Hindawi
Security and Communication Networks
Volume 2023, Article ID 8723991, 15 pages
https://doi.org/10.1155/2023/8723991

Research Article

Reversible Data Hiding in Encrypted Image via Joint Encoding of Multiple MSB and Pixel Difference

Ping Kong,[1,2] Di Fu,[2] Lin Huang,[3] Liang Zhou,[1] Jian Li,[4] and Chuan Qin[3]

[1]*Shanghai Key Laboratory of Molecular Imaging, Jiading District Central Hospital Affiliated Shanghai University of Medicine and Health Sciences, Shanghai 201318, China*
[2]*School of Medical Instrument and Food Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*
[3]*School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*
[4]*School of Cyber Security, Qilu University of Technology (Shandong Academy of Sciences), Shandong Provincial Key Laboratory of Computer Networks, Jinan 250353, China*

Correspondence should be addressed to Jian Li; ljian20@gmail.com

Received 8 September 2022; Accepted 10 October 2022; Published 29 April 2023

Academic Editor: Feng Ding

Reversible data hiding in encrypted image (RDHEI) has become a research hotspot, which can effectively protect image content privacy. An RDHEI scheme based on the joint encoding of multiple MSBs (most significant bits) and pixel difference is proposed in this paper. A block-based image encryption method is adopted on the content owner side, which can securely protect the image contents while retaining the spatial correlation within each block. By the joint encoding strategy of multiple MSB and pixel difference, the redundancy within the bit plane is sufficiently compressed to accommodate more additional data; thus, a high embedding rate can be achieved. According to different kinds of available keys, image decryption and data extraction can be separably conducted on the receiver side. Experimental results show that our scheme can achieve a higher embedding rate than some state-of-the-art schemes.

1. Introduction

With the rapid development of internet, multimedia data have become the main communication carrier in daily life. However, multimedia data are vulnerable to tampering attacks during transmission; so, it is vital to determine whether multimedia data have been tampered with. The current schemes for identifying whether multimedia data have been tampered can be roughly divided into three categories: data embedding based schemes, which realize tampering detection by embedding additional data, i.e., the additional data represent the secret information that the data hider wishes to embed into carrier data before data transmission [1]; perceptual hashing based schemes, which realize tampering detection by comparing the hash values before and after tampering [2]; and forensics based schemes, which realize tampering detection by identifying traces left by tampering operations [3]. Among them, forensics-based methods involve the most extensive fields, such as linguistic text [4], facial information [5], and resampling [6]. With the development of forensics-based methods, a large number of antiforensics technologies [7] have emerged, and the confrontation between the two schemes has effectively promoted the healthy development of each other. However, this work focuses on a special data embedding-based scheme, i.e., reversible data hiding (RDH) [8], which belongs to the data embedding-based schemes. RDH allows the receiver to recover cover image with no distortion while extracting the additional data accurately. Based on this characteristic, RDH has become a research hotspot.

According to the embedding mechanism, classical reversible data hiding schemes are mainly divided into three major categories, i.e., lossless compression [9], difference expansion (DE) [10], and histogram shifting (HS) [11]. The

schemes based on lossless compression usually create room to embed data through image compression. The DE-based schemes embed data into the room vacated by expanding the difference of adjacent pixel values. After extensive research, DE has been developed for integer transformation (IT) [12–16] and prediction-error expansion (PEE) [17–26]. In the HS-based schemes, data embedding can be realized by shifting the peak point of image histogram. Besides, a novel reversible data hiding scheme based on code division multiplexing was proposed [27], which utilized the Walsh–Hadamard matrix to generate orthogonal spreading sequences, and the data can be overlappingly embedded without interfering. Therefore, multilevel embedding can improve the embedding capacity, and most elements of different spreading sequences will be mutually cancelled during the overlapping embedding process, which will maintain the image in good quality even with a high payload. However, RDH mentioned above is not suitable for scenes that require protection of the image ciphertext content. As a result, in order to protect both embedded data and the image content, RDH in encrypted images (RDHEI) has emerged. RDHEI schemes usually include three roles: content owner, data hider, and receiver. The content owner encrypts the cover image to protect the privacy by the encryption key and sends the encrypted image to the data hider. According to the vacated room in the encrypted image, the data hider utilizes the data hiding key to embed data. The receiver extracts the embedded data, decrypts the image, and finally recovers the original cover image.

