other data security areas. Therefore, it has been a research hotspot in recent years.

The existing RDH-EI schemes are mainly divided into two main categories: vacating room after encryption (VRAE) [14–22] and reserving room before encryption (RRBE) [23–39]. They are based on the order in which the hidden room is vacated and image encryption. In general, the vacating room and reserving room mean analyzing the current image through certain techniques to create some space for other data hiding. The space can be empty bits, and data will be placed directly in empty positions. It can also be locations with data, but the data can be modified to reflect embedded information and can be completely restored.

Under the VRAE framework, the sender first encrypts the cover image, then uses some technologies to make space in the encrypted image, and finally uses the data hiding key to embed data in the vacated room. Zhang [14] realized data hiding through flipping the low 3 bits LSB of pixels, but there is error data extraction. Hong et al. [15,16] improved the wave function and used the correlation of pixels to reduce the error rate. To vacate the room, the LSBs of encrypted pixels were compressed with a special matrix [17]. Kim [18] used Hamming code to hide data in compressed and encrypted image blocks and introduced quantization to adjust the embedding capacity. Block encryption [19–22] was used to preserve the correlation in the encrypted image, and then traditional RDH technologies such as HS and difference expansion were designed to embed data.

In the above schemes, some methods may have error codes in data extraction or image recovery, which are not completely reversible. Some methods are completely reversible but limit the embedding capacity. Therefore, the RRBE framework was developed to improve the situation. Compared with the previous VRAE schemes, RRBE schemes are more convenient for the high embedding capacity, reversibility, and security.

Under the RRBE framework, the sender pre-processes the cover image before encryption to reserve room or generate a new image format conducive to hiding. In previous studies, classifying image blocks and pixels by using some rules is a common technical means. Ma et al. [23] classified image blocks by smoothing function calculation to reserve the LSBs of pixels in blocks. Qin et al. [24] divided image blocks according to the predesigned threshold and reserved room by compressing some blocks' LSB. Wu et al. [25] classified the cover image into blocks with different scales. Then, the lowest two bits of each pixel were used to vacate room. The pixels were sorted according to the predefined fixed classification mode in [26,27]. For the above methods, the adaptive classification method [23–25] has the advantage of mining the hidden space as much as possible but requires uncertain auxiliary information. The fixed classification methods [26,27] have fixed costs but limit the amount of hidden space. Prediction technology is also applied through high performance [28–30]. Most significant bit (MSB) prediction was proposed in [31–34] based on the high correlation of the adjacent pixel values of the cover image, which significantly improves the embedding capacity and the image recovery quality. Interpolation techniques [26,35,36], compression [24,37,38], and HS [39] have also been further studied and applied to encrypted images.

These schemes all use a single hider and maintain the amount of data that is always kept consistent with the size of the cover image during transmission. In addition, in recent years, some scholars have proposed multiple hiders schemes [40–42] based on secret sharing, which has also promoted the development of this field.

The above technologies have improved recovery quality and embedding rates. However, there may be problems with reserving room and how data embedding can be achieved after encryption.

It can be found that some problems exist in previous studies, such as the fact that the secret data cannot be extracted completely and accurately in some algorithms [14–16], so complete reversibility cannot be achieved. The embedding capacity needs to be improved [19,20,23,33], but the image recovery quality will be affected. The joint schemes at the receiver [14,26,27,37] and others need to process the data in order for the operation to be simple but low in flexibility; the separable schemes such as [17–25,33,36] are crucial

to achieving independence in data extraction and image recovery but also increase the complexity. Therefore, RDH-EI is more challenging than traditional RDH. Balancing the embedding rate, the accuracy of secret data extraction, the quality of image recovery, and separability is a technical challenge. In this paper, all the challenges mentioned above are considered. A separable RDH-EI scheme based on a mixed multi-layer embedding strategy is proposed to improve the embedding capacity under error-free data extraction conditions. The scheme belongs to the RRBE framework.

The contributions of the proposed scheme are as follows:

- A two-round block segmentation method with different sizes is proposed, which extends the number of hidden blocks and is conducive to providing embedding room;
- Adaptive multi-bit embedding (AMLE) and fixed bit layer embedding (FMLE) strategies are designed to embed all data jointly, which can compress auxiliary data to reserve more room for data embedding;
- Separability and full reversibility can be guaranteed;
- Decrypted images maintain satisfactory quality at maximum embedding capacity.

The rest of the paper is organized as follows. Section 2 shows the related works. Section 3 describes the specific scheme. Section 4 gives the experimental results and performance analysis. Section 5 presents the conclusion.

2. Related Works

This section mainly reviews the relevant RDH-EI algorithms from recent years on the study of available hidden spaces.

In [19,20], block encryption was adopted. Then, the encrypted image was divided into 2×2 blocks, and lossless compression techniques such as Huffman coding were used to compress the bit plane to vacate space. Finally, the cover image was recovered by the shared MSBs. However, the block was too small, which affected the embedding capacity.

In [23], selecting available blocks is in the pre-processing step. A function is defined to measure the first-order smoothness of blocks, and the blocks are classified into smooth blocks and complex blocks by the function value. The LSB bits of the pixels in the smooth blocks are then reserved for hiding. This scheme provides multi-layer embedding, but with the increase in the used bit planes, the image recovery performance decreases significantly. Therefore, this scheme studies three LSB planes at most.

In [24], the threshold T is designed in advance, ranging from 5–100. The smaller T is, the fewer image blocks are available. The space can be reserved by compressing the LSB layer in the block set of the smooth region. Because only part of the image block is modified during data embedding, the quality of the directly decrypted image is satisfactory, but the embedding capacity is small.

In [25], the cover image was processed into blocks with different scales after three rounds of segmentation and marked. The lowest two LSBs of each pixel in the available block are reserved spaces. Then, the average and reference values of pixels in the block are embedded as auxiliary data. The algorithm is affected by the image block size and fixed threshold, and the space needs further improvement.

In the above mentioned works, lossless compression technology, the block size, the fixed threshold, single-layer embedding, and other factors affect the algorithms' performance. Although these algorithms have made some accomplishments, they still need to be improved in terms of embedding capacity and decryption quality.

3. Proposed Scheme

While ensuring that both the cover image content and secret data can be securely transmitted and restored, to improve the embedding capacity and decryption quality of the cover image, this paper proposes a separable RDH-EI scheme.

The scheme includes image pre-processing, image encryption, data embedding, and data extraction and recovery. Figure 1 shows the basic framework of the proposed scheme. The sender segments the cover image into non-overlapping blocks by two-round segmenta-

tion and classifies the blocks into available hidden blocks (AHBs) and unavailable hidden blocks (UHBs) based on the correlation of pixels. Then, it generates a location map and creates the hidden room by encoding the pixel information in the AHBs in turn. Next, the cover is encrypted with an encryption key *Ke*. Finally, all data, including secret data and auxiliary data, are embedded by two embedding strategies (AMLE and the FMLE) based on the hiding key *Kd*. After transmission, the receiver can extract the required data and recover the cover image separately, depending on the type of key possessed.

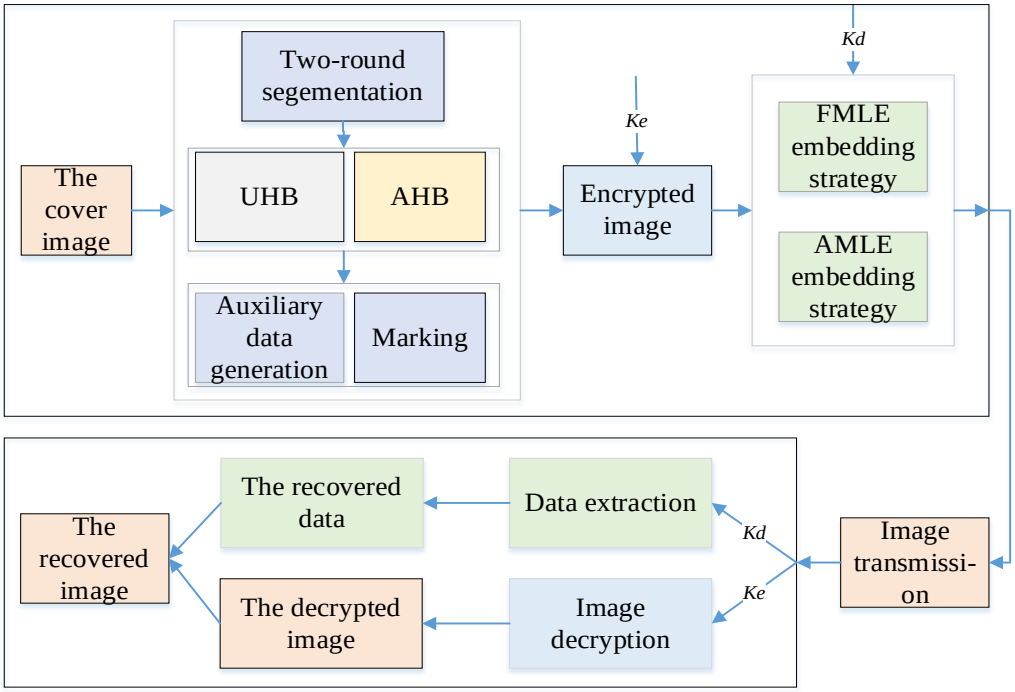


Figure 1. The proposed scheme based on the mixed multi-bit layer embedding strategy.

