applied sciences

MDPI

Article

Reversible Data Hiding in Encrypted Image Based on Multi-MSB Embedding Strategy

Dewang Wang, Xianquan Zhang *, Chunqiang Yu * and Zhenjun Tang

Guangxi Key Lab of Multi-Source Information Mining & Security, Guangxi Normal University, Guilin 541004, China; dewang_wang@126.com (D.W.); zjtang@gxnu.edu.cn (Z.T.)

* Correspondence: zxq6622@163.com (X.Z); yu_chunqiang@126.com (C.Y)

Received: 19 February 2020; Accepted: 15 March 2020; Published: 18 March 2020

Abstract: In this paper, a reversible data hiding method in encrypted image (RDHEI) is proposed. Prior to image encryption, the embeddable pixels are selected from an original image according to prediction errors due to adjacent pixels with strong correlation. Then the embeddable pixels and the other pixels are both rearranged and encrypted to generate an encrypted image. Secret bits are directly embedded into the multiple MSBs (most significant bit) of the embeddable pixels in the encrypted image to generate a marked encrypted image during the encoding phase. In the decoding phase, secret bits can be extracted from the multiple MSBs of the embeddable pixels in the marked encrypted image. Moreover, the original embeddable pixels are restored losslessly by using correlation of the adjacent pixels. Thus, a reconstructed image with high visual quality can be obtained only when the encryption key is available. Since exploiting multiple MSBs of the embeddable pixels, the proposed method can obtain a very large embedding capacity. Experimental results show that the proposed method is able to achieve an average embedding rate as large as 1.7215 bpp (bits per pixel) for the BOW-2 database.

Keywords: reversible data hiding; high capacity; encrypted image; multi-MSB embedding strategy

1. Introduction

Image encryption and data hiding are two main means for data security. The former aims to transform the meaningful image into a noise-like one to prevent image content leakage [1–4], while the latter embeds secret data into a cover image imperceptibly. In image encryption, the original image is the one to be protected, while, in data hiding, the secret data is the information that should be undisclosed. Traditional data hiding technology is usually irreversible, and the embedding process will bring permanent distortion to the original carrier, which is not accepted in some cases such as military images, medical images, and judicial evidence collection where the original carrier needs to be restored without distortion. Take into account requirements of lossless recovery of the original carrier, reversible data hiding (RDH) was proposed.

RDH technology has made great progress in the past decade. For example, early reversible data hiding technologies losslessly compress the least significant bit (LSB) planes or quantization residuals to accommodate for secret bits [5–8]. Later, Tian [9] proposed an RDH method using difference expansion (DE). This method divides an image into a series of pixel pair and embeds a secret bit into a pair of pixels by expanding the difference of this pixel pair. Based on the DE method, some improved reversible data hiding algorithms were proposed in [10–13]. Another classic reversible data-hiding algorithm is histogram shifting (HS) [14–21]. HS methods aim to generate a sharp histogram by counting pixel values [14], pixel difference [15,16], or prediction errors [17–21]. The peak bin is expanded for data embedding, and the other bins need to be shifted for reversibility. Among these methods, HS-based methods can achieve better embedding performance.

Nowadays, RDHEI has received increasing attention in the research community, in which both the original image and secret data need to be protected. In RDHEI, image encryption and data embedding are accomplished by different users separately. The content-owner first encrypts the original image to a noise-like one according to encryption key, then the data hider embeds secret data into the encrypted image using a data-hiding key without knowing its original content. The receiver can perform different operations according to the available keys. This can be applied in many scenarios such as Cloud storage, medical image management system.

In general, RDHEI can be divided into two categories, namely, vacating room after encryption (VRAE) [22–31] or reserving room before encryption (RRBE) [32–39]. Puech et al. [22] used an AES (advanced encryption standard) encryption algorithm to encrypt the original image, and then randomly selected a location in each 4×4 pixel block to embed the secret bits. In order to extract secret bits during the decryption phase, they performed a local standard deviation analysis of the marked image. The embedding rate (ER) of this method is very small, i.e., only 0.0625 bpp. Zhang [23] encrypted the original image using the stream cipher firstly and then divided the encrypted image into blocks and each pixel block was divided into two parts. In one part, the three LSBs of each pixel were flipped to embed a secret bit. On the receiver side, a fluctuation function was designed for data extraction and image recovery. Yu et al. [24] improves the method of Zhang et al. [23], so that the visual quality of decrypted images is improved. Wu et al. [25] adopted a prediction error to introduce two RDH methods in the encryption domain, namely, joint method and separable method. In order to make full use of the spatial correlation in the original image, Li et al. [26] abandoned the idea of block segmentation and adopted the strategy of random diffusion, and the performance is better than that of Zhang [23]. In [27], Zhou et al. designed a method to embed secret bits through a public key mechanism. In the decoding processing, a powerful SVM (support vector machine) classifier with two classes is designed to distinguish encrypted and unencrypted image patches, so that the embedded bits and the original image can be decoded jointly. Qian et al. [28] introduced an RDH method based on distributed source coding. After stream encryption of an image, a series of pixels is selected from the encrypted image for compression to vacate room for embedding data. However, the entropy value of the image after encryption will reach a maximum, which makes it difficult to vacate room to embed bits, so that the embedding rate is usually very low. For this case, some special encryption-based methods are proposed. Xu et al. [29] designed a special encryption mode to encrypt the non-sampled interpolation error, and then used the improved histogram shifting and DE technology to embed data. Li et al. [30] divide the image into several crosses, the pixels of each cross are encrypted with the same key, and secret data is embedded by shifting the difference histogram of the encrypted image. Yu et al. [31] encrypted the original image with key transmission and calculated two-layer pixel errors to generate an error histogram with three peak bins. Data hiding is performed by shifting the error histogram.

In order to improve the embedding rate, Ma et al. [32] first proposed a RRBE-based method. They exploited the traditional RDH method to achieve self-embedding before encryption so as to obtain room for data embedding. The embedding rate of Ma et al. is higher than that of the previous methods. Zhang et al. [33] predicted some pixel values before encryption so that through histogram shifting to embed data in the predicted error, and a special encryption scheme is designed to encrypt the predicted error. Yi et al. [34] increased the embedding rate through improving the method of Zhang et al. [33] by using half of the pixels in the original image to predict the other half of the pixels. In order to better utilize the correlation between adjacent pixels, Cao et al. [35] considered a patch-level sparse representation which can reserve a large redundant room for embedding data in the encrypted image. Zhang et al. [36] encrypted the original image by using the public key encryption technology with homomorphism, and then embedded data into LSB of pixels in the encrypted image. In [37], a novel method of RDHEI was proposed, Nguyen et al. first divided the pixels into smooth and complex regions according to the four neighborhood pixels of the pixels, and then encrypted the original image. The secret data is embedded in the median plane of the smooth pixels of the encrypted image. Compared with VRAE-based RDHEI methods, these RRBE-based RDHEI methods have improved

the embedding capacity and the visual quality of decrypted images, but due to the limitation of the embedding method, the hiding capacity is not particularly large. Based on this, Puteaux et al. [38] designed a new MSB-prediction based on a method which embeds data in the MSB of the pixels. In [38], two approaches are proposed, which are the CPE-HCRDH approach (high-capacity reversible data hiding approach with correction of prediction errors) and the EPE-HCRDH approach (high-capacity reversible data hiding approach with embedded prediction errors). In the CPE-HCRDH approach, the embedding rate can reach 1 bpp for every image, and in the EPE-HCRDH approach, the embedding rate is slightly lower, but still between 0.8 bpp and 1 bpp for most tested images. Subsequently, on the basis of method in Puteaux et al. [38], Yi et al. [39] further improved the maximum embedding rate by using the method of embedding data at the two-MSB of pixels. For most tested images, the embedding rate was larger than 1 bpp.

Although [38,39] achieving a high embedding rate, only one MSB and two MSBs are exploited in [38] and [39], respectively. The utilization of MSBs is still unsatisfactory in these two methods and embedding rate still has improvement room. In this paper, an RDHEI method based on a multi-MSB embedding strategy is proposed. Integrating prediction technology with pixel correlation, the multiply MSBs can be exploited to improve embedding rate. The proposed scheme has the following highlights:

- By using a multi-MSB embedding strategy, the secret bits can be embedded in the encrypted images without any pixel oversaturation in the plaintext domain.
- More importantly, by using multi-MSB embedding strategy, secrets bits can be directly extracted from encrypted domain from the multi-MSB of pixels without any error. The reconstructed image with very high visual quality can be obtained only in the case that the encryption key is obtained.
- Compared with the other state-of-the-art methods [38,39], the proposed method can achieve a significantly higher maximum embedding rate.

The remainder of this paper is organized as follows. The details of the proposed method are introduced in Section 2. Section 3 describes how to choose the optimal parameters. Experimental results and performance comparison are presented in Section 4. Finally, the proposed method is concluded in Section 5.