Existing RDHEI schemes can be divided into two categories as follows: vacating room after encryption (VRAE) to embed data [28–41] and reserving room before encryption (RRBE) to embed data [42–51]. Zhang first introduced an RDHEI scheme belonging to VRAE in [28]. The image content was encrypted by stream encryption in this scheme. Then, the data hider embedded information by flipping LSBs (least significant bits). Finally, the receiver side extracted the information according to the image correlations. In [29], a side-match mechanism and a new smoothness evaluation method were introduced to increase the extraction and recovery accuracy. The scheme [30] reduced the number of pixels with flipped LSBs to improve the performance of rate distortion of the decrypted image. To increase the embedding rate and achieve complete reversible recovery, a separable RDHEI scheme which vacated room by compressing LSBs was proposed in [31], and the image decryption and data extraction can be separably conducted. To achieve better rate-distortion performance, Qin proposed a RDHEI scheme with separable capability and high quality of directly-decrypted image [32]. This scheme divided the encrypted block into two sets, i.e., the complex set and the smooth set, and then compressed the LSB plane corresponding to the smooth set to vacate more space for accommodating additional bits. In [33], an RDHEI scheme based on the prediction error was introduced and a better image visual quality can be obtained. An effective SVM classifier was adopted for image recovery and to improve the embedding rate [34]. In [35], conventional RDH algorithms can be conducted on encrypted images while the correlations between adjacent pixels were preserved. The scheme [36] transferred the redundancy of the original image to its encryption result and achieved a better performance of the embedding rate. In order to improve the security of [36], the algorithm in [37] proposed an encryption method and improved the embedding rate through the sparse encoding. Fu et al. [38] vacated room by compressing the blocks according to the occurrence frequency of MSB. The scheme [39] rearranged block-based MSB and vacated room by run-length coding. The scheme [40] utilized Huffman coding to compress the MSB layers with all values as same. With the rapid development of cloud storage and computing, cloud services are also threatened, i.e., encrypted images are vulnerable to tampering attacks from third parties when uploaded into the cloud. To tackle these challenges, an RDHEI via secret sharing based on GF (P) and GF (2^8) was proposed in [41]. This scheme first applied a specific encryption method through block and pixel permutation and Shamir's secret sharing. Only when sufficient shares were obtained, the secret message and original image can be recovered losslessly.

As the other type of the representative RDHEI scheme, reserving room before encryption (RRBE) can obtain a relatively high embedding rate. In [42], Ma et al. first presented an RDHEI scheme-based RRBE, in which the LSBs of part pixels are embedded into the remaining pixels with a conventional RDH method in the plaintext domain. Then, a prediction-based technique was proposed in [43]. While the preprocessing operations contain both block coding and conventional RDH, these schemes require a large computation for image owners. To reduce the computation, some methods based on prediction coding were proposed to vacate room [44–47]. To further increase the embedding capacity, a binary block embedding (BBE) compression method based on the bit plane was used to increase the embedding payload in [48]. In addition, the RDHEI can also be used to protect some special images, such as compressed images [49], medical images [50], and palette images [51].

In this paper, we propose an RDHEI scheme based on the joint encoding of multiple MSB and pixel difference, in which the MSBs bit plane and LSBs bit plane are efficiently encoded, respectively. For the MSBs bit plane, a multiple MSB encoding method is presented to compress MSBs for vacating more space. For the LSBs bit plane, a pixel difference encoding method is designed to compress the LSBs bit plane to make full use of image correlation. In general, the contributions of this paper include the following: (1) two efficient encoding methods are designed to spare more space to accommodate additional data; (2) image decryption and data extraction can be separably conducted; and (3) our scheme achieves a higher embedding rate than some state-of-the-art schemes.

The remainder of the paper is organized as follows. Section 2 describes our RDHEI scheme based on the joint encoding of multiple MSB and pixel difference in detail. Section 3 reports the experimental results, parameter analysis, and comparisons. Finally, conclusions are provided in Section 4.

2. Proposed Scheme

A reversible data hiding in encrypted images scheme based on the joint encoding of multiple MSB and pixel difference is proposed in this section. As shown in Figure 1, the framework of the proposed scheme consists of three phases: (1) content owner: the original image is encrypted by its content owner using the encryption key; (2) data hider: two novel methods are designed to vacate more space and then more additional data, i.e., the additional data represents the secret information that the data hider wishes to embed, which can be embedded with the data hiding key; and (3) receiver: data extraction, image decryption, and recovery can be conducted separably by the receiver through different kinds of available keys.

2.1. Image Encryption. In the image encryption stage, the original image I_o is first segmented into k nonoverlapping blocks sized $s \times s$. Then, according to the encryption key $K_e = \{K_e^{(1)}, K_e^{(2)}, K_e^{(3)}\}$, the bit plane XOR encryption within blocks, block scrambling encryption, and block-based pixel scrambling are performed to obtain the final encrypted image I_e. The image encryption process is illustrated in Figure 2.