3.1. Pre-Processing

In the proposed scheme, pre-processing refers to the stage before image encryption. The main function is to reserve room. This section carefully illustrates the steps. It contains two-round segmentation, parameters design, auxiliary data generation, and other steps.

- First-round segmentation.

Input the cover image with the size of $M \times N$ and decompose it into non-overlapping blocks with the size of $m \times n$ in the main order of rows. The number of pixels in a block is $l_1 = m \times n$. The block is represented as the column vector $X = \{x_1, x_2, \ldots x_{l_1}\}$.

Next, set threshold parameter T and calculate differences. T takes the exponent of 2, as the condition of whether the block can be hidden. Then taking the first pixel x_1 of each block as a reference, calculate the difference between X and x_1, and record it as:

$$D = \{d_1, d_2, \ldots d_{l_1}\}. \quad (1)$$

Let the maximum value of D be max(D). If max(D) < T, it means that it is a smooth block. Then mark the block as an AHB by setting LSB of x_1 to 1, and type BT_1. Otherwise, mark the block as an UHB by setting LSB to 0. Since the hidden room needs to be reserved in the AHBs, the auxiliary data generated by them is recorded. The auxiliary data for each AHB are as follows. The binary representation of (1) is:

$$D_{01} = \left\{(d_1)_{01}, (d_2)_{01}, \ldots (d_{l_1})_{01}\right\}. \quad (2)$$

The positive and negative of D are recorded by parameter w with 1 bit. The length of D_{01} is related to T, usually $\log_2(T)$ bit. To reach pixels recovery without loss at the receiver, w and the D_{01} are combined to generate the auxiliary data E_1, which is represented as follows:

$$E_1 = \left\{ w_1, (d_1)_{01}, w_2, (d_2)_{01}, \ldots w_{l_1}, (d_{l_1})_{01} \right\}. \quad (3)$$

- Second-round segmentation.

After first-round segmentation, the image blocks are divided into AHBs and UHBs. Then, the UHBs are used for second-round segmentation. First, the UHB is divided into two sub-blocks, P and Q, to expand the embedded space. They are generated by the segmentation of the column vector X and represented as follows:

$$P = \{x_1, x_2, \ldots x_{l_2}\}, Q = \{x_{l_2+1}, x_{l_2+2}, \ldots x_{l_1}\}. \quad (4)$$

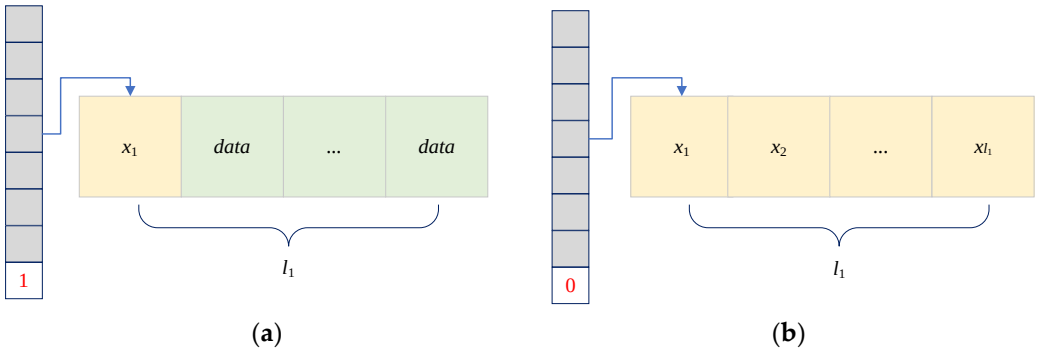
Second, the sub-blocks are classified in the same way as the first round, and then the smooth blocks are marked as AHBs, type BT_2, and the other blocks are complex, marked as UHBs, type BT_3. Unlike the first round, the marker bit for P is in the penultimate position of x_1, and the marker bit for Q is still in the LSB. After the above processing, the whole image is marked into three categories. The description of the blocks is shown in 0.

Table 1 describes the characteristics of the three types of blocks. Column 2 shows the block type. Column 3 shows the length, l_1, l_2, or l_1-l_2. Column 4 shows the number of each type. The last column shows the reference pixels in different types of blocks, which may be x_1 or x_{l2+1}.

Table 1. Description of different types of blocks.

Block Type	Classification	Length	Number	Reference Pixel
BT_1	AHB	l_1	N_1	x_1
BT_2	AHB	$l_2(P)$ or l_1-$l_2(Q)$	N_2	x_1 or x_{l2+1}
BT_3	UHB	l_1-$l_2(Q)$ or $l_2(P)$	N_3	x_{l2+1} or x_1

The above process of classification and position marking is shown in Figure 2.



(**a**) (**b**)

Figure 2. *Cont.*

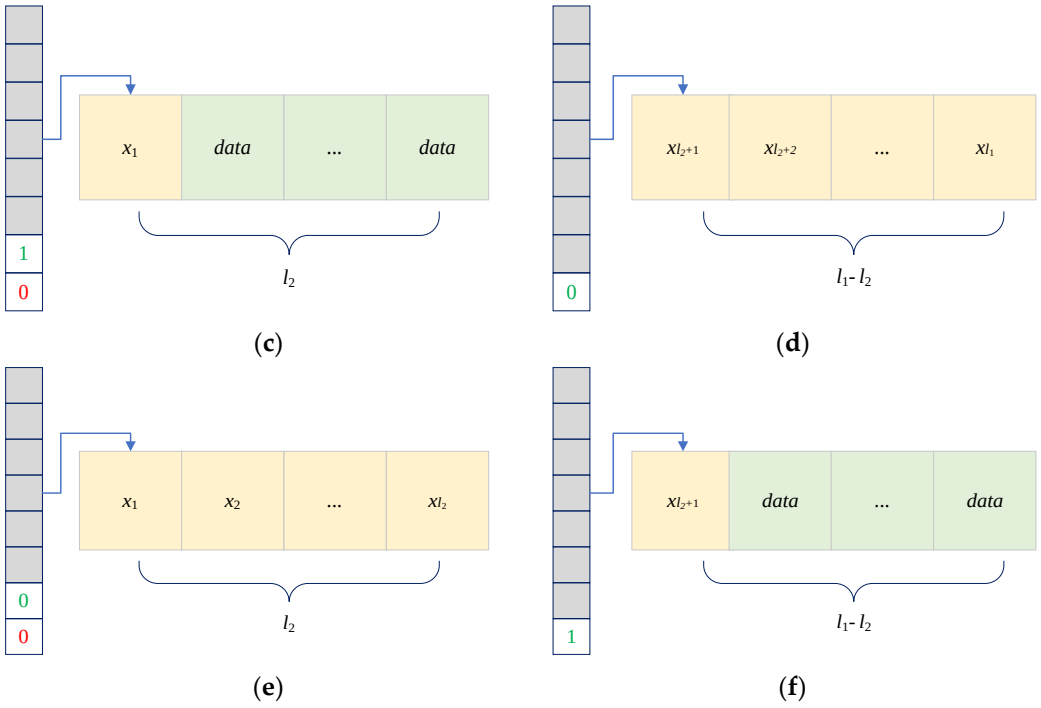


Figure 2. Schematic diagram of block processing: (**a**) AHB_BT_1, (**b**) UHB, (**c**) AHB_BT_2 (P), (**d**) UHB_BT_3(Q), (**e**) UHB_BT_3(P), and (**f**) AHB_BT_2(Q).