2. Proposed Method

In this section, an RDH method in encrypted image is proposed, which includes image encryption, data hiding in encrypted image, data extraction, and image restoration. The proposed method is suitable for many classic scenes. For example, as shown in Figure 1, the content owner is a user of the cloud. In order to prevent the information of the image from being leaked, he encrypted the image before uploading it to the cloud. As the third party, the cloud can provide data embedding services to generate a marked encrypted image without knowing the original image content, that is, the cloud is the data hider of our method. The recipient is also a cloud user. He can get a marked encrypted image from the cloud and then extract data and decrypt the image. We introduce our method through the content owner, data hider, and recipient. The content owner conducts a series of operations including selecting embeddable pixels, generating a location map, rearranging the image, and encrypting the image. Data hider embeds secret bits into the encrypted image by data-hiding key without knowing the contents of the original images. At the receiving end, if the legal recipient only has a data-hiding key, then he can extract secret bits without error. If he only has the encryption key, then he can obtain a reconstructed image. If he has both data-hiding key and encryption key, then he can extract data without error and restore the image losslessly.

The outline of this section is as follows: selection of embeddable pixels is introduced in Section 2.1. Image encryption and data embedding are presented in Sections 2.2 and 2.3, respectively. Finally, data extraction and image recovery are introduced in Section 2.4.

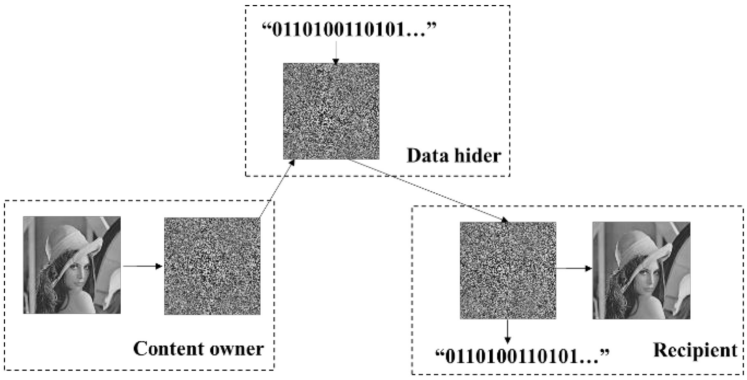


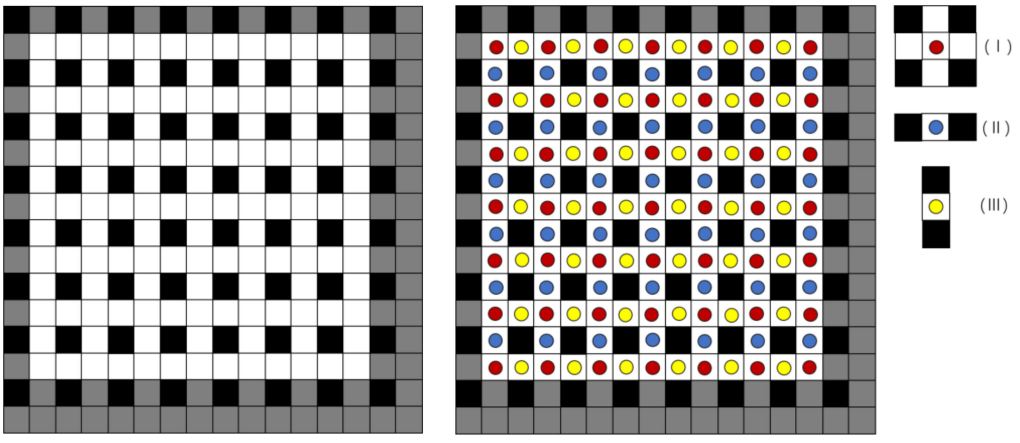
Figure 1. Tripartite relationship of the proposed method.

2.1. Selection of Embeddable Pixels

Suppose that the original image I is an 8-bit gray-scale image sized $H \times W$. Then the value of each pixel $I(i,j)$ belongs to [0,255] ($1 \le i \le H$, $1 \le j \le W$). As shown in Figure 2a, the content owner divides the original image into black, white, and grey parts. Then the black pixels are denoted as I_B, the white pixels are denoted as I_W, and the grey pixels at the edge are denoted as I_{ED}. For each white pixel, there are some black pixels around it. According to the distribution of these black pixels, namely, the number and location of the surrounding black pixels, we divide these white pixels into three cases, I, II, and III, as shown in Figure 2b. For case I, the white pixel has four diagonal black pixels surrounding it. For case II, two black pixels are located on the left and right of the white pixel, respectively. For case III, two black pixels are located on the upper and lower margins of the white pixel, respectively. Thus, the number of black pixels surrounding white pixels may be 2 or 4. Then, the local complexity of each white pixel $I_W(i,j)$ is calculated by Equation (1).

$$\Delta(i,j) = \sqrt{\frac{\sum_{k=1}^{u}\left(I_B^k(i,j) - I_{ave}(i,j)\right)^2}{u}},\ u = 2 \text{ or } 4 \tag{1}$$

where $I_B^k(i,j)$ ($k = 1, \ldots, u$) are the black pixels which are surrounding with $I_W(i,j)$, and $I_{ave}(i,j)$ is the average value of these surrounding pixels.



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

Figure 2. (**a**) The original image is divided into three part: black pixels, white pixels, and grey pixels, (**b**) three types of white pixels.

A large value of $\Delta(i,j)$ indicates that the current white pixel $I_W(i,j)$ is located in a relatively complex region. Thus, we define a threshold T that determines whether the current pixel is in a smooth region

or in a complex region. If $\Delta(i,j) \le T$, then the current pixel $I_W(i,j)$ is located in a smooth region, and we have $I_W(i,j) \in I_W^s$; otherwise, $I_W(i,j) \in I_W^c$, where I_W^s and I_W^c are two pixel sets which are located in a smooth region and complex region, respectively. In order to achieve accurate prediction for the current pixel, we only predict the pixels in I_W^s. For the white pixels in three cases, I, II, and III, we use their corresponding nearest black pixels to predict them. Suppose that $I'^s_W(i,j)$ is the prediction value of $I_W^s(i,j)$. According to different cases in Figure 2b, $I'^s_W(i,j)$ is predicted as

$$I'^s_W(i,j) = \begin{cases} \frac{I(i-1,j-1)+I(i-1,j+1)+I(i+1,j-1)+I(i+1,j+1)}{4} & \text{(I)} \\ \frac{I(i,j-1)+I(i,j+1)}{2} & \text{(II)} \\ \frac{I(i-1,j)+I(i+1,j)}{2} & \text{(III)} \end{cases} \quad (2)$$

According to prediction error between $I_W^s(i,j)$ and $I'^s_W(i,j)$, the pixels in I_W^s are classified into two categories, and one of which is used to embed data. The criterion for pixel classification is

$$\left| I_W^s(i,j) - I'^s_W(i,j) \right| \le 2^{m-1} - 1 \quad (3)$$

where $m \in \{3,4,5,6,7\}$. If the current white pixels satisfy Equation (3), it means that these pixels can be used for embedding data and denoted as $I_W^{s\prime}$. Otherwise, it means that these pixels cannot be used for embedding data and denoted as I_W^{s*}. Since secret data is embedded into the MSBs of the pixels to generate marked pixels in the proposed method, the MSBs of those pixels will be changed due to modification in the encoding phase. In order to guarantee reversibility, only the pixels which satisfy Equation (3) (i.e., $I_W^{s\prime}$) can be used to embed data. The proof process is as follows.

Note that the 1th~mth LSBs of the marked pixels keep unchanged; these original bit planes can be restored after decryption operation. While the (m+1)th ~ 8th MSBs of these pixels are changed after data embedding, we need to use the surrounding pixels to restore their original values in the decoding phase. Suppose that the directly decrypted value of marked pixel is X. Then the 1th~mth LSBs of X remain unchanged, and each bit of the (m+1)th ~ 8th MSBs of X may be 0 or 1, so there are 2^{8-m} different possible values for different combinations of the (m+1)th ~ 8th MSBs. Suppose that these different values are X_l (l = 1, … , 2^{8-m}), then one of them must be equal to the original pixel value. Let X_k ($1 \le k \le 2^{8-m}$) be equal to the original pixel value (i.e., $X_k = I_W^s(i,j)$). Then, according to Equation (3), X_k satisfies Equation (4):

$$\left| X_k - I'^s_W(i,j) \right| \le 2^{m-1} - 1 \quad (4)$$

Note that the black pixels surrounding $I_W^s(i,j)$ are not modified; thus, the original value of these pixels can be restored losslessly after directly decryption. Therefore, the same predicted value $I'^s_W(i,j)$ can be obtained according to Equation (2).