2.1.1. Block-Based Bit Planes XOR. Denote $p_{i,j}$ as the decimal pixel value at the coordinate (i, j) in I_o, and u represents the u-th bit of $b_{i,j}$, where $1 \le i \le M$, $1 \le j \le N$ and $1 \le u \le 8$. That implies

$$b_{i,j,u} = \frac{p_{i,j}}{2^{u-1}} \bmod 2, u = 1, 2, \ldots, 8, \tag{1}$$

$$p_{i,j} = \sum_{u=1}^{8} b_{i,j,u} \bullet 2^{u-1}. \tag{2}$$

For each block of image I_o, the content owner adopts the subkey of $K_e^{(1)}$ to encrypt separately, while the number of blocks $k = \lfloor MN/s^2 \rfloor$, $K_e^{(1)}$ is composed of k subkeys $K_e^{(1),(1)}, \ldots, K_e^{(1),(t)}, \ldots, K_e^{(1),(k)}$, where t denotes the index of blocks. The subkey $K_e^{(1),(t)}$ is composed of 8 bits, i.e., $K_e^{(1),(t,1)}, \ldots, K_e^{(1),(t,u)}, \ldots, K_e^{(1),(t,8)}$, corresponding to the 8 bit planes in a block. Then, perform the bit plane XOR operation using the following equation:

$$b_{i,j,u}^{'} = K_e^{(1),(t,u)} \oplus b_{i,j,u}. \tag{3}$$

Thus, the encrypted decimal pixel value can be calculated by using the following equation:

$$p_{i,j}^{'} = \sum_{u=1}^{8} b_{i,j,u}^{'} \bullet 2^{u-1}. \tag{4}$$

2.1.2. Scrambling of Blocks and Pixels. After the block-based bit plane XOR, most areas of the image I_o are protected except for the contour part. To ensure the security of the image content, the image owner conducts the block scrambling and pixel scrambling operations, respectively. $K_e^{(2)}$ is a nonrepeated pseudorandom sequence with values ranging from 1 to k, and the locations of the k blocks are disordered with $K_e^{(2)}$. Similarly, $K_e^{(3)}$ consists of k nonrepeated pseudorandom sequences with values ranging from 1 to s^2, and the locations of all s^2 pixels in the block with the index u are scrambled with $K_e^{(2),(u)}$. After scrambling the pixels in all blocks, the final encrypted image I_e is produced. It can be found from the abovementioned encryption process that two encryption processes, i.e., the block-based bit plane XOR and pixel scrambling, were performed within the block. Therefore, we believe that the pixels in the encrypted block still retain a well correlation, and the correlation analysis of encrypted blocks will be first performed in the next subsection.

2.2. Data Hiding. Based on the abovementioned encryption method, it is worth noting that all encryption procedures are implemented based on blocks; thus, we believe that the pixel correlations of adjacent pixels in the encrypted block are retained. Therefore, we first perform a correlation analysis of encrypted blocks on the two datasets [52, 53] with the block size of 4×4, and the optimal block size will be discussed in detail in the experimental section, and then two efficient encoding methods are designed to compress encrypted blocks based on the correlation.

2.2.1. Correlation Analysis. By analyzing the correlation of encrypted image blocks, we find an interesting phenomenon: (1) the first few bit planes are usually uniform, i.e., all "0" or "1," and such bit planes are denoted as the "uniform bit plane" in our scheme; (2) the bits in the next bit plane of the uniform bit plane are usually sparse and denoted as the "nonuniform bit plane"; and (3) the difference between the decimal numbers of the rest bit planes is small and the rest bit planes are denoted as the "LSBs bit plane." The distribution of abovementioned bit planes is illustrated in Figure 3, where the number of bits occupied by the MSBs and LSBs bit plane is determined according to the uniform bit plane, nonuniform bit plane, and LSBs bit plane.

As shown in Figure 4, U_{num} represents the number of uniform bit plane, and the case of $U_{\text{num}} = 0$ only accounts for 12% and 17% of the two datasets [52, 53], respectively. The cases of $U_{\text{num}} \neq 0$ account for 88% and 83% of these two datasets, respectively, i.e., the vast majority of encrypted blocks contain the uniform bit plane. Therefore, an encoding method similar to run-length coding is designed to compress the uniform bit plane, i.e., several bits are utilized to represent the quantity of the uniform bit plane and some other bits are utilized to record the specific content of the uniform bit plane.