Figure 2 gives the results of block processing after the steps above. The sub-figures contain schematic diagrams of two-round block segmentation, classification, and position marking. In Figure 2a,b show the two types of image blocks obtained after the first-round segmentation. (a) describes the AHB. The value of the marker position (LSB) is 1 and the block type is BT_1. The data are embedded in 2~l_1 bytes. (b) describes the UHB, which needs to be segmented for the second round to obtain sub-blocks P and Q. The marker value is 0. P may be as (c) and (e), and Q may be as (d) and (f). There are two possible types of each sub-block. Hence, P and Q have four combinations, such as (c) and (d), (e) and (f), or other combinations, but the total length of P and Q is l_1. The type of AHBs in P and Q is BT_2. The data are embedded in 2-l_2 bytes or l_2 + 1~l_1 bytes. The specific segmentation method will affect the performance of the scheme.

As shown in Figure 2, the specific location is marked, and so the marked image is obtained. In addition, the mark value and two location maps, Map1 and Map2, are generated. Among them, Map1 is used to store the marks of first-round segmentation and Map2 is used to store the marks of second-round segmentation. At the same time, the original LSB of x_1 and the length: S_1, S_2, L_{S_1}, and L_{S_2} are also recorded.

$$L_{S_1} = (M \times N)/(m \times n). \tag{5}$$

$$L_{S_2} = N_2 + N_3 = 2 \times ((M \times N)/(m \times n) - N_1). \tag{6}$$

Similarly, E_2 of sub-blocks P and Q is obtained in the same way as E_1. So far, all auxiliary data: S_1, S_2, E_1, and E_2, and their length: L_{S_1}, L_{S_2}, LE_1, and LE_2 are obtained.

3.2. Encryption

The pseudo-random matrix with the same size as the marked image bit matrix is generated by the encryption key Ke, and then, XOR is performed with the marked image bit matrix to obtain the encrypted image.

The range of the gray pixel value $f(i, j)$ at the position (i, j) in the marked image is [0, 255]. Let the bit of each pixel be: $b_{i,j,1}$, $b_{i,j,2}$, ... $b_{i,j,8}$, and the relationship between it and the gray value $f(i, j)$ is shown in (7) and (8):

$$b_{i,j,k} = \left\lfloor \frac{f(i,j)}{2^{k-1}} \right\rfloor \bmod 2, k = 1, 2, \ldots 8 \tag{7}$$

$$f(i,j) = \sum_{k=1}^{8} (b_{i,j,k} \times 2^{k-1}), k = 1, 2, \ldots 8. \tag{8}$$

Then perform the XOR operation between the pseudo-random binary array $r_{i,j,k}$, and $b_{i,j,k}$. As shown in (9):

$$B_{i,j,k} = b_{i,j,k} \oplus r_{i,j,k}, k = 1, 2, ...8. \tag{9}$$

where B is the encrypted image. The encryption key Ke is also used as the decryption key, which is reversible, ensuring that the image content before encryption is completely restored in the decryption phase.

3.3. Data Embedding

In this section, Kd is used to control the embedding process of secret data. Namely, the secret data are embedded in the encrypted image B through the embedding strategy. This section describes two strategies: the FMLE strategy and the AMLE strategy.

- The FMLE strategy

In this way, all AHBs use a fixed threshold. Each AHB uses the same T, so all hidden pixels in the AHBs use the same number of error bits with the length of $\log_2(T)$. The remaining number of embeddable bit layers is fixed.

Figure 3 illustrates the FMLE strategy. Figure 3a,b show how the AHBs for types of BT_1 and BT_2 are processed under the strategy. Below, blocks of type BT_1 will be elaborated as the example. In Section 3.1, x_1 in the block is LSB marked to obtain the marked block (see the left part in the sub-figure). Figure 3a gives the strategy of the first AHB. To consider decryption recovery at the receiver, it is necessary to embed m and n. The strategy adopted by other AHBs is shown in Figure 3b. Then, the encrypted block (see the middle part of the sub-figure) is obtained by encryption in Section 3.2. The encrypted image block loses relevance, and the LSB mark is covered. Therefore, during data embedding, the two location maps, Map1 and Map2, and the hiding key are used to guide secret data embedding. If the value in the location map is 1, the block belongs to the AHB, otherwise, it belongs to the UHB. Meanwhile, the value is embedded in the MSB position in x_2 of the block with 1 bit in each block. In this paper, x_1 is the reference pixel and plays an essential role in the cover recovery phase, so it cannot be used for data embedding.

The data embedding structure is given in the third sub-figure of Figure 3a,b. The auxiliary information T, E_1, and S of the block are embedded in the low position of the pixels. The remaining space is used for secret data hiding. Thus, the order of embedding data is MSB, m, n, T, E_1, S_1, and secret data in the first AHB. For other AHBs, unlike Figure 3a, there is no need to embed the parameters again.

Figure 3c shows the processing method of UHBs. Although secret data are not hidden in the UHBs, the marking information MSB must also be embedded for the receiver to correctly distinguish the AHBs and the UHBs.

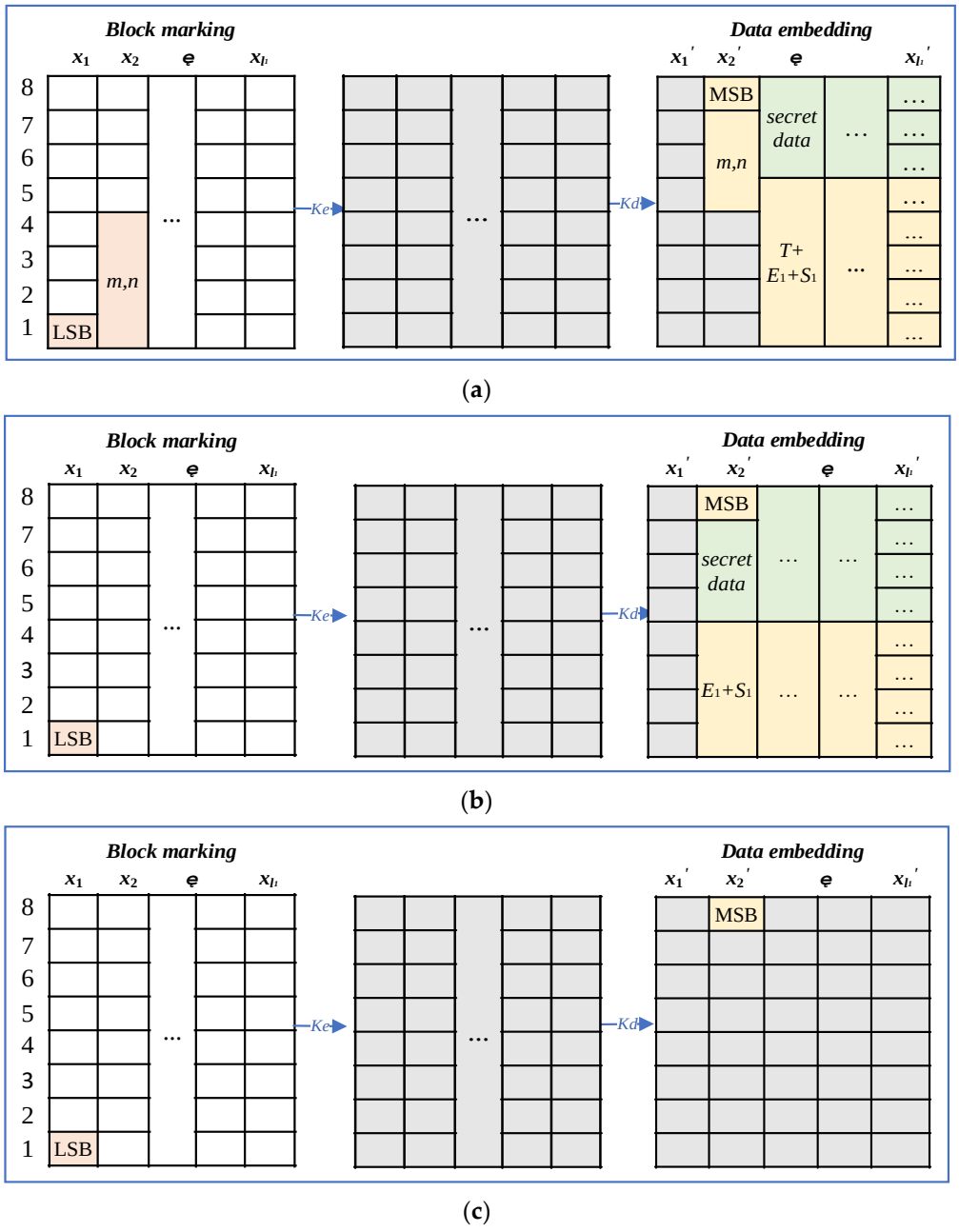


Figure 3. Schematic diagram of the FMLE embedding strategy: (**a**) the AHB (the 1st BT_1 block), (**b**) the AHB (other BT_1 blocks and BT_2 blocks), and (**c**) the UHB (BT_3).