Next, we will prove that only one value of X_l (i.e., X_k) satisfies Equation (4). Here we assume that both two possible values X_{k_1}, X_{k_2}($1 \le k_1, k_2 \le 2^{8-m}$, $k_1 \ne k_2$) satisfy Equation (4); then:

$$\left| X_{k_1} - I'^s_W(i,j) \right| \le 2^{m-1} - 1 \quad (5)$$

$$\left| X_{k_2} - I'^s_W(i,j) \right| \le 2^{m-1} - 1 \quad (6)$$

According to Equation (5) and Equation (6), we have:

$$\left| X_{k_1} - X_{k_2} \right| \le 2^m - 2 \quad (7)$$

Note that the (1th~mth) LSBs of X_{k_1} and X_{k_2} are the same, and at least one bit of the (m+1)th~8th MSBs of X_{k_1} and X_{k_2} is different. We know that each bit plane in (m+1) th ~ 8th MSBs represents a decimal value which is equal to or larger than 2^m; thus the assumption and Equation (7) are invalid. Thus, only one value of X_l satisfies Equation (4), this value is equal to the original pixel value.

For example, if $m = 6$, the current pixel $I(i, j)$ belongs to I^s_W, $I(i-1, j) \in I_B$, $I(i+1, j) \in I_B$ and $I(i-1, j) = 104$, $I(i+1, j) = 112$, then:

$$I'^s_W(i, j) = \frac{104 + 112}{2} = 108$$

Suppose that the directly decrypted value of the marked pixel is X, and the 1th~6th LSBs of X are '101001'. Obviously, the 1th~6th LSBs of original pixel $I(i, j)$ are also '101001', and the two MSBs of $I(i, j)$ may be '00', '01', '10', or '11'. Then we can obtain the four possible values: 41('00101001'), 105('01101001'), 169('10101001'), 233('11101001'). It is obvious that only 105 satisfies Equation (4) among these four values, i.e.:

$$|105 - 108| \leq 2^5 - 1$$

Therefore, we can recover the original value 105 of the pixel $I(i, j)$.

Since the pixels in I^{s*}_W are not suitable for data embedding, these pixels should be marked. To address this problem, a location map LM is utilized which is a 0-1 matrix with same size as the original image. In the LM, the pixels in I^{s*}_W are marked with 1, and the other pixels are marked with 0.

2.2. Image Encryption

Actually, to construct the encrypted image, there are three steps: auxiliary information generation, image rearrangement, and image encryption. Firstly, the auxiliary information is generated which is needed at the decoding phase. Next, the pixels are rearranged to produce a rearranged image. Last, the rearranged image is encrypted to generate its final version.

1) Auxiliary information generation: In order to correctly extract data and restore the original image losslessly during the decoding phase, auxiliary information is essential, including: threshold T (8 bits), parameter m (3 bits), the number of pixels in $I^{s\prime}_W$: S (16 bits), and Location map LM. LM is compressed losslessly by arithmetic coding to reduce its size, and its compressed version is denoted as L_{clm} and the size of L_{clm} is l_{clm}. A parameter n (16 bits) is required to record the value of l_{clm} for extracting the auxiliary information in advance. And the sequence of parameters in auxiliary information is as follows: T, m, S, n, L_{clm}. Consequently, the total size of the auxiliary information is (43+ l_{clm}) bits.

2) Image rearrangement: As Figure 2a shows, the black pixels occupy a quarter of the image; hence, we arrange these black pixels on the top quarter of the image in raster scan order, and the different types of pixels belonging to I_W are arranged by the order of $I^{s\prime}_W \rightarrow I^{s*}_W \rightarrow I^c_W$ after I_B. Finally, the pixels in I_{ED} are arranged on the end of image. Figure 3 is a rearranged vision of the original image which is denoted as R.

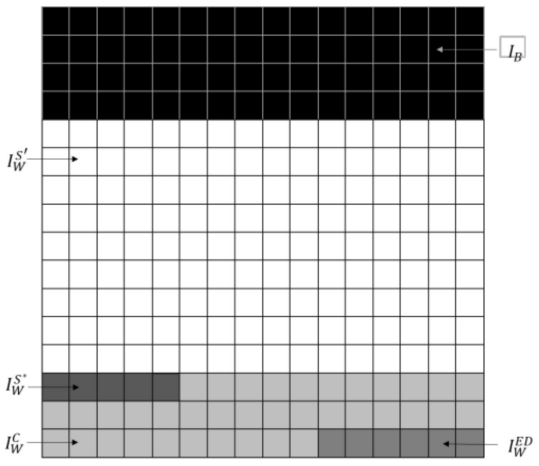


Figure 3. Rearranged vision of the original image.

3) Image encryption: In order to prevent the original image content leakage, it is encrypted with the encryption key K_e, which is generated by encryption algorithm RC4 [40] due to its security and efficiency. Let $R_l(i,j)$ be the lth (l = 1, 2, . . . , 8) bit of $R(i,j)$; then:

$$R_l(i,j) = \left\lfloor \frac{R(i,j)}{2^{l-1}} \right\rfloor mod\ 2 \tag{8}$$

where $\lfloor\cdot\rfloor$ is a floor function, and each encrypted bit $E_l(i,j)$ can be calculated by

$$E_l(i,j) = R_l(i,j) \oplus p_l(i,j) \tag{9}$$

where notation $\oplus$ denotes the exclusive-or operation, and $p_l(i,j)$ is the standard stream cipher generated by the encryption key K_e. Then the encrypted pixel value $E(i,j)$ can be calculated by:

$$E(i,j) = \sum_{l=1}^{8} E_l(i,j) \times 2^{l-1} \tag{10}$$

After that, the encrypted image E can be obtained.

2.3. Data Embedding in the Encrypted Image

After receiving the encrypted image E and auxiliary information, the data hider can embed secret bits into the image even if he/she does not know the content of the original image. Firstly, the bit planes of pixels which are encrypted in I_{ED} are replaced with auxiliary information, and then the replaced bit planes and secret bits are concatenated as embedded data, and the embedded data is embedded into the encrypted image. An overview of the process of encoding phase is presented in Figure 4.

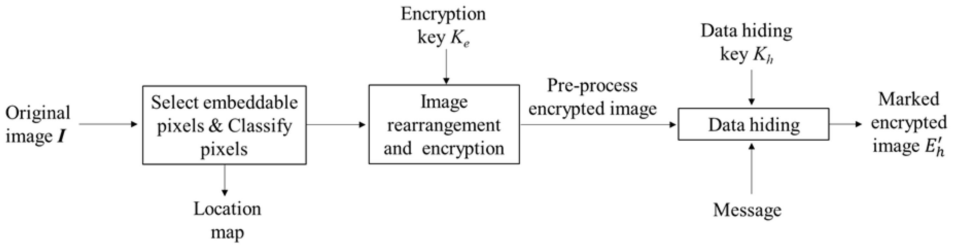


Figure 4. Overview of the encoding phase.

The number of pixels in $I_W^{s\prime}$ is S, which are immediately arranged behind the pixels of I_B. Thus, we can know which part of the pixels belong to $I_W^{s\prime}$. Then the data hider can use the data-hiding key K_h to pseudo-randomly select the pixels of $I_W^{s\prime}$ for data embedding. In the proposed method, the (m+1)th~8th MSBs of these selected pixels are directly replaced by embedded data to generate a marked encrypted image, and the values of the marked pixels are derived as:

$$E'_h(i,j) = \sum_{l=1}^{m} E_l(i,j) \times 2^{l-1} + \sum_{l=m+1}^{8} d_{l-m} \times 2^{l-1} \tag{11}$$

where d_{l-m} is a secret bit with a value of 0 or 1.

Each pixel of $I_W^{s\prime}$ can accommodate (8-m) bits of data, and the number of pixels in $I_W^{s\prime}$ is determined by m and T. Thus, the embeddable capacity of an image under fixed m and T is:

$$Cap = (8-m) \times S \tag{12}$$

Obviously, the maximum embeddable capacity of an image depends on the value of m and T. How to obtain the optimal values of m and T is introduced in Section 3.

2.4. Data Extraction and Image Recovery

In the decoding phase, the recipient can perform different operations according to the availability of encryption key and data hiding key. There are three possible scenarios:

1) The recipient only has the data hiding key K_h.
2) The recipient only has the encryption key K_e.
3) The recipient has both the data hiding key K_h and encryption key K_e.

An overview of the process of decoding phase is presented in Figure 5.