As shown in Figure 5, statistical experiments representing the sparsity of the nonuniform bit plane are also given on two datasets, where the parameter m represents the sparsity of the nonuniform bit plane, i.e., the smaller of the number "0" or "1" in the nonuniform bit plane, and the nonuniform bit plane is sparse only when $m = 1, 2, 3$, where the parameter m is determined by the parameter n_{T}

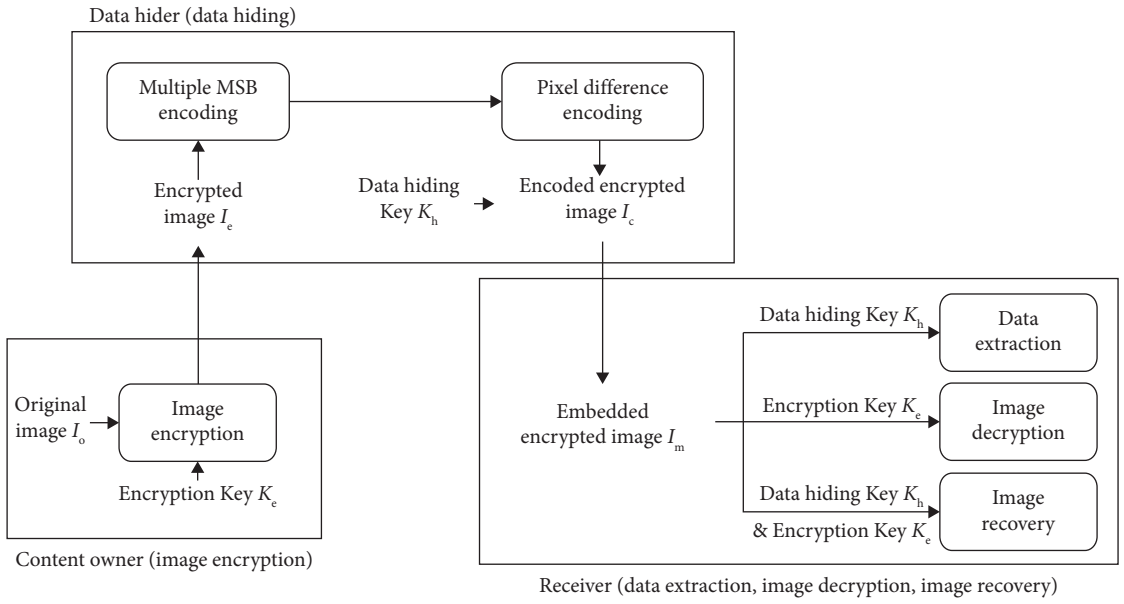


Figure 1: The flow chart of the proposed scheme.

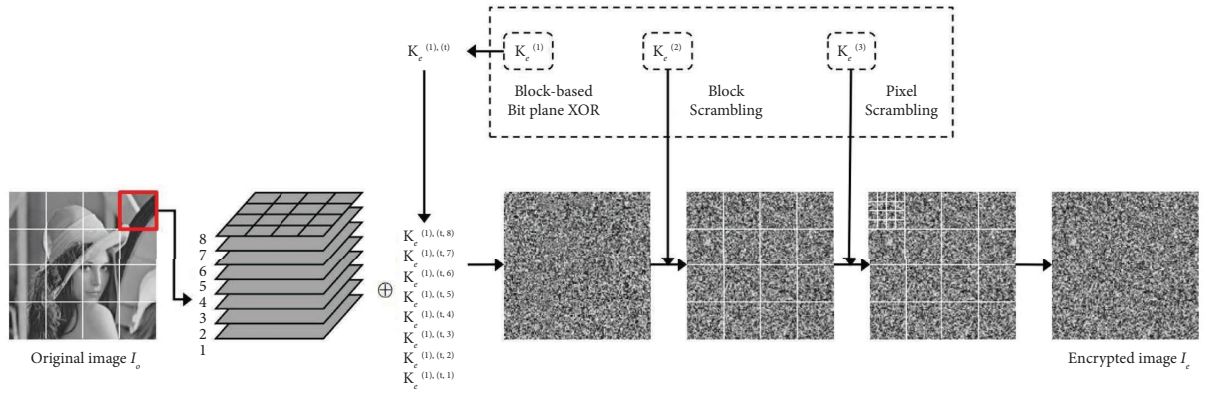


Figure 2: The image encryption process.