- The AMLE strategy

In this way, all AHBs adopt dynamic thresholds. Each AHB directly uses the max(D) of the block itself as the threshold and the number of error bits is dynamic with a length of $\log_2(\max(D))$. Because the max(D) corresponding to each block is inconsistent, the remaining number of embeddable bit layers in a hidden pixel is dynamic.

Figure 4 illustrates the AMLE strategy. The process is consistent with the FMLE strategy, but there is a difference in data embedding. Under the FMLE strategy, the fixed T must be embedded into the first AHB but there is no need in other AHBs. However, during the embedding process under the AMLE strategy, each dynamic T generated by the original block can be embedded. T may be max(D)-1, max(D)-2, . . . and so on. As shown in Figure 4a,b, they are embedded with fixed bits in AHBs in turn. If the max(D) < T, the number of error bits will decrease. Therefore, this strategy dynamically compresses the

auxiliary information E_1 into E_1' which further increases the available hidden space for secret data embedding.

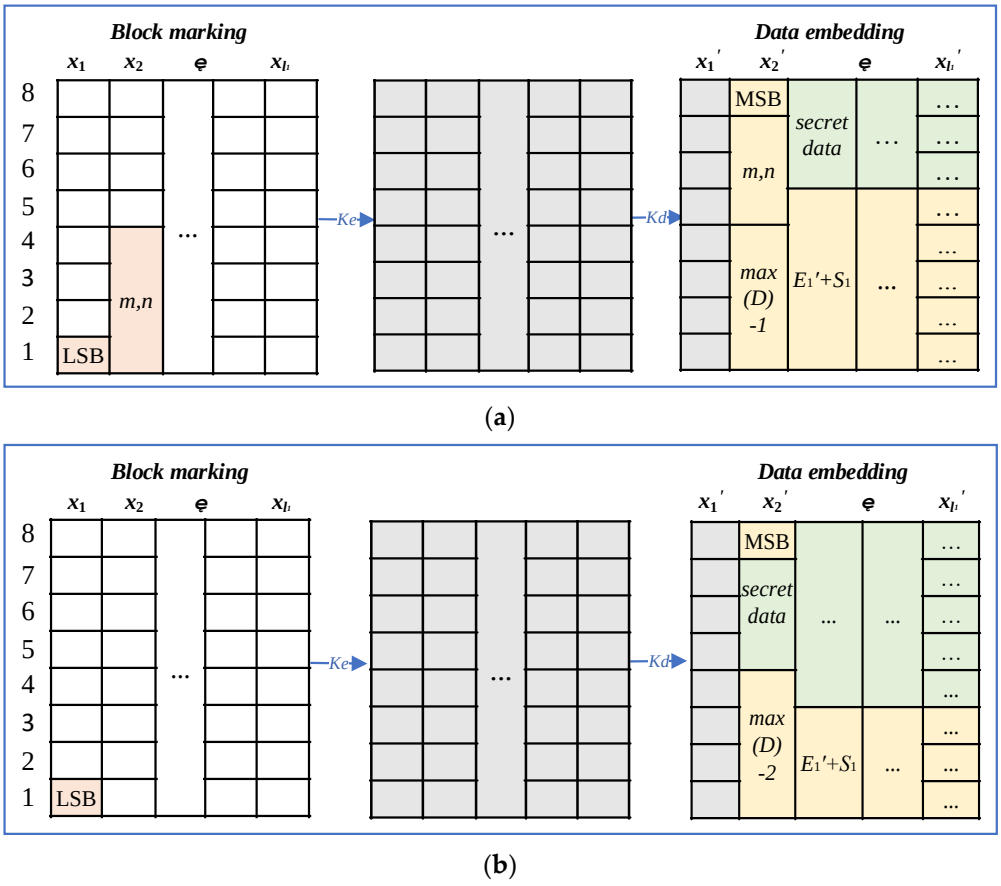


Figure 4. Schematic diagram of the AMLE embedding strategy for AHBs with the type of BT_1: (**a**) the AHB (the 1st BT_1 block) and (**b**) the AHB (other BT_1 blocks and BT_2 blocks).

The data embedding structure is given in the third sub-figure of Figure 4a,b. The auxiliary information max*(D)*, E_1 and S_1 of the block are embedded in the low position of the pixels. The order of embedding data is MSB, m, n, max*(D)*, E_1', S_1, and secret data in the first AHB. For other AHBs, embed the above parameters except for m, n.

In addition, the processing of UHBs is the same way as that of the FMLE strategy.

- Capacity analysis and comparison of the two strategies

Table 2 describes the comparison of the capacity and use conditions by using examples. Taking the AHB (BT_1) as an example, the capacity and error bits costs of the two strategies are compared as follows.

Table 2. Comparison of two embedding strategies (FMLE and AMLE).

Max (D)	Single Pixel		Single Block		Comparison (Single Block)	
	FMLE (bits)	AMLE (bits)	FMLE (bits)	AMLE (bits)	Decrease in cost (bits)	Increase in cost (bits)
0	n_f	0	$n_f \times (l_1 - 1)$	0	$n_f \times (l_1 - 1)$	n_f
1	n_f	1	$n_f \times (l_1 - 1)$	$(l_1 - 1)$	$(n_f - 1) \times (l_1 - 1)$	n_f
2~3	n_f	2	$n_f \times (l_1 - 1)$	$2 \times (l_1 - 1)$	$(n_f - 2) \times (l_1 - 1)$	n_f
4~7	n_f	3	$n_f \times (l_1 - 1)$	$3 \times (l_1 - 1)$	$(n_f - 3) \times (l_1 - 1)$	n_f
8~15	n_f	4	$n_f \times (l_1 - 1)$	$4 \times (l_1 - 1)$	$(n_f - 4) \times (l_1 - 1)$	n_f
...	...	...	...	...	...	...
$T/2$~T-1	n_f	n_f	$n_f \times (l_1 - 1)$	$n_f \times (l_1 - 1)$	0	n_f

In Table 2, $n_f = \log_2(T)$ means the number of error bits represented by T in binary for a single hidden pixel. Column 1 gives several cases when the pixel difference within the block is less than T and the embedding conditions are met. Columns 2 and 3 indicate the difference coding length of a single pixel in the AHB under the two strategies. It can be found that under the FMLE strategy, all the differences are encoded as n_f bits. While under the AMLE strategy, the bits required for the difference coding are gradually increased until they are the same as in the case of FMLE. Columns 4 and 5 show the coding length of the AHB of BT_1 type under the two strategies. Columns 6 and 7 represent the number of bits saved and the increased cost in each block compared with AMLE and FMLE. Theoretically, AMLE uses dynamic thresholds, and the length of E' will be shorter than E in all probability, but the fixed cost will also increase, so it is necessary to judge which strategy to choose in the processing process. The following is a brief analysis.

Let the threshold be T, the length of an AHB be l_1, and the max(D) of the block be T/k ($k > 1$), where k is the scale factor. Under the FMLE strategy, the length of E generated by this block is:

$$L_{BE} = (l_1 - 1) \times \log_2(T)(\text{bits}). \tag{10}$$

Under the AMLE strategy, the length of E generated by this block is:

$$L_{BE'} = (l_1 - 1) \times \log_2(T/k) + \log_2(T)(\text{bits}). \tag{11}$$

The calculation is as follows:

$$\begin{aligned} L_{BE} - L_{BE'} &= (l_1 - 1) \times \log_2(T) - ((l_1 - 1) \times \log_2(T/k) + \log_2(T)) \\ &= (l_1 - 1) \times n_f - ((l_1 - 1) \times n_f - (l_1 - 1) \times \log_2(k) + n_f) \\ &= (l_1 - 1) \times \log_2(k) - n_f \end{aligned} \tag{12}$$

If $L_{BE} - L_{\mathrm{BE}'} > 0$, the block uses FMLE to generate more auxiliary information and the hidden space is less. Therefore, AMLE should be adopted. Otherwise, FMLE is adopted. At different lengths, a mixed embedding strategy can ensure a reduction in auxiliary information and an increase in embedding capacity. The specific calculation is as follows:

$$\begin{aligned} (l_1 - 1) \times \log_2 k - n_f &> 0 \\ l_1 &> \frac{\log_2 k + n_f}{\log_2 k} \\ l_1 &> n_f + 1 \end{aligned} \tag{13}$$

Formula (13) gives the judgment conditions for selecting two strategies. The derivation here is based on l_1, which applies to all AHBs. The results show that the block length affects the selection of the embedding strategy in the case of given T. When the block length is greater than $n_f + 1$, the pure capacity of the hidden block can be guaranteed to increase. However, when the block length does not meet this condition, it is also acceptable. The capacity increase caused by the small difference front can offset the cost of the large difference and have a balance, which can also realize the overall capacity increase. However, this situation depends on the gray values distribution. The more uniform the distribution, the lower the requirements on the block length and the easier it is to hide data.