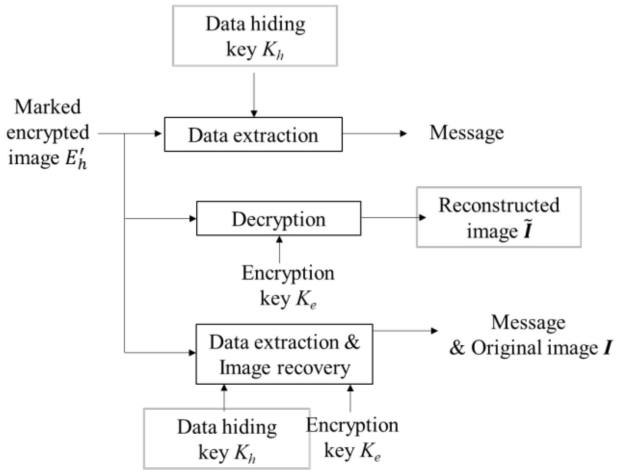


Figure 5. Overview of the decoding phase.

■In the first scenario, the recipient has the right to use the data hiding key to extracted data in the encrypted domain. First of all, the auxiliary information is extracted from the pixels in I_{ED} of the encrypted image. The recipient can obtain the position of the marked pixels according to data hiding key. Then $(8 - m)$ bits data can be extracted from marked pixels as follows:

$$d_{l-m} = \left\lfloor \frac{E'_h(i,j)}{2^{l-1}} \right\rfloor \bmod 2,\ l = m+1,..,8 \tag{13}$$

After every marked pixel is processed completely, all the data are extracted. Obviously, since the entire process is performed in the encrypted domain, it can effectively avoid the contents of the original image being leaked.

■In the second scenario, if the recipient only has encryption key K_e, then the reconstructed image $\widetilde{I}$ can be obtain, and the detailed process is as follows.

Step 1: The auxiliary information is extracted from the pixels in I_{ED} of the encrypted image.

Step 2: The pixels of the encrypted image are decrypted directly by using the encryption key:

$$D_l(i,j) = E'_l(i,j) \oplus p_l(i,j), l = 1,2,\ldots,8 \tag{14}$$

Then the pixel values after decryption are calculated as:

$$D(i,j) = \sum_{l=1}^{8} D_l(i,j) \times 2^{l-1} \tag{15}$$

where $E'_l(i,j)$ and $D_l(i,j)$ are the lth bit values of the marked pixel and decrypted pixel, respectively.

Step 3: The pixels are arranged to their original position. The process is as follows.

1) According to the previous scan order, the pixels of I_B and I_{ED} can be arranged to the original position.

2) The original positions of pixels in I_W are scanned according to raster scanning mode, and then $\Delta(i, j)$ is calculated one by one according to Equation (1). If $\Delta(i, j)>T$, then the pixel in this position belongs to I_W^C. By doing so, we can know the total number of pixels in I_W^C, and these pixels are rearranged in front of the pixels in I_{ED}; thus, it is easy to get the pixel positions in I_W^C in the rearranged image. Then the pixels in I_W^C can be arranged to original positions one by one, and the remaining positions of pixels in I_W belong to I_W^S.

3) From the auxiliary information, the compressed *LM* is decompressed to get the positions of the pixels in $I_W^{S^*}$. Then the pixels are arranged to their original positions according to the location map. Hence, the other positions belong to the pixels of $I_W^{S'}$. Therefore, all pixels can be arranged to their original position.

Step 4: Since some pixels in $I_W^{S'}$ are embedded with data and some pixels in I_{ED} are replaced by auxiliary information, the pixels of these two parts cannot be restored to original values after direct decryption. For the marked pixels in $I_W^{S'}$, their surrounding pixels in I_B have been restored after direct decryption. Then these surrounding pixels can be used to recover the original value of the pixels in $I_W^{S'}$, which has been introduced in Section 2.1. For the pixels in I_{ED} which are replaced by auxiliary information, their original pixel values cannot be restored without errors in this scenario. Therefore, we use the average value of their surrounding black pixels to restore the edged pixels in I_{ED}.

Finally, the reconstructed image $\widetilde{I}$ is obtained. The flow chart of steps is shown in Figure 6.

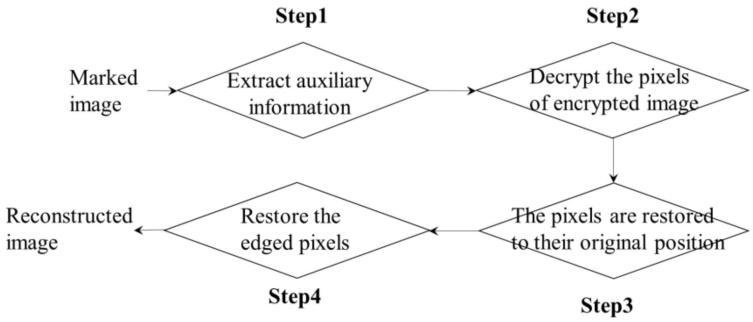


Figure 6. Flow chart of the steps.

Since most of pixels have been restored to their original values except for the edged pixels in I_{ED}, the image reconstructed $\widetilde{I}$ is approximated with the original image.

In the last scenario, if the recipient has both data-hiding key K_h and encryption key K_e, he can extract the embedded data and recover the original image without error.

The recipient uses the data-hiding key to extract data, which includes secret bits and the bit planes of pixels in I_{ED} which are encrypted. By using the encryption key, these encrypted bit planes can be decrypted directly. As introduced in the second scenario, only the pixels in I_{ED} cannot be restored to original values, but we can obtain the bit planes of these edged pixels in this scenario. All the pixels in I_{ED} can be restored to original values by replacing these bit planes. Therefore, all the pixels in the reconstructed image $\widetilde{I}$ can be restored losslessly in this scenario.

3. Details of Selection of Parameters

The proposed method will be tested on standard images with size of $512 \times 512 \times 8$. The maximum embedding rate and the peak signal-to-noise ratio (PSNR) of decrypted image are employed to evaluate the performance of the proposed method. To achieve the highest ER and PSNR, a few implement details of selecting parameters need to be introduced in advance.

3.1. Parameters of Maximum Embedding Rate

In this paper, the data is embedded into the multi-MSB of pixels, and the embedding capacity of an image is *Cap* which has been introduced in Section 2.3. In fact, the embedded data include secret bits and bit planes of edged pixels. Thus, the pure embedding capacity of secret bits is actually (*Cap*-43-l_{clm}) bits. Then, the pure embedding rate (*PER*) can be defined as:

$$PER = \frac{Cap - 43 - l_{clm}}{H \times W} \tag{16}$$

For each image, the maximum *PER* is determined by the value of m and T. In the proposed method, we use the parameter T to select the smooth pixels, and only the smooth pixels are predicted to choose the embeddable pixels. It is well-known that there are more smooth pixels and more embeddable pixels for the larger value of T. Meanwhile, it will generate more pixels in I_W^{s*} which are marked by the location map. The compressed location map is embedded into the edged pixels which is finite. Therefore, it is necessary to ensure that these edged pixels can accommodate the compressed location map and other auxiliary information. The number of edged pixels is 2043 and there are 16,344 bits which can accommodate auxiliary information. Thus, the following condition must be satisfied:

$$43+l_{clm} \leq 16344 \tag{17}$$

On the other hand, the value of m determines bit number that an embeddable pixel can accommodate. The smaller the m, the more data bits can be embedded into an embeddable pixel; however, the pixels that satisfy the embeddable condition would be fewer (i.e., Equation (3)).

From what has been discussed above, in order to obtain the maximum *PER* of an image, an optimal combination (m, T) need to be chosen. For each value of m (i.e., 3, 4, 5, 6, 7), the value of T gradually increases from a sufficiently small value of 1 to 50 at most with step size of 1 until the Equation (17) is not satisfied. For each combination (m, T) that satisfies Equation (17), we can obtain the *PER* by using Equation (16). Finally, we select an optimal combination (m, T) to achieve the maximum *PER* of an image.

3.2. Parameters of the Highest PSNR of Decrypted Image

As introduced in Section 2.4, after decrypting the image with only the encryption key, some pixels in I_{ED} cannot be recovered to original value since their bit planes were replaced by auxiliary information. Hence, the PSNR of the decrypted image is mainly affected by these unrecoverable edged pixels. Thus, under a given embedding rate *GER*, the smaller the amount of auxiliary information is, the fewer edged pixels need to be replaced, and the higher the PSNR is. Therefore, the optimal combination (m^*, T^*) is selected as Equation (18) to obtain the highest PSNR of the decrypted image under a given embedding rate.

$$\begin{cases} (m^*, T^*) = \arg\, min(l_{lcm}) \\ \quad s.t.\ \ PER \geq GER \end{cases} \tag{18}$$

4. Experimental Results and Analysis

In this section, we conduct several experiments to evaluate the proposed method. In Sections 4.1 and 4.2, we introduce the maximum embedding rate in tested images, and then a detailed example for the proposed method is presented. Finally, the proposed method is compared with other methods in Section 4.3.