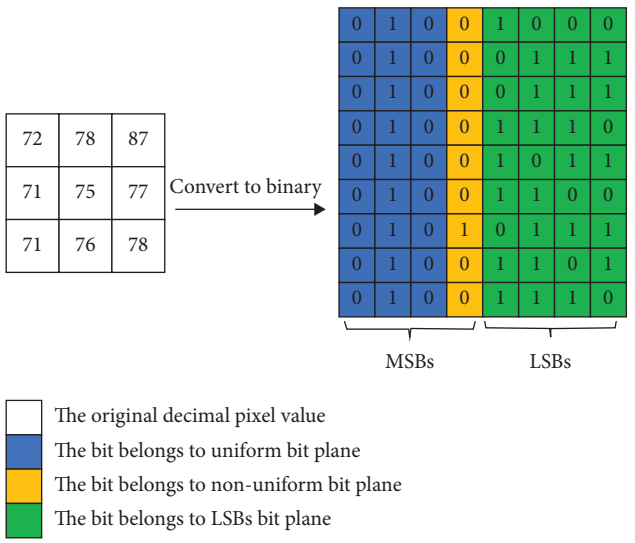


Figure 3: The divisions of the MSBs bit plane and the LSBs bit plane.

calculated in equation (6). Nearly 44% nonuniform bit plane of the encryption blocks is redundant. Therefore, the compressed method in [37] can be exploited to compress the sparse nonuniform bit plane. Here, we refer to the compression method for the uniform and the nonuniform bit planes as the multiple MSB encoding method.

Different from the previous correlation analysis method, we consider the current LSBs bit plane to be compressible when the bit length required to record the maximum difference is smaller than the original bit length. Therefore, we first calculate the difference between the corresponding decimal numbers of the LSBs bit plane in the encrypted block. As shown in Figure 6, the abscissa represents the bit length corresponding to the LSBs bit plane, and the ordinate represents the maximum difference length between the decimal numbers corresponding to the LSBs bit plane. We find the case where the length of the largest difference is less than the length of the LSBs bit plane in many blocks, that is, the LSBs bit plane of these blocks has strong correlation. Therefore, the LSBs bit plane can be compressed with the

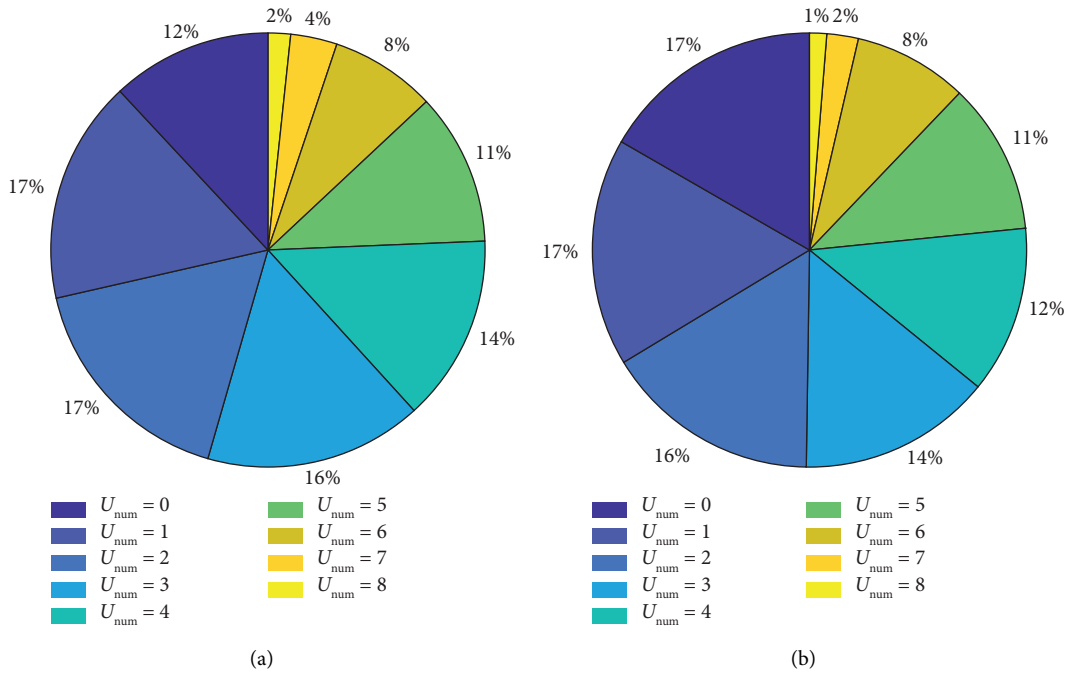


Figure 4: Average proportion of U_{num} for the images in the datasets. (a) BOSSBase [52]. (b) BOWS-2 [53].