Thus, with this step, a mixed embedding strategy with a condition is created and used to generate encrypted transmission image containing secret data.

3.4. Data Extraction and Image Recovery

- Case 1: Receiver only has hiding key *Kd*

Firstly, extract parameters m and n at the position of the second pixel. In the embedding strategy, regardless of whether the first block belongs to AHB or UHB, the data of m and n are in the second pixel, and its length is fixed. Next, according to *Kd*, extract all data and classify them. The received image is segmented into blocks with a size of $m \times n$, and the MSB of the second pixel in each block (except the first block) is extracted. If MSB is 1, AHBs

(BT_1) can be acquired. Otherwise, segment the block for the second round, and extract the MSB in the same way to obtain AHBs (BT_2) and UHBs (BT_3). Finally, all data are extracted from all AHBs. The extracted data includes secret data and auxiliary data.

For AHBs belonging to BT_1, the auxiliary data T is extracted from the first pixel based on the AMLE strategy structure, and the number of error bits is fixed. For AHBs belonging to BT_2, the max(D) costs fixed bits in the AMLE strategy structure. Based on max(D), each block recalculates a different T and gives a different number of error bits. LE_1, LE_2, L_{S_1}, and L_{S_2} can be calculated by m, n, the number of error bits and the number of pixels in an AHB.

By using the above parameters and the mixed embedding strategy, separate auxiliary data S_1, E_1, S_2, E_2, and secret data can be obtained.

- Case 2: Receiver only has encryption key Ke

Firstly, extract parameters m, n like Case 1. Then, directly decrypt the received image with Ke to obtain the first decrypted image, De_1. De_1 is segmented with m and n to improve the decrypted quality. The LSB of x_1 is extracted in each block. If the LSB mark is 0, second-round segmentation and mark information extraction are performed. After processing, the blocks marked 0 are classified as UHBs, and the remaining blocks are classified as AHBs. Finally, the first pixel in each block is used to restore other pixels in the AHBs, and the pixels are unchanged in the UHBs. Here, the final decrypted image De_2 can be obtained.

- Case 3: Receiver has encryption key Ke and hiding key Kd

Perform subsequent processing according to the results in Case 1 and Case 2. Take the decrypted image De_2 in Case 2 as the image object to be processed. Then, recover the LSBs of the reference pixels in AHBs and UHBs with S_1 and S_2, and calculate the other pixel values by E_1 and E_2 in the AHBs. The calculation of pixel values is reversible according to the principle in Section 3.1. So far, a completely lossless recovered image can be obtained.

4. Experimental Results and Discussion

To demonstrate the validity and superiority of our scheme, the experiment adopts standard grayscale images from publicly available datasets [43]. They are Lena, Airplane, Peppers, and Baboon. The experimental configurations are a 3.60 GHz Intel i7 processor, a Windows 7 operating system, and Matlab R2018a.

4.1. Embedded Capacity Analysis

Several factors affect the embedding capacity. One is the size of the block: m and n. The larger size will reduce L_{S_1} and L_{S_2}. However, it will also increase the max(D) to limit the number of AHBs and the embedding capacity. Therefore, the two-round segmentation of block size needs to be studied. The second factor is T, which directly determines the number of AHBs. The larger T is, the greater the number of AHBs that satisfy the condition.

This section uses the BT_1 block in AHBs as an example to simulate the above parameters and analyzes the relationship between them and the embedding capacity. Parameters m and n are set to 2×2, 2×4, and 4×4. T is set to 4, 8, 16, 32, and 64. Here, the FMLE strategy is used for analysis, and the results are shown in the figure below.

In Figure 5a, the horizontal axis represents different thresholds, and the vertical axis represents the average pure capacity of the experimental images. The curves in (a) represent four block size schemes: $m2n2$, $m2n4$, and $m4n4$, where m * n * denotes that the block is with m rows and n columns. Each curve has five points, meaning the pure capacity when T = 4, 8, 16, 32, and 64 from left to right. It can be found that when T is fixed, the pure capacity corresponding to the three schemes is close, and $m2n4$ is more stable from the overall average. From the results of different T, during the change of T from 4 to 64, several curves show a trend of increasing first and then decreasing, which means that the pure capacity also increases first and then decreases. The average pure capacity is the highest when T = 16. Therefore, in the subsequent experiments, T = 16 is used for simulation.

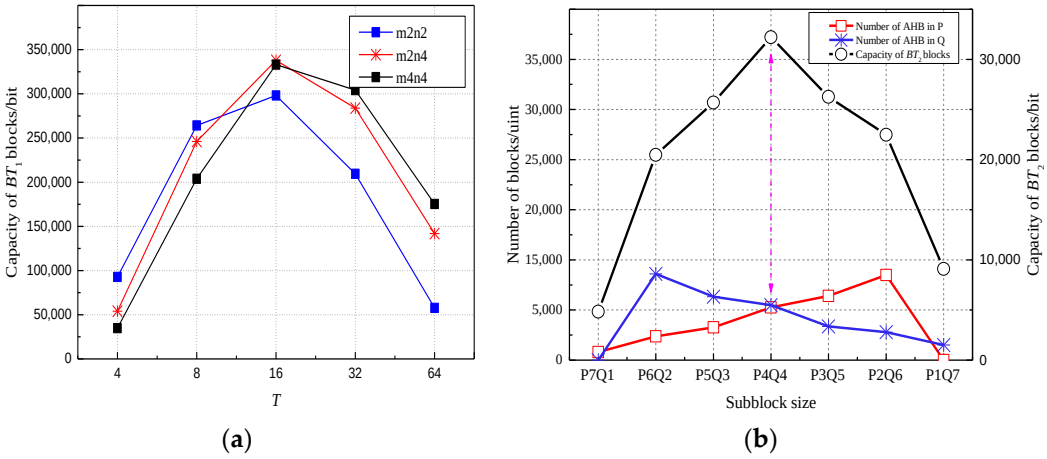


Figure 5. Influence of parameters on embedding capacity: (**a**) the effect of the block segmentation method and different thresholds on capacity and (**b**) the effect of the sub-block segmentation method on capacity.

Figure 5b shows the embedding ability of the BT_2 block in the sub-block segmentation method (T = 16, m = 2, n = 4). The horizontal axis represents the segmentation method, and the vertical axis represents two data types. The left Y axis represents the number of BT_2 blocks in the P and Q sub-blocks, and the right Y axis represents the pure embedding number of BT_2 blocks under different sub-block segmentation methods. $Pm2Qn2$ refers to two sub-blocks and their length. Where $m2 + n2 = 8$ and $m2, n2 \in [1,7]$. Sub-blocks $Q1$ and $P1$ with length 1 have no embedded capacity. The figure shows that the number of AHBs increases with the decrease in P length from 7 to 2, and Q has the same trend because the smaller the block length, the easier it is to meet the hidden conditions. It can be observed that the $P4Q4$ mode corresponds to the highest embedding capacity. Therefore, $m2 = 4$ and $n2 = 4$ are the second-round segmentation method, which also means $l_2 = 4$. According to the embedding strategy selection conditions in Table 1, $l_2 = 4 < n_f + 1 = 5$. Therefore, AMLE is applied to BT_1 blocks ($l_1 = 8$) and FMLE is applied to BT_2 blocks ($l_2 = 4$) to maximize the embedded capacity.

Figure 6 shows the number and proportion of each block type with T = 16 as an example. It can be found that in the image with a smooth texture, the proportion of AHBs is high, such as Peppers (78%), Lena (76%), and Airplane (75%). In the complex image Baboon, the proportion of AHBs is 32%, and the proportion of UHBs is 68%, which is the highest among several images. Generally, the higher the proportion of AHBs, the stronger the embedding ability of the cover image.

The scheme is verified on the experimental images according to the above parameters, and the cover image of each stage is shown as follows:

Figure 7 shows two examples of the results of each stage in the processing process. The process of all cover images is similar. Column 1 shows the original cover images (a) and (f). Then, the original cover images are encrypted using stream cipher to obtain figures (b) and (j). Next, according to the Kd, the secret data are embedded to obtain the encrypted images containing secret data (c) and (h). After transmission, the receiver decrypts the received images to obtain figures (d) and (i). Figure 7e,j are the final recovered images. Visually, the encrypted images and the transmission image are messy, improving the security of the secret data and the content of the cover image at the same time. The experimental data also show that the PSNR of the encrypted image and the transmission image are between 8-10dB, which has a very low correlation with the original image. Meanwhile, the decrypted and recovered images are highly similar to the original cover images.