4.1. The Maximum Embedding Rate for the Tested Images

We take ten standard images, including three publicly available image "Lena", "Airplane", "man", and seven images selected from the BOWS-2 database [41] to illustrate the pure embedding rate of

the proposed method. For convenience of description, we name these seven images F_1, F_2, ... , F_7, as shown in Figure 7. The maximum pure embedding rates of ten images under different values of m are shown in Table 1. From Table 1, we can see that in some cases, the embedding rate of some images is very low or even 0 when the value of m is small. This is mainly because the number of embeddable pixels will be few for moderate smooth images when the value of m is small. Meanwhile, it will generate more pixels to be marked. If the edged pixels are not enough to accommodate the auxiliary information, we do not embed secret bits in this scenario. Thus, the embedding rate is 0. For an image, when five different values of m are utilized, there are five different *PER*; and finally the largest one of the five *PER* is selected as the maximum *PER* of the image. It can be seen that different images may have different values of m when the maximum *PER* is obtained, which mainly depends on the smoothness of an image. For example, in the ten tested images, images F_1 and F_2 are smoother than other images, and the maximum pure embedding rates obtained when $m = 3$ were 3.5123 bpp and 3.0972 bpp, respectively. In general, the smoother the image, the more accurate the predicted pixel values will be; we can use a smaller value of m to obtain the maximum *PER*.

Figure 7. Seven tested images selected from the BOWS-2 database.

Table 1. Maximum *PER* of ten tested images under different values of m (bpp).

Image	$m = 3$	$m = 4$	$m = 5$	$m = 6$	$m = 7$	Maximum
Lena	0	0.5862	**1.7964**	1.4202	0.7404	**1.7964**
Airplane	0	1.4800	**1.8318**	1.3911	0.7298	**1.8318**
Man	0	0	1.1140	**1.3256**	0.7335	**1.3256**
F1	**3.5123**	2.9023	2.2132	1.4860	0.7439	**3.5123**
F2	**3.0972**	2.6224	2.0087	1.3796	0.7385	**3.0972**
F3	0	**1.1315**	1.1306	1.0010	0.6632	**1.1315**
F4	0	0	0.0621	0.4573	**0.6599**	**0.6599**
F5	0	0	0.6452	**0.9384**	0.6750	**0.9384**
F6	0	0	0.3930	0.3681	**0.5790**	**0.5790**
F7	0	**2.5708**	2.0930	1.4341	0.7327	**2.5708**

4.2. A Detailed Example for the Proposed Method

In this part, we apply the common image Lena to introduce the performance of the proposed method. Figure 8 shows the results. For the experiment results, the given embedding rate are 0.5 bpp and 0.9 bpp, and the optimal parameters (m^*, T^*) are (7, 5) and (6, 4), respectively. Figure 8a is the original image Lena. In Figure 8b, under (m^*, T^*) = (7, 5), the upper quarter of the rearranged image is composed of black pixels, and the appearance of the original image can be seen. The other three-quarters of the rearranged image are composed of various white pixels and edged pixels, which looks disordered. Figure 8c is the rearranged image under (m^*, T^*) = (6, 4). The rearranged images are encrypted to become noise-like, as shown in Figure 8d,e.

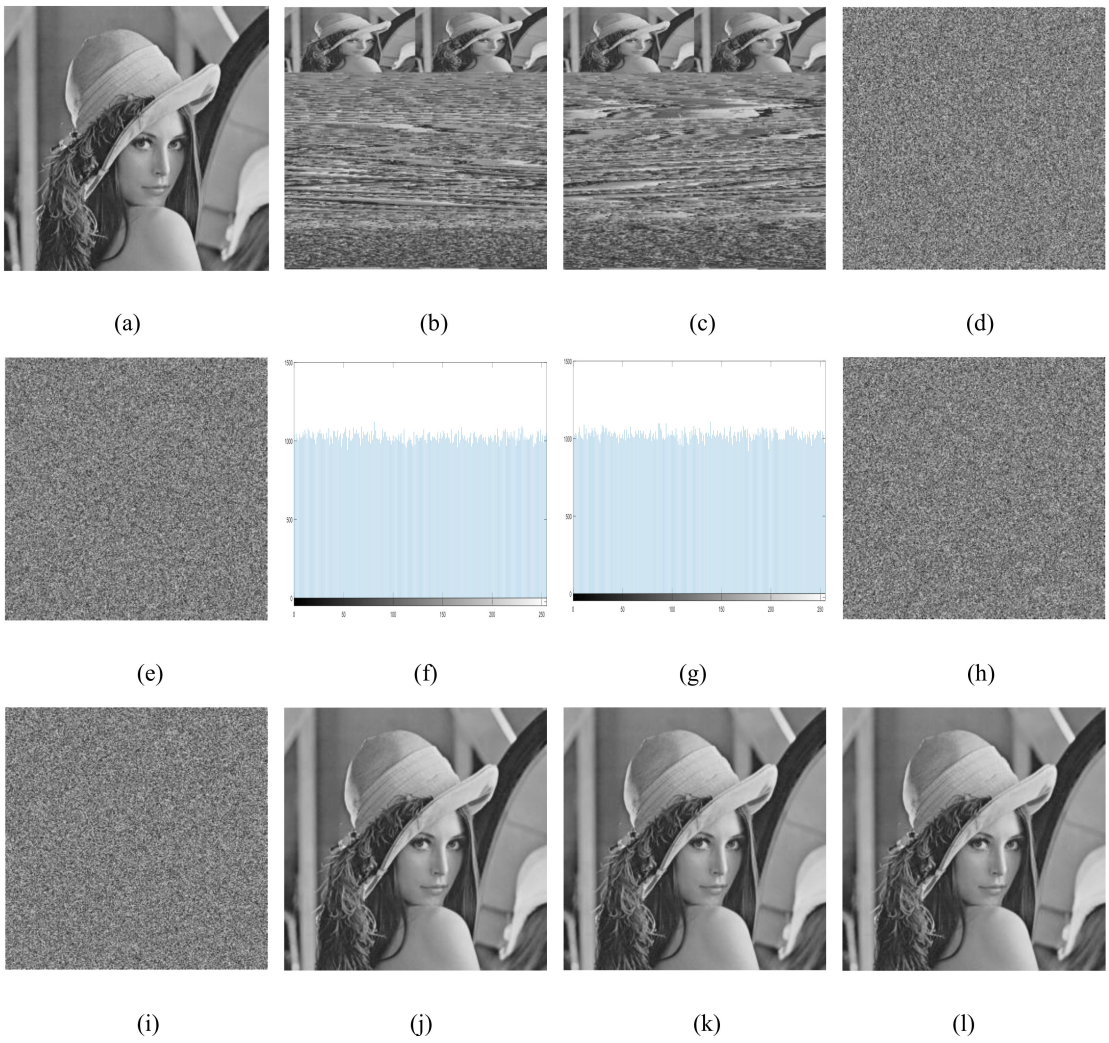


Figure 8. Experimental results for the proposed method. (**a**) Original image Lena. (**b**) Rearranged image with (m, T) = (7,5). (**c**) Rearranged image with (m, T) = (6,4). (**d**) The encrypted image of Figure 8b. (**e**) The encrypted image of Figure 8c. (**f**) Histogram of Figure 8d. (**g**) Histogram of Figure 8e. (**h**) Marked image with 0.5 bpp. (**i**) Marked image with 0.9 bpp. (**j**) Reconstructed image under 0.5 bpp with only encryption key. (**k**) Reconstructed image under 0.9 bpp with only encryption key. (**l**) Reconstructed image with both encryption key and data-hiding key.

It is well-known that each original image has its own histogram characteristic which can be used for retrieving original image. As shown in Figure 8f,g, the histograms of the encrypted images are nearly uniformly distributed, which can well protect the content of the image to withstand the statistical attack. In addition, we utilize correlation of adjacent pixels and information entropy to illustrate

image encryption security further. To test the correlation between two adjacent pixels, 10,000 pairs of two horizontally, vertically, and diagonally adjacent pixels from an image are randomly selected, respectively, and then the corresponding correlation coefficient r_{xy} of each pair is calculated using the following equations:

$$cov(x,y) = E\{(x - E(x))(y - E(y))\} \tag{19}$$

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{20}$$

where x and y are values of the two adjacent pixels in the image, $E(x)$ is the mean value of x, and $D(x)$ is the variance of x. If r_{xy} is close to 1, x and y have a strong correlation. Otherwise, if r_{xy} is close to 0, it indicates there is no correlation between x and y. Table 2 shows the comparison of adjacent pixel correlation between the original image and Figure 8d,e. It is obvious that the values of pixel correlation are extremely close to 0. It indicates encryption in our method is able to highly disorganize correlation of pixels in three directions.

Table 2. Correlation coefficients of two adjacent pixels in the original image and Figure 8d,e.