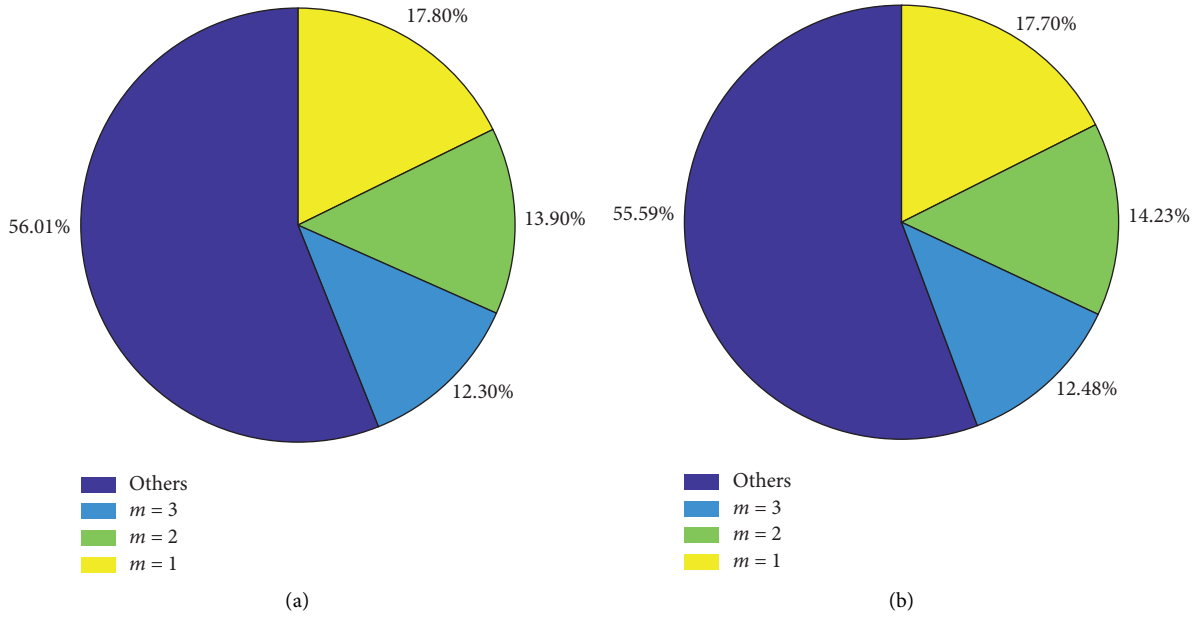


Figure 5: Average proportion of various nonuniform bit planes for the images in the datasets. (a) BOSSBase [52]. (b) BOWS-2 [53].

pixel difference encoding to vacate more space for additional data embedding.

It can be seen from the abovementioned correlation analysis that the three types of bit planes, i.e., the uniform bit plane, nonuniform bit plane, and LSBs bit plane, all have the redundancy. Therefore, different block-based encoding methods are designed to compress corresponding bit planes. The details of these methods will be described in the next two subsections.

2.2.2. Multiple MSB Encoding. First, the encrypted image is segmented into $s \times s$ sized nonoverlapping blocks, and the number of uniform bit plane $U_{num} \in [0, 8]$. Therefore, only three bits are needed to represent the quantity of the uniform bit plane, and to ensure the recoverability of uniform bit plane, U_{num} bits are used to represent the original content of the uniform bit plane, i.e., the content is 0 or 1. All uniform bit planes that satisfy the following conditions can be compressed into $U_{num} + 3$ bits.

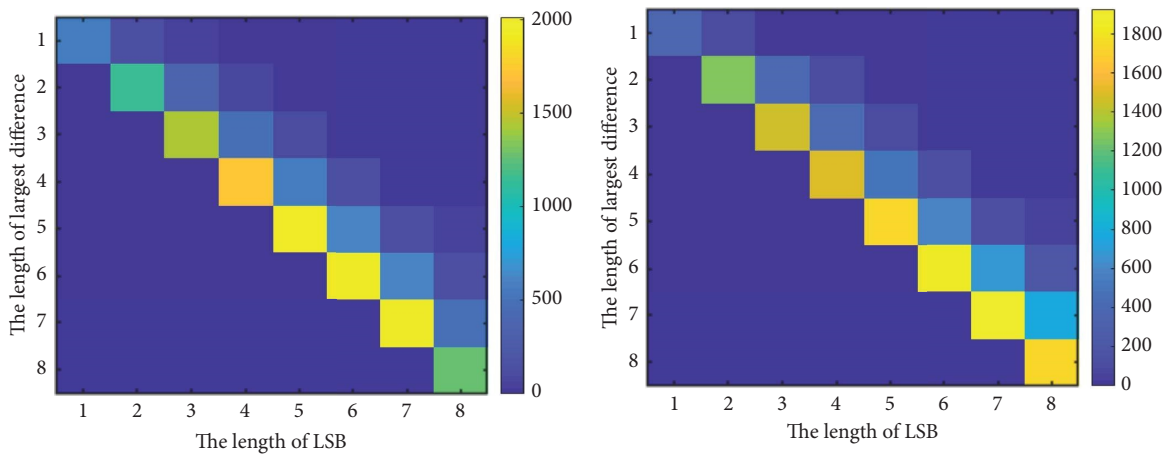


Figure 6: Average proportion of various LSBs bit plane for the images in the datasets. (a) BOSSBase [52]. (b) BOWS-2 [53].

$$U_{\text{num}} + 3 \le U_{\text{num}} \times s^2, \tag{5}$$

where the block size s will affect the performance of uniform bit plane compression. The optimization of s will be discussed in the experimental section.