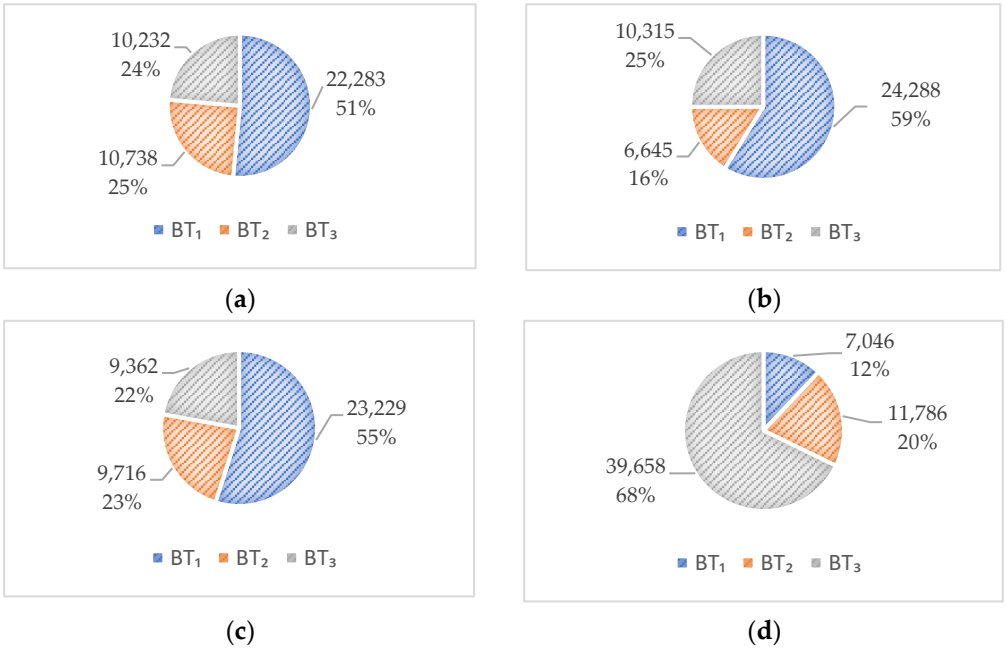


Figure 6. Schematic diagram of the proportion of different types of image blocks: (**a**) Lena, (**b**) Airplane, (**c**) Peppers, and (**d**) Baboon.

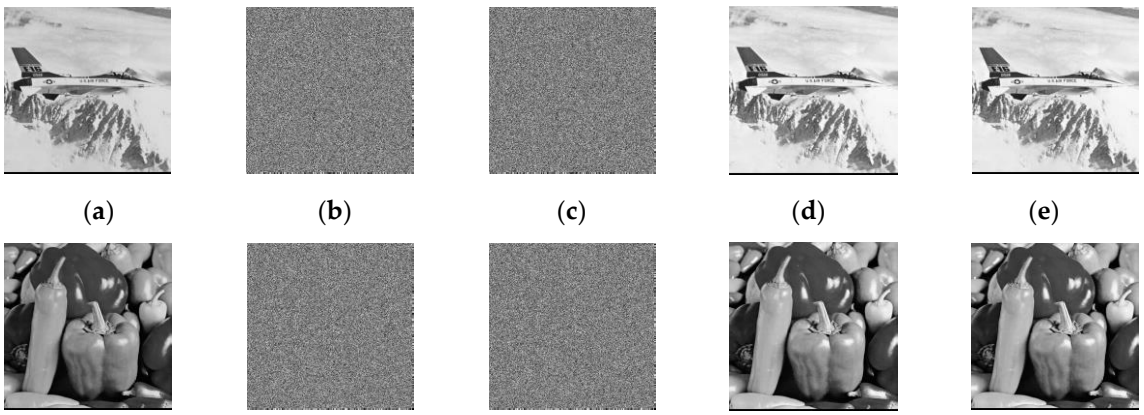


Figure 7. Results of the proposed scheme. (**a**–**e**) describe the process for Airplane; (**f**–**j**) describe the process for Peppers.

Table 3 shows the experimental results of the proposed scheme with different T. The table describes the pure embedding rate, decryption quality, and the recovered image quality with different T. It can be found that the embedding rate increases first and then decreases with T from 4 to 64. The first three images achieve the maximum embedding rate when $T = 16$. However, the last image reaches the maximum at $T = 32$. From the average point of view, $T = 16$ is optimal. Note that in the experimental results of the Baboon image, two lines are blank. This is because the algorithm requires embedding a large amount of auxiliary data to achieve separable reception and lossless recovery, and Baboon images cannot provide such a large space under these two parameters. At the same time, the decryption quality decreases with the increase in T, but the decryption quality is also acceptable at the maximum embedding rate. As for quality of the recovered image, the sign "+∞" means that it can be entirely consistent with the original cover image.

Table 3. Experimental results with different T.

Image	T	Embedding Rate /bpp	PSNR of the Decrypted Image/dB	PSNR of the Recovered Image/dB
Lena	4	0.32	47.06	+∞
	8	1.47	40.16	+∞
	16	1.83	34.92	+∞
	32	1.8	30.54	+∞
	64	1.64	26.73	+∞
Airplane	4	1.27	46.27	+∞
	8	2.03	41.08	+∞
	16	2.22	36.65	+∞
	32	2.19	32.12	+∞
	64	2.06	27.55	+∞
Peppers	4	0.02	47.75	+∞
	8	1.25	40.12	+∞
	16	1.79	34.44	+∞
	32	1.75	30.86	+∞
	64	1.6	27.58	+∞
Baboon	4	/	/	/
	8	/	/	/
	16	0.26	35.75	+∞
	32	0.54	28.54	+∞
	64	0.48	22.46	+∞

To verify the performance, the maximum embedding capacity is compared with the various algorithms in Table 4. In these algorithms, the maximum embedding rate that can be achieved in works [14,15,18,20,22–24] does not exceed 1 bpp, but it does in works [19,25,34], and the proposed scheme is between 1–2 bpp. Chen et al. [19] used block encryption and achieved stable capacity by lossless compression in blocks of 2×2 size. Wu et al. [25] used three rounds of block segmentation to obtain scalable sub-blocks according to the average value of the blocks to provide embedding space. Wang et al. [34] embedded data into the multiple MSBs of the embeddable pixels according to the strong correlation. After two-round segmentation, the proposed scheme combines two strategies to obtain the best embedding rate. The maximum embedding rates of the experimental images are 1.83 bpp, 2.22 bpp, 1.79 bpp, and 0.54 bpp. However, the embedding rate of complex image Baboon is low due to two factors. One is that the number of AHBs in the image with a complex texture is small. The second is that the amount of auxiliary data is relatively large.

Table 4. Comparison of the maximum embedding capacity for different schemes (/bpp).

Image	Zhang [14]	Hong [15]	Kim [18]	Chen [19]	Wang [20]	Xu [22]	Ma [23]	Qin [24]	Wu [25]	Wang [34]	Proposed
Lena	0.024	0.015	0.25	1.4	0.88	0.811	0.886	0.0325	1.192	1.796	1.83
Airplane	0.019	0.015	0.25	1.57	0.72	0.796	0.918	0.0325	1.22	1.832	2.22
Peppers	0.016	0.015	0.25	1.36	0.82	0.663	0.624	0.0325	1.173	1.844	1.79
Baboon	0.005	0.015	0.25	0.47	0.23	0.377	0.741	0.0325	0.316	0.484	0.54
Average	0.016	0.015	0.25	1.2	0.663	0.662	0.792	0.0325	0.975	1.489	1.6

Overall, combining the FLME strategy and the AMLE strategy is more helpful in improving capacity, especially for the image with a smooth texture. Compared with the above schemes, the proposed algorithm has a larger capacity than previous algorithms.

4.2. Image Quality Analysis

The receiver needs to decrypt and restore the transmission image. Below is an analysis and comparison of the recovery quality of the cover image.