Direction	Original Image	Figure 8d	Figure 8e
horizontal	0.9688	−0.0155	0.0077
vertical	0.9842	0.0044	0.0242
diagonal	0.9565	−0.0104	0.0027

The entropy is the most outstanding feature of the randomness. The information entropy $H(X)$ of a message source X can be calculated as

$$H(X) = -\sum_{i=0}^{L-1} P(x_i) \log_2 P(x_i) \tag{21}$$

where $X=\{x_0, x_1, \ldots, x_{L-1}\}$ and $P(x_i)$ is the probability of x_i. If the information entropy is close to the maximum value, it means the encrypted image acquires excellent properties of randomness. For a grayscale image, x_i is an integer with the range of [0,255]. More specifically, each pixel in gray scale image can be encoded by 8 binary bits, and the ideal value of information entropy is 8. Information entropies of Figure 8d,e are 7.9831 and 7.9754, respectively. In summary, excellent histogram distribution, correlation coefficient, and entropy of encrypted images indicate the encryption method is secure in the proposed method.

After image encryption, secret bits are embedded into the multi-MSB of pixels in the proposed method, it will greatly change the value of pixels, but there is no need to consider the visual damage of the image due to data embedding in the encrypted image. Figure 8h,i shows the marked encrypted image when the pure embedding rate are 0.5 bpp and 0.9 bpp, respectively. In the proposed method, if only the data-hiding key is available, then the embedded data can be extracted from encrypted domain without any error. If only the encryption key is available, the reconstructed image after decryption is obtained, as shown in Figure 8j,k, which are reconstructed images after decryption. We can see that the two reconstructed images are approximate replications of the original ones, which is indicated by PSNRs of 85.98 dB and 74.25 dB, respectively. In fact, the reconstructed image is overwhelming similar to the original image in our proposed method, because only some edged pixels of the reconstructed image are changed compared with original value. If the data hiding key and encryption key are all available, then the recipient can not only extract the data without any error but also obtain the reconstructed image which is same as original image, as shown in Figure 8l. Obviously, in this case, the PSNR of the reconstructed image is approximating $+\infty$.

In order to obtain the best PSNR of the decrypted image with only the decryption key for a different given embedding rate, the optimal parameters (m^*, T^*) need to be chosen. As shown in Table 3,

for the image Lena, the optimal parameters are obtained under a different given embedding rate. From Equation (18), we can know that if the auxiliary information is smaller, the PSNR of the decrypted image will be higher. Generally, on the premise that the pure embedding rate is higher than given embedding rate, the larger the value of m, the fewer pixels need to be marked (i.e., the pixels in I_W^{S*}), which means when the amount of auxiliary information is smaller, then the PSNR of the reconstructed image after decryption is higher. It is observed from Table 3, when the given embedding rate increases gradually, the optimal value of m decreases in order to ensure that the pure embedding rate is higher than the given embedding rate.

Table 3. Optimal parameters (m^*, T^*) under different *GER* (bpp) of image Lena.

GER	0.1	0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7
parameters	(7,1)	(7,2)	(7,5)	(7,19)	(6,4)	(6,6)	(6,12)	(5,5)	(5,7)

4.3. Comparisons with Related Methods and Analysis

In order to demonstrate superiority of the proposed method, we compare the proposed method with different framework methods [31,35,36,38,39] in terms of maximum pure embedding rate and reconstructed image quality with only encryption key. It is noted that the method by Yu et al. [31] is based on the VRBE framework, while the other methods [35,36,38,39] are based on the BBRE framework. In particular, two MSB prediction methods proposed by Puteaux et al. [38] and Yi et al. [39] are overwhelmingly related with the proposed method. Overall, the five compared methods are all about the RDH algorithm in the encryption domain.

The proposed method and another two methods proposed in [38] and [39] are all based on an MSB embedding strategy. Thus, we compared with these two methods in terms of maximum embedding rate. We use the eight well-known images of Lena, Airplane, Man, Crowd, Baboon, Hill, Peppers, and Lake for comparison. The comparison results are as shown in Table 4. For the approach of Puteaux et al. CPE [38], the original image is modified in order to avoid all prediction errors. We can see that the maximum embedding rate is 1 bpp for each tested image because every pixel of the image can accommodate a bit in this approach. However, it will cause some damage to the quality of the reconstructed image. In the approach of Puteaux et al. EPE [38], the information about the error location is recorded in the encrypted image, thus the embedding capacity is slightly smaller than the approach of Puteaux et al. CPE [38]. However, during the decoding phase, the original image can be recovered losslessly. For the method proposed by Yi et al. [39], different from the two approaches in [38], two bits can be embedded into one pixel, which can improve the embedding rate especially in smooth images. In the methods proposed in [38] and [39], one or two bits of data can be embedded into a pixel at most, which limits the embedding rate. In the proposed method, the number of data bits that a pixel can accommodate increases to 5 (at most 5). For the images with different smooth levels, we can select the most suitable value of m to embed data which is more flexible than the methods in [38] and [39]. For our method and Yi et al. [39], the maximum embedding rate differences of eight tested images are relatively large, because the embedding capacity of these two methods has a great relationship with the smoothness of the image. When the image is more complex, the embedding capacity is smaller. For the image Baboon, which is more complex, the embedding rate is much smaller than that of other images. Compared with the two approaches in [38], the proposed method has a higher embedding rate in all tested images except for image Baboon, and the average embedding rate of the eight tested images increases by 0.4268 bpp and 0.4647 bpp, respectively. Compared with Yi et al. [39], the proposed method has a higher pure embedding rate in all eight tested images, and the average embedding rate increases by 0.4437 bpp.

Table 4. Maximum embedding rate comparison between the proposed method with methods [38,39] (bpp).

Image	Lena	Airplane	Man	Crowd	Baboon	Hill	Peppers	Lake	Average
Puteaux et al. CPE [38]	1	1	1	1	1	1	1	1	1
Puteaux et al. EPE [38]	0.9779	0.9817	0.9698	0.9830	0.8499	0.9919	0.9705	0.9722	0.9621
Yi et al. [39]	1.1408	1.2498	1.0111	1.0030	0.4093	1.0404	1.1131	0.8973	0.9831
Proposed	1.7964	1.8318	1.3256	1.5081	0.4838	1.3410	1.8443	1.2834	1.4268

In order to further demonstrate the superiority of the proposed method in maximum embedding rate, we compare our method with the methods of [38] and [39] on 10,000 tested images from the BOWS-2 database [41]. As shown in Table 5, our average embedding rate of 10,000 images is 1.7215 bpp, which is higher than 1 bpp and 0.9681 bpp of two approaches in Puteaux et al. [38] and 1.3512 bpp in Yi et al. [39]. Especially for those extremely smooth images, the proposed method can achieve a very high embedding rate. Ideally, nearly 3/4 of the pixels of the whole image can be used to embed data, and each pixel can be embedded into 5 bits of data. Thus, the highest embedding rate of an image is close to 3.75 bpp. As can be seen from Table 4, the highest embedding rate of the proposed method reaches 3.7140 bpp. But for those complex images, the embedding rate is not ideal, the worst case is only 0.1863 bpp. To better compare the embedding rate of different images, we randomly selected 500 images from the 10,000 images, and the embedding rate of these images were obtained. The experimental results are shown in Figure 9. Compared with the methods proposed by Puteaux et al. EPE [38] and Yi et al. [39], respectively, the proposed method had a significant improvement on most images in terms of embedding rate. For these 500 tested images, the embedding rates of 495 images in the proposed method were higher than those of Puteaux et al. EPE [38], and the embedding rates of 475 images were higher than that of Yi et al. [39]. For the proposed method, the embedding rates of most tested images were between 1.6 bpp and 2.7 bpp, which are satisfying. For Yi et al.'s method [39], the embedding rates were between 1.2 bpp and 1.7 bpp for most images, which are obviously lower than the proposed method. In Puteaux et al. EPE [38], the maximum embedding rates of most images were between 0.8 bpp and 1 bpp. We used three straight lines to indicate the average embedding rate of the 500 tested images for three methods. For the proposed method, the average embedding rate of 500 images was 2.0061 bpp. Correspondingly, the average embedding rates of [38,39] were 0.9736 bpp and 1.4531 bpp, respectively. For these 500 tested images, the proposed method was higher than Puteaux et al. EPE [38] and Yi et al. [39] by 1.0325 bpp and 0.5530 bpp, respectively. Overall, the experimental results demonstrate that the embedding rate of the proposed method is approximately 1.8 and 1.3 times those of the methods [38] and [39], respectively.

Table 5. Maximum embedding rate comparison between the proposed method and methods [38,39] on the BOW-2 database (bpp, 10,000 images).