Figure 8 shows the results of related schemes. Zhang [14] made a data extraction error in the non-smooth area, which affected image recovery. Zhang [17] and Qin et al. [24] used the MSB of adjacent pixels for data decryption and recovery. Xu et al. [22] and Ma [23] used traditional RDH methods such as histogram shift for hiding and data recovery. Yi et al. [30] realized data recovery through multi-round extraction of block-level prediction error. The decryption performance of the proposed scheme in this paper is about 30–65 dB on the image with a smooth texture. At 1 bpp, the range of PSNR fluctuates from 40 dB to 50 dB and can be maintained at about 30 dB+ at the maximum embedding rate. On the Baboon image with a complex texture, its decryption performance decreases. Still, it is relatively stable as a whole and is slightly lower than 30 dB at the maximum embedding rate, which does not affect the use. At the same time, in [14,17,24], the cover can be completely reversible only when certain conditions are met. However, Refs [22,23,30] and the proposed scheme can realize the real reversibility of the cover image. In general, when the PSNR is greater than 30 dB, it is considered to have good visual quality. Therefore, the proposed scheme can achieve better decryption performance than other schemes.

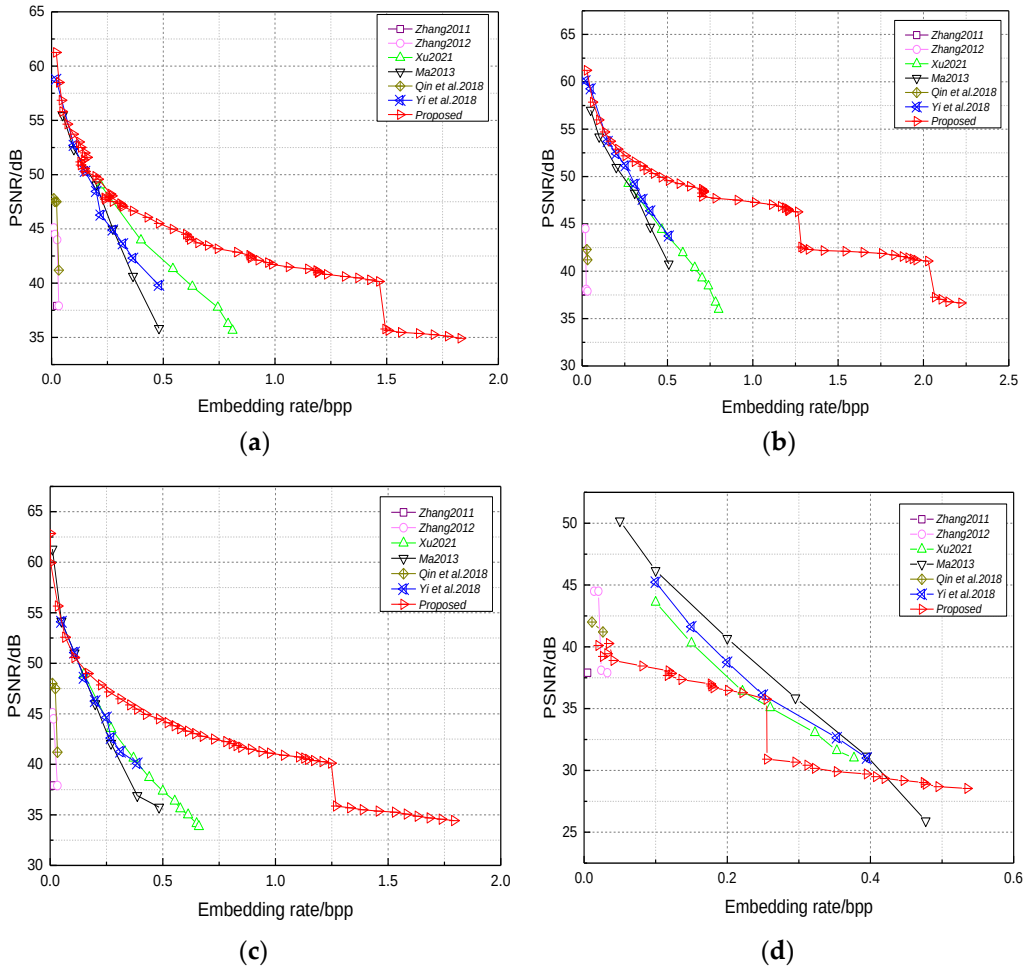


Figure 8. Performance between decryption quality and embedding rate compared with schemes Zhang [14], Zhang [17], Xu [22], Ma [23], Qin [24] and Yi [30]. (**a**) Lena, (**b**) Airplane, (**c**) Peppers, and (**d**) Baboon.

Table 5 shows other metrics comparisons of the proposed scheme with the related works. Column 2 gives a comparison of the separability of each scheme. Among them, schemes [14–16,30] adopted the joint method at the receiver, but the other schemes adopted the separable method to achieve more flexibility. During the data recovery extraction phase, the data accuracy cannot be fully guaranteed in [14–16], and the recovered image is lossy.

However, the data can be extracted accurately in [17]. The common point of [14–17] is that the scheme can realize the lossless recovery of the cover image only if the block size satisfies certain embedding conditions. Compared with the above schemes, the data-error-free extraction and the cover image lossless recovery can be guaranteed in the rest schemes and the proposed scheme.

Table 5. Comparison of other metrics.

Schemes	Separability	Error in the Extracted Data	Error in the Recovered Image	Reserving Room
Zhang [14]	No	Yes	Yes	No
Hong et al. [15]	No	Yes	Yes	No
Liao et al. [16]	No	Yes	Yes	No
Zhang [17]	Yes	No	Yes	No
Kim [18]	Yes	No	No	No
Chen et al. [19]	Yes	No	No	No
Wang et al. [20]	Yes	No	No	No
Xu et al. [22]	Yes	No	No	No
Ma et al. [23]	Yes	No	No	Yes
Qin et al. [24]	Yes	No	No	Yes
Wu et al. [25]	Yes	No	No	Yes
Yi et al. [30]	No	No	No	Yes
Wang et al. [34]	Yes	No	No	Yes
Weng et al. [41]	No	No	No	No
Proposed	Yes	No	No	Yes

In addition, in the works [14–39] and the proposed scheme, secret data are embedded in a single encrypted cover image. In the works [40–42], secret data are embedded in multiple shared and encrypted images. In contrast, the works using secret sharing technology belong to the VRAE framework and have great potential in terms of embedding capacity. However, the amount of data during transmission increased several times, and the receiver needs multiple transmission images to recover the original cover image. In the proposed scheme and works [14–39], the focus is more on improving the load capacity while ensuring that the amount of data transmitted remains unchanged and recovering secret data and the cover image through a single transmission image.

Therefore, combined with the experimental data in Table 4 and Figure 8, and the above analysis, it can be seen that the proposed scheme is superior in terms of embedding ability, image recovery quality, and separability compared with the related works.

5. Conclusions

This paper focuses on improving embedding capacity and recovery performance in encrypted images and proposes a separate RDH-EI scheme. The scheme can achieve complete error-free extraction of secret data, high decryption of the cover image, and realize lossless recovery of the cover image with auxiliary data. It is mainly supported by the following four points. (1) The scheme divides the cover image into blocks of different types through two-round processing, maximizing the embeddable space as much as possible. (2) The mixed embedding strategy adopts different methods for blocks of different types, further increasing the pure capacity and hiding more secret data. (3) the spatial correlation of the original image is utilized to improve the quality of the directly decrypted image and obtain a high-quality decrypted image. (4) The auxiliary data help the receiver to achieve separable data extraction and increase the flexibility of the scheme.

Compared to the state of-the-art works, the proposed scheme is superior in terms of embedding capacity, decryption quality, and separability. It can be applied in data privacy protection, data hiding, efficient transmission, and other scenarios. In further study, the research focus will be on that how to reduce auxiliary data and improve the embedding capacity to realize the transmission of secret data more efficiently and safely.

Author Contributions: Conceptualization, resources, writing—review and editing, supervision, project administration and funding acquisition, are provided by Q.Z.; methodology, software, validation, formal analysis, investigation, data curation, writing—original draft preparation and visualization, are provided by R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National State Key Laboratory Fund (6142411432107, 6142811204306, 61428411184406).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article as no new data were created or analyzed in this study.

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

References

1. Alexander, G.C.; Gürcü, Ö. Adaptive to pixel value and pixel value difference irreversible spatial data hiding method using modified LSB for grayscale images. *J. Inf. Secur. Appl.* **2023**, *70*, 103314. [CrossRef]
2. Shi, Y.Q.; Li, X.L.; Zhang, X.P.; Wu, H.T.; Ma, B. Reversible data hiding: Advances in the past two decades. *IEEE Access* **2016**, *4*, 3210–3237. [CrossRef]
3. Celik, M.U.; Sharma, G.; Tekalp, A.M.; Saber, E. Lossless generalized-LSB data embedding. *IEEE Trans. Image Process* **2005**, *14*, 253–266. [CrossRef] [PubMed]
4. Zhang, T.C.; Weng, S.W.; Wu, Z.J.; Lin, J.; Hong, W. Adaptive encoding based lossless data hiding method for VQ compressed images using tabu search. *Inf. Sci.* **2022**, *602*, 128–142. [CrossRef]
5. Abbasi, R.; Xu, L.X.; Amin, F.; Luo, B. Efficient lossless compression based reversible data hiding using multilayered n-bit localization. *Secur. Commun. Netw.* **2019**, *2019*, 8981240. [CrossRef]
6. He, W.G.; Xiong, G.Q.; Wang, Y.M. Reversible data hiding based on multi-predictor and adaptive expansion. *IET Image Process* **2022**, *16*, 888–899. [CrossRef]
7. Mehbodniya, A.; Douraki, B.; Webber, J.L.; Alkhazaleh, H.A.; Elbasi, E.; Dameshghi, M.; Zitar, R.A.; Abualigah, L. Multi-layer reversible data hiding based on the difference expansion method using multilevel thresholding of host images based on the slime mould algorithm. *Processes* **2022**, *10*, 858. [CrossRef]
8. Nguyen, T.S.; Huynh, V.T.; Vo, P.H. A novel reversible data hiding algorithm based on enhanced reduced difference expansion. *Symmetry* **2022**, *14*, 1726. [CrossRef]
9. He, W.; Cai, Z. Reversible data hiding based on dual pairwise prediction-error expansion. *IEEE Trans. Image Process* **2021**, *30*, 5045–5055. [CrossRef]
10. Bai, Y.Q.; Jiang, G.Y.; Zhu, Z.J.; Xu, H.Y.; Song, Y. Reversible data hiding scheme for high dynamic range images based on multiple prediction error expansion. *Signal Process Image Commun.* **2021**, *91*, 116084. [CrossRef]
11. Hung, C.C.; Lin, C.C.; Wu, H.C.; Lin, C.Y. A study on reversible data hiding technique based on three-dimensional prediction-error histogram modification and a multi-layer perceptron. *Appl. Sci.* **2022**, *12*, 2502. [CrossRef]
12. Wang, Y.M.; Xiong, G.Q.; He, W.G. High-capacity reversible data hiding in encrypted images based on pixel-value-ordering and histogram shifting. *Expert. Syst. Appl.* **2023**, *211*, 118600. [CrossRef]
13. Weng, S.W.; Tan, W.L.; Ou, B.; Pan, J.S. Reversible data hiding method for multi-histogram point selection based on improved crisscross optimization algorithm. *Inform. Sci.* **2021**, *549*, 13–33. [CrossRef]
14. Zhang, X.P. Reversible data hiding in encrypted image. *IEEE Signal Process Lett.* **2011**, *18*, 255–258. [CrossRef]
15. Hong, W.; Chen, T.S.; Wu, H.Y. An improved reversible data hiding in encrypted images using side match. *IEEE Signal Process Lett.* **2012**, *19*, 199–202. [CrossRef]
16. Liao, X.; Shu, C.W. Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. *J. Vis. Commun. Image R.* **2015**, *28*, 21–27. [CrossRef]
17. Zhang, X.P. Separable reversible data hiding in encrypted image. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 826–832. [CrossRef]
18. Kim, C. Separable reversible data hiding in encrypted AMBTC images using Hamming code. *Appl. Sci.* **2022**, *12*, 8225. [CrossRef]
19. Chen, M.K.; Chang, C.C. High-capacity separable reversible data hiding method in encrypted images based on block-level encryption and Huffman compression coding. *Connect. Sci.* **2021**, *33*, 975–994. [CrossRef]
20. Wang, Y.M.; Cai, Z.C.; He, W.G. High capacity reversible data hiding in encrypted image based on intra-block lossless compression. *IEEE Trans. Multimed.* **2021**, *23*, 1466–1473. [CrossRef]
21. Ge, H.L.; Chen, Y.; Qian, Z.X.; Wang, J.J. A High Capacity Multi-Level Approach for reversible data hiding in encrypted images. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 2285–2295. [CrossRef]
22. Xu, D.W.; Su, S.B. Reversible data hiding in encrypted images with separability and high embedding capacity. *Signal Process Image Commun.* **2021**, *95*, 116274. [CrossRef]

23. Ma, K.D.; Zhang, W.M.; Zhao, X.F.; Yu, N.H.; Li, F.H. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 553–562. [CrossRef]
24. Qin, C.; Zhang, W.; Cao, F.; Zhang, X.P.; Chang, C.C. Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection. *Signal Process* **2018**, *153*, 109–122. [CrossRef]
25. Wu, H.B.; Li, F.Y.; Qin, C.; Wei, W.W. Separable reversible data hiding in encrypted images based on scalable blocks. *Multimed. Tools Appl.* **2019**, *78*, 25349–25372. [CrossRef]
26. Qiu, Y.Q.; Cai, C.H.; Zeng, H.Q.; Feng, G.; Lin, X.D.; Qian, Z.X. Joint reversible data hiding in encrypted images with the self-correcting ability. *J. Xidian Univ.* **2021**, *48*, 107–116. [CrossRef]
27. Panchikkil, S.; Manikandan, V.M.; Zhang, Y.D. A pseudo-random pixel mapping with weighted mesh graph approach for reversible data hiding in encrypted image. *Multimed. Tools Appl.* **2022**, *81*, 16279–16307. [CrossRef]
28. Li, F.Y.; Zhu, H.J.; Yu, J.; Qin, C. Double linear regression prediction based reversible data hiding in encrypted images. *Multimed. Tools Appl.* **2020**, *80*, 2141–2159. [CrossRef]
29. Motomura, R.; Imaizumi, S.; Kiya, H. A reversible data hiding method with prediction-error expansion in compressible encrypted images. *Appl. Sci.* **2022**, *12*, 9418. [CrossRef]
30. Yi, S.; Zhou, Y.C.; Hua, Z.Y. Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion. *Signal Process Image Commun.* **2018**, *64*, 78–88. [CrossRef]
31. Puteaux, P.; Puech, W. An efficient msb prediction-based method for high-capacity reversible data hiding in encrypted images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1670–1681. [CrossRef]
32. Tasi, Y.Y.; Liu, H.L.; Kuo, P.L.; Chan, C.H. Extending multi-msb prediction and huffman coding for reversible data hiding in encrypted HDR images. *IEEE Access* **2022**, *10*, 49347–49358. [CrossRef]
33. Yi, P.Y.; Yin, Z.X.; Qian, Z.X. Reversible data hiding in encrypted images with two-msb prediction. In Proceedings of the 2018 IEEE International Work-shop on Information Forensics and Security (WIFS), Hong Kong, China, 11–13 December 2018. [CrossRef]
34. Wang, D.W.; Zhang, X.Q.; Yu, C.Q.; Tang, Z.J. Reversible data hiding in encrypted image based on multi-msb embedding strategy. *Appl. Sci.* **2020**, *10*, 2058. [CrossRef]
35. Xiao, D.; Wang, Y.; Xiang, T.; Bai, S. High-payload completely reversible data hiding in encrypted images by an interpolation technique. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 1732–1743. [CrossRef]
36. Wang, J.J.; Li, G.X.; Xia, G.E.; Sun, Z.R. A separable and reversible data hiding algorithm in encrypted domain based on image interpolation space. *Acta Electron. Sin.* **2020**, *48*, 92–100. [CrossRef]
37. Chen, F.; Yuan, Y.; He, H.J.; Tian, M.; Tai, H.M. Multi-msb compression based reversible data hiding scheme in encrypted images. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 905–916. [CrossRef]
38. Singh, O.P.; Singh, A.K. Data hiding in encryption–compression domain. *Complex Intell. Syst.* **2021**, 1–14. [CrossRef]
39. Kittawi, N.; Al-haj, A. Reversible data hiding using bit flipping and histogram shifting. *Multimed. Tools Appl.* **2022**, *81*, 12441–12458. [CrossRef]
40. Chen, Y.C.; Hung, T.H.; Hsieh, S.H.; Shiu, C.W. A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 3332–3343. [CrossRef]
41. Weng, C.Y.; Yang, C.H. Reversible data hiding in encrypted image using multiple data-hiders sharing algorithm. *Entropy* **2023**, *25*, 209. [CrossRef]
42. Chen, B.; Lu, W.; Huang, J.; Weng, J.; Zhou, Y. Secret sharing based reversible data hiding in encrypted images with multiple data-hiders. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 978–991. [CrossRef]
43. Available online: https://ccia.ugr.es/cvg/index2.php (accessed on 1 May 2023).