Methods	Best Case	Worst Case	Average
Puteaux et al. CPE [38]	1	1	1
Puteaux et al. EPE [38]	1	0.3805	0.9681
Yi et al. [39]	1.992	0.115	1.3512
Proposed	3.7140	0.1863	1.7215

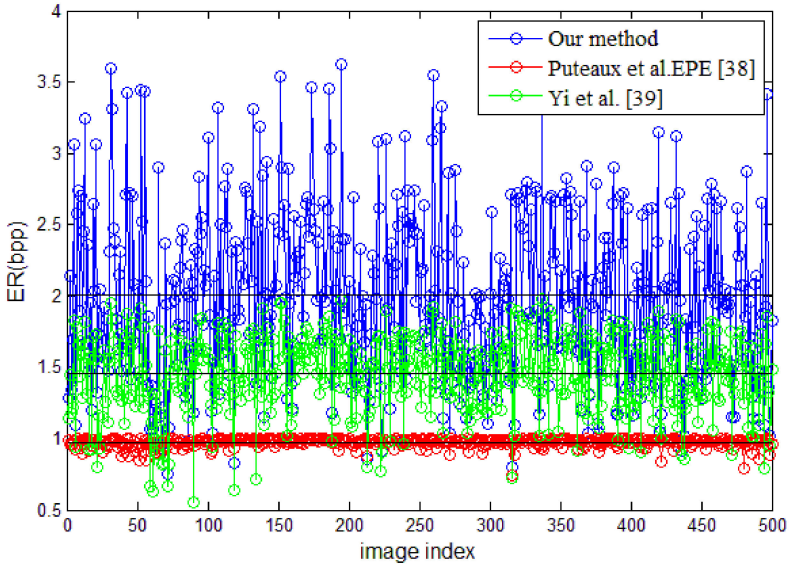


Figure 9. *ER* comparison with Puteaux et al. EPE [38] and Yi et al. [39] on 500 images, randomly selected from BOWS-2 database.

For RDH in the encrypted domain, the visual quality of the reconstructed image with only knowledge of the encryption key is an important evaluating index. Our method was compared with other relevant methods [31,35,36,38,39]. To do this, the well-known images of Lena, Airplane, and Man were used. Figure 10 shows the rate-distortion curves generated from the three tested images. For all tested images, since the approach of Puteaux et al. EPE [38] and the method of Yi et al. [39] do not need to use overhead, the reconstructed images are lossless (i.e., PSNR—>+∞). The method proposed in [35] considers patch-level sparse representation when embedding secret data. In addition, the learned dictionary is also embedded into the encrypted image. With the powerful representation of sparse coding, the large space can be vacated, so that the more secret bits can be embed in the encrypted image. As we can see from Figure 10, with the increase of embedding data, the PSNR of the decrypted image is obviously decreasing. For example, for the image Lena, when the embedding rate increased from 0.05 bpp to 0.75 bpp, the corresponding PSNR decreased from 54 dB to 30.8 dB. In [36], a lossless, reversible, and combined data-hiding method based on probability homomorphism is proposed. In the lossless scheme, the additional data is embedded into several least significant bit planes of ciphertext pixels by multi-layer wet paper coding, and the ciphertext pixels are replaced by new values. In the reversible scheme, the image histogram is reduced by preprocessing before image encryption, so that the modification of the encrypted image during data embedding will not cause pixel oversaturation in the plaintext domain. Because of the compatibility between lossless and reversible schemes, the two kinds of data embedding operations can be performed simultaneously in the encrypted image. For an image, the higher the embedding rate is, the higher the distortion of the decrypted image. In [31], a separable and error-free reversible data-hiding method for encrypted images based on two-layer pixel error is proposed. A histogram of the error of two adjacent layers of encrypted pixels is used to embed the secret data through histogram shifting to generate a labeled encrypted image. The embedding capacity is determined by the value of parameter K, which is used for determining which prediction errors are used to embed secret data. The higher the value of K, the more secret data can be embedded, but meanwhile, the decrypted image suffers more distortion. Although the PSNRs of reconstructed images of our method cannot reach +∞, they are all very high. For the reconstructed images Lena, Airplane, and Man, the PSNRs are between 64 dB and 86 dB, 63 dB and 66 dB, and 64 dB and 90 dB respectively, which are much higher than the approach of Puteaux et al. CPE [38] or methods [31,35,36]. In fact, only some edged pixels are changed and most of pixels can be recovered losslessly. The reconstructed image of our method is very close to the original image.

On the other hand, the proposed method can achieve a higher embedding rate than all the compared methods [31,35,36,38,39].

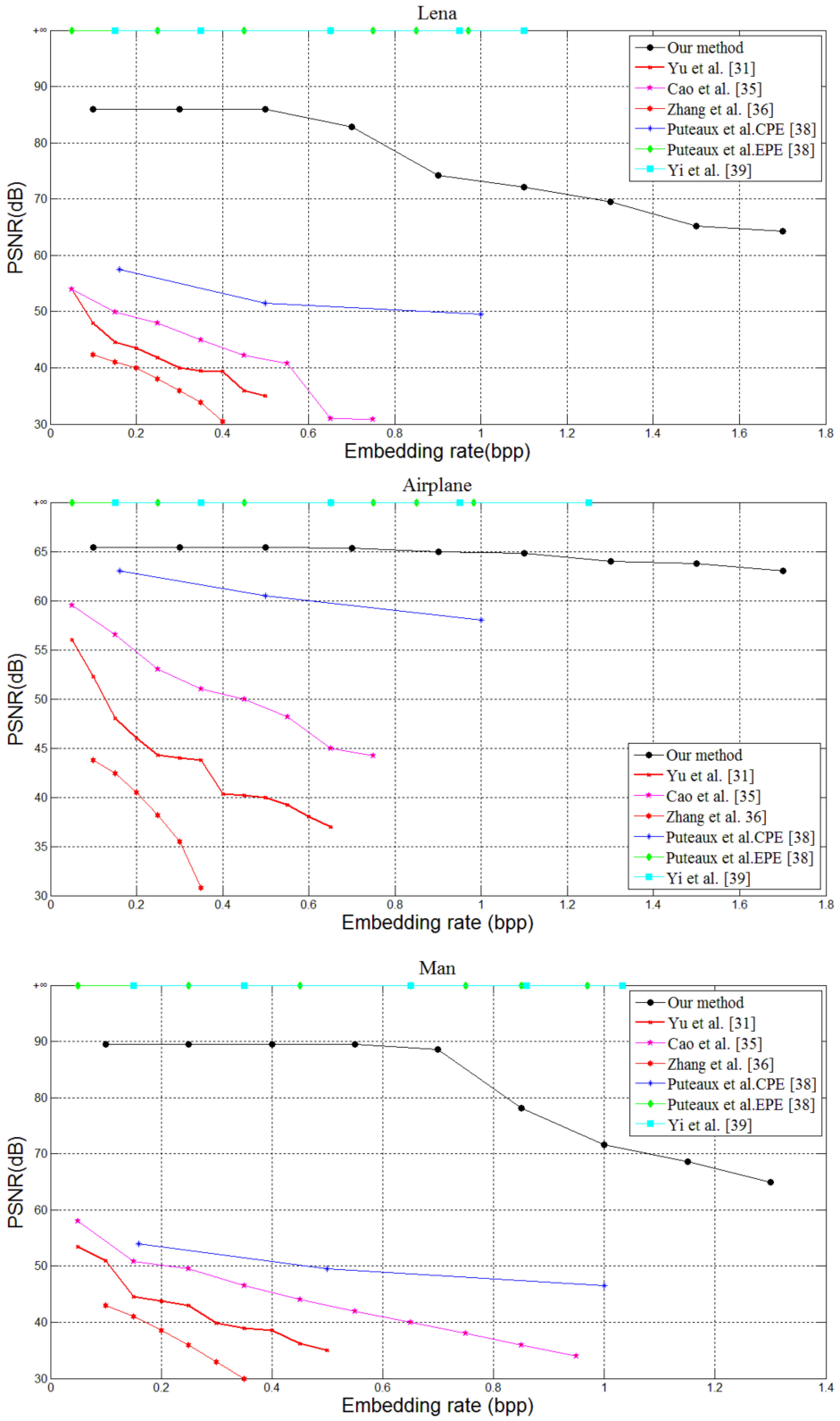


Figure 10. Performance comparisons for the tested images Lena, Airplane, and Man with the methods [31,35,36,38,39].

In summary, the proposed method is able to recover the embedded data and original image without errors and achieves an excellent trade-off between the embedding rate and the visual quality of the reconstructed image with only the encryption key.

5. Conclusions

In this paper, we proposed an efficient RDHEI by using multi-MSB embedding strategy with a very high embedding rate, which is much higher than the related methods [31,35,36,38,39]. In the proposed method, the values of m may be different for different images when the maximum embedding rates are obtained. In general, the smoother the image is, the smaller the value of m selected to obtain the maximum embedding rate. Under a certain embedding rate, we can select optimal values of (m, T) to obtain the highest PSNR of a decrypted image with only the encryption key. For the reconstructed image with only the encryption key, only some parts of edged pixels are damaged, and the other pixels of the reconstructed image are all restored to the original value. This means that within the maximum embeddable capacity, no matter how much data is embedded, the visual quality of the reconstructed image will not be significantly damaged, and most of the pixel values can be losslessly recovered. The experimental results show that the proposed method can achieve excellent embedding performance.

Author Contributions: D.W. and X.Z. conceived and designed the experiments; C.Y. performed the experiments; C.Y and Z.T. analyzed the data; X.Z contributed analysis tools; D.W. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61762017, Grant 61962008, and Grant 81701780, in part by the Guangxi Natural Science Foundation under Grant 2017GXNSFAA198222, and Grant 2017GXNSFBA198221, in part by the Project of Guangxi Science and Technology under Grant GuiKeAD17195062, in part by the Project of Guangxi "Bagui Scholar" Team for Innovation and Research, in part by the Guangxi Talent Highland Project of Big Data Intelligence and Application.

Acknowledgments: The authors would like to thank the anonymous referees for their valuable comments and suggestions.

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

References

1. Tang, Z.; Wang, F.; Zhang, X. Image encryption based on random projection partition and chaotic system. *Multimed. Tools Appl.* **2016**, *76*, 8257–8283. [CrossRef]
2. Tang, Z.; Song, J.; Zhang, X.; Sun, R. Multiple-image encryption with bit-plane decomposition and chaotic maps. *Opt. Lasers Eng.* **2016**, *80*, 1–11. [CrossRef]
3. Hua, Z.; Jin, F.; Xu, B.; Huang, H. 2D Logistic-Sine-coupling map for image encryption. *Signal Process.* **2018**, *149*, 148–161. [CrossRef]
4. Hayat, U.; Azam, N.A. A novel image encryption scheme based on an elliptic curve. *Signal Process.* **2019**, *155*, 391–402. [CrossRef]
5. Fridrich, J.; Goljan, M.; Du, R. Lossless Data Embedding—New Paradigm in Digital Watermarking. *EURASIP J. Adv. Signal Process.* **2002**, *2002*, 185–196. [CrossRef]
6. Xuan, G.; Zhu, J.; Chen, J.; Shi, Y.-Q.; Ni, Z.; Su, W. Distortionless data hiding based on integer wavelet transform. *Electron. Lett.* **2002**, *38*, 1646. [CrossRef]
7. Celik, M.; Sharma, G.; Tekalp, A.M.; Saber, E. Lossless generalized-LSB data embedding. *IEEE Trans. Image Process.* **2005**, *14*, 253–266. [CrossRef]
8. Celik, M.U.; Sharma, G.; Tekalp, A.M. Lossless watermarking for image authentication: A new framework and an implementation. *IEEE Trans. Image Process.* **2006**, *15*, 1042–1049. [CrossRef]
9. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. [CrossRef]
10. Alattar, A.M. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156. [CrossRef]
11. Coltuc, D.; Chassery, J.-M. Very Fast Watermarking by Reversible Contrast Mapping. *IEEE Signal Process. Lett.* **2007**, *14*, 255–258. [CrossRef]

12. Wang, X.; Li, X.; Yang, B.; Guo, Z. Efficient Generalized Integer Transform for Reversible Watermarking. *IEEE Signal Process. Lett.* **2010**, *17*, 567–570. [CrossRef]
13. Peng, F.; Li, X.; Yang, B. Adaptive reversible data hiding scheme based on integer transform. *Signal Process.* **2012**, *92*, 54–62. [CrossRef]
14. Ni, Z.; Shi, Y.; Ansari, N.; Su, W. Reversible data hiding. *Int. Symp. Circuits Syst.* **2003**, *16*, 354–362.
15. Li, X.; Zhang, W.; Gui, X.; Yang, B. A Novel Reversible Data Hiding Scheme Based on Two-Dimensional Difference-Histogram Modification. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1091–1100.
16. Nguyen, T.-S.; Chang, C.-C.; Huynh, N.-T. A novel reversible data hiding scheme based on difference-histogram modification and optimal EMD algorithm. *J. Vis. Commun. Image Represent.* **2015**, *33*, 389–397. [CrossRef]
17. Li, X.; Li, J.; Li, B.; Yang, B. High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Process.* **2013**, *93*, 198–205. [CrossRef]
18. Peng, F.; Li, X.; Yang, B. Improved PVO-based reversible data hiding. *Digit. Signal Process.* **2014**, *25*, 255–265. [CrossRef]
19. Ou, B.; Li, X.; Zhao, Y.; Ni, R. Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. *Signal Process. Image Commun.* **2014**, *29*, 760–772. [CrossRef]
20. Wang, D.; Zhang, X.; Yu, C.; Tang, Z. Reversible Data Hiding by Using Adaptive Pixel Value Prediction and Adaptive Embedding Bin Selection. *IEEE Signal Process. Lett.* **2019**, *26*, 1713–1717. [CrossRef]
21. He, W.; Cai, J.; Zhou, K.; Xiong, G. Efficient PVO-based reversible data hiding using multistage blocking and prediction accuracy matrix. *J. Vis. Commun. Image Represent.* **2017**, *46*, 58–69. [CrossRef]
22. Puech, W.; Chaumont, M.; Strauss, O. A reversible data hiding method for encrypted images. *Electronic Imaging* **2008**, *6819*.
23. Zhang, X. Reversible Data Hiding in Encrypted Image. *IEEE Signal Process. Lett.* **2011**, *18*, 255–258. [CrossRef]
24. Yu, J.; Zhu, G.; Li, X.; Yang, J. An Improved Algorithm for Reversible Data Hiding in Encrypted Image. In *Computer Vision*; Springer Science and Business Media LLC: Shanghai, China, 2012; Volume 7809, pp. 384–394.
25. Wu, X.; Sun, W. High-capacity reversible data hiding in encrypted images by prediction error. *Signal Process.* **2014**, *104*, 387–400. [CrossRef]
26. Li, M.; Xiao, D.; Peng, Z.; Nan, H. A Modified Reversible Data Hiding in Encrypted Images Using Random Diffusion and Accurate Prediction. *ETRI J.* **2014**, *36*, 325–328. [CrossRef]
27. Zhou, J.; Sun, W.; Dong, L.; Liu, X.; Au, O.C.; Tang, Y.Y. Secure Reversible Image Data Hiding Over Encrypted Domain via Key Modulation. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *26*, 441–452. [CrossRef]
28. Qian, Z.; Zhang, X. Reversible Data Hiding in Encrypted Image with Distributed Source Encoding. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *26*, 1. [CrossRef]
29. Xu, D.; Wang, R. Separable and error-free reversible data hiding in encrypted images. *Signal Process.* **2016**, *123*, 9–21. [CrossRef]
30. Li, M.; Xiao, D.; Zhang, Y.; Nan, H. Reversible data hiding in encrypted images using cross division and additive homomorphism. *Signal Process. Image Commun.* **2015**, *39*, 234–248. [CrossRef]
31. Yu, C.; Zhang, X.; Tang, Z.; Xie, X.; Xie, A. Separable and Error-Free Reversible Data Hiding in Encrypted Image Based on Two-Layer Pixel Errors. *IEEE Access* **2018**, *6*, 76956–76969. [CrossRef]
32. Ma, K.; Zhang, W.; Zhao, X.; Yu, N.; Li, F. Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 553–562. [CrossRef]
33. Zhang, W.; Ma, K.; Yu, N. Reversibility improved data hiding in encrypted images. *Signal Process.* **2014**, *94*, 118–127. [CrossRef]
34. Yi, S.; Zhou, Y.; Shuang, Y. An improved reversible data hiding in encrypted images. In Proceedings of the 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), Chengdu, China, 12–15 July 2015; pp. 225–229.
35. Cao, X.; Du, L.; Wei, X.; Meng, D.; Guo, X. High capacity reversible data hiding inencrypted images by patch-level sparse representation. *IEEE Trans. Cybernet* **2016**, *46*, 1132–1143. [CrossRef]
36. Zhang, X.; Long, J.; Wang, Z.; Cheng, H. Lossless and reversible data hiding in encrypted images with public-key cryptography. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*(9), 1622–1631. [CrossRef]
37. Nguyen, T.-S.; Chang, C.-C.; Chang, W.-C. High capacity reversible data hiding scheme for encrypted images. *Signal Process. Image Commun.* **2016**, *44*, 84–91. [CrossRef]

38. 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]
39. Puyang, Y.; Yin, Z.; Qian, Z. Reversible Data Hiding in Encrypted Images with Two-MSB Prediction. In Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, 11–13 December 2018; pp. 1–7.
40. Stalling, W. *Cryptography and Network Security: Principles and Practice*, 3rd ed.; Prentice-Hall: Upper Saddle River, NJ, USA, 2003.
41. Bas, P.; Furon, T. Image Database of BOWS-2. Available online: http://bows2.ec-lille.fr/ (accessed on 20 June 2017).