Second, the sparsity of the nonuniform bit plane can be judged according to m. When $m \le n_T$, it means the nonuniform bit plane is sparse, and the sparse matrix compression method, such as [37], can be adopted to compress the nonuniform bit plane. When $m > n_T$, it means the nonuniform bit plane is not sparse and will be merged into the LSBs bit plane, i.e., the nonuniform bit plane becomes the part of the LSBs bit plane and will be compressed by the pixel difference encoding method. The threshold n_T is calculated according to the following equation.

$$n_T = \underset{x}{\arg\max}\left\{1 + 2^{\bullet}\log_2 x + x^{\bullet}\log_2\left(\frac{s^2}{2}\right)\right\} \le s^2, \tag{6}$$

where $\{1 + 2^{\bullet}\log_2 x + x^{\bullet}\log_2 (s^2/2)\}$ represents the length of compressed bits of the nonuniform bit plane. After the threshold n_T is determined, the nonuniform bit plane that satisfies the following condition can be compressed into $NU_{\text{num}} + 1$ bits.

$$NU_{\text{num}} + 1 \le s^2, \tag{7}$$

where NU_{num} is the quantity of the compressed nonuniform bit plane, and the one bit is the indicator bit, i.e., "1" represents sparse and "0" represents not sparse, which is used to record the sparsity of the nonuniform bit plane.

As shown in Figure 7, an example of multiple MSB encoding is presented. The pixels of an encrypted block with the size of 3×3 are first converted into 8-bit binary numbers. When $U_{\text{num}} = 3$ and $m = 1$, the uniform bit plane can be compressed into "011010" among which "011" is the indicator bit of the uniform bit plane, and the nonuniform bit plane can be compressed into "1111011" among which "1" is the indicator bit of the nonuniform bit plane. To facilitate data extraction on the receiver side, the indicator bit of the nonuniform bit plane is placed in the next bit of the original content of the uniform bit plane, i.e., the red block in Figure 7(b).

2.2.3. Pixel Difference Encoding. From the correlation analysis in Section 2.2, we can find that the LSBs bit plane also has redundancy. To fully utilize the redundancy of the LSB bit plane, a pixel difference encoding method is designed to compress the LSBs bit plane, and the detailed procedure is given as follows.

(1) For each block sized $s \times s$, the length of the LSBs bit plane is first calculated and denoted as L_a. And then the maximum decimal value among the $s \times s$ pixels in the current block is denoted as $P_{\text{m}} \in [0, 255]$. In order to ensure the recoverability of LSBs bit plane, the position of P_{m} in the encrypted block and the value of P_{m} should be recorded, and the bit length required to record them are calculated as follows:

$$L_a = \begin{cases} 8 - U_{\text{num}} - 1, if\ NU_{\text{num}} + 1 \le s^2, \\ 8 - U_{\text{num}}, if\ NU_{\text{num}} + 1 > s^2, \end{cases} \tag{8}$$

$$l_1 = L_a, \tag{9}$$

$$l_2 = \log_2 s^2, \tag{10}$$

where l_1 and l_2 represent the length of P_m and its position, respectively.

(2) Calculate the difference between P_m and each of the $s^2 - 1$ decimal values corresponding to the L_a LSBs of other pixels in the current block, and denote the difference as d_τ, $\tau = \{1, 2, 3, \ldots, s^2 - 1\}$. To facilitate data extraction, the largest length of d_τ should be recorded in b_3 with l_3 bits, i.e., the l_3 represents the bit length required to record b_3. However, if the length of d_τ is close to l_1, it means that the compression effect is not good; thus, l_3 is calculated as follows:

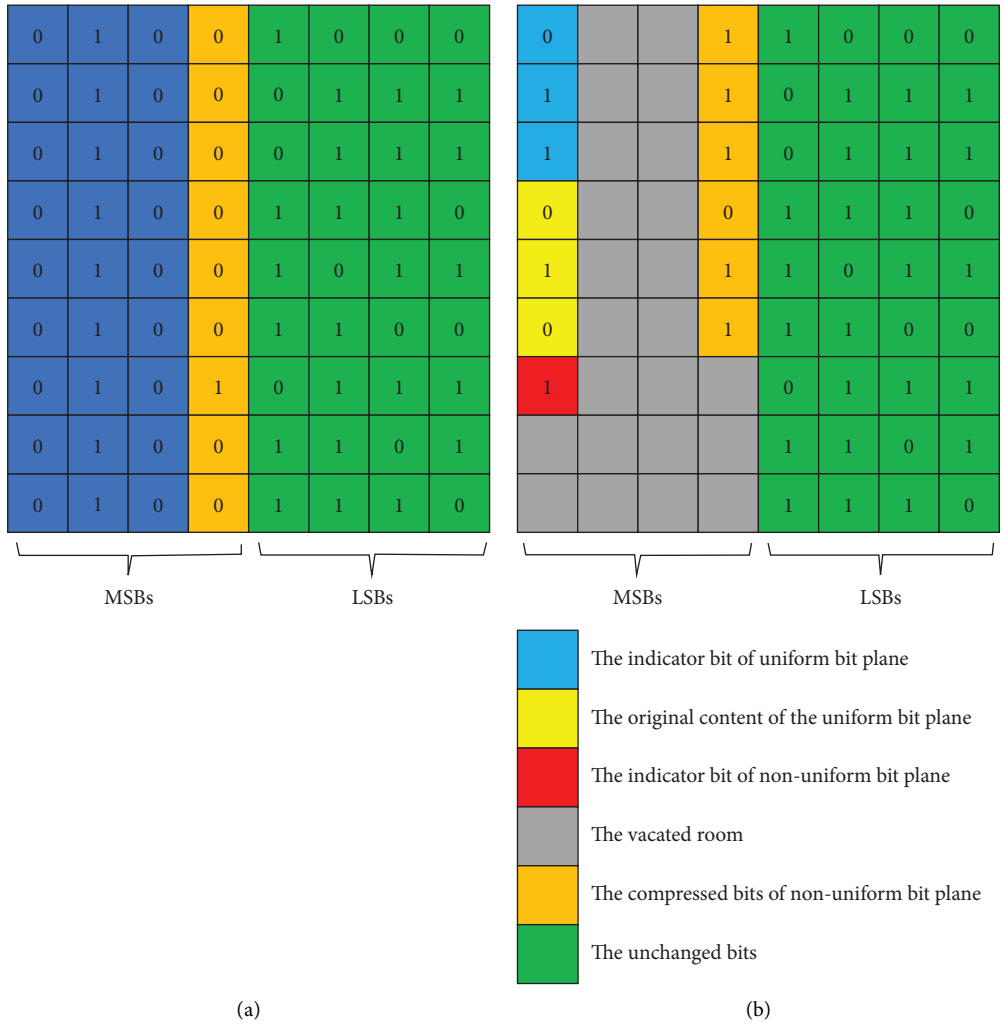


Figure 7: An example of the multiple MSB encoding method. (a) Encrypted block. (b) Compressed encrypted block with the multiple MSB encoding method.

$$l_3 = \log_2(l_1 - 1). \tag{11}$$

(3) When the length of d_τ is determined, the bit length of each difference d_τ can be calculated as follows:

$$l_4 = \sum_{k=1}^{l_3} 2^{b_3^k}. \tag{12}$$

(4) The LSBs bit plane is compressible only if the following condition is satisfied:

$$l_1 + l_2 + l_3 + l_4 \times \left(s^2 - 1\right) + 1 \le L_a \times s^2, \tag{13}$$

where L_a denotes the length of the LSBs bit plane, and the one bit indicates the compressibility of the LSBs bit plane, i.e., "1" represents compressible and "0" represents not compressible.

An example of th encrypted block being compressed by multiple MSB encoding and pixel difference encoding is given in Figure 8. It can be seen that each bit plane can be compressed to vacate space. To facilitate data extraction, the three kinds of bit planes are assigned with 3, 1, and 1 indicator bits to indicate whether the corresponding plane is compressed or not. To ensure reversible recovery, some auxiliary information needs to be recoded, which contains the indicator bits, the original content of the incompressible plane that are replaced by the indicator bits, and the compressed content of each bit plane.

2.3. Data Embedding. After the joint encoding of multiple MSB and pixel difference, the encrypted image is compressed and a lot of space is vacated to embed data. Before embedding, the additional data are XOR encrypted by a pseudorandom bit stream generated from the data-hiding key K_h. Finally, the encrypted additional data will be embedded into the vacated room by LSB replacement.

2.4. Data Extraction and Image Recovery. According to the available keys, different operations can be conducted on the marked encrypted image I_m. (1) The decrypted image with almost the same visual content as the original image can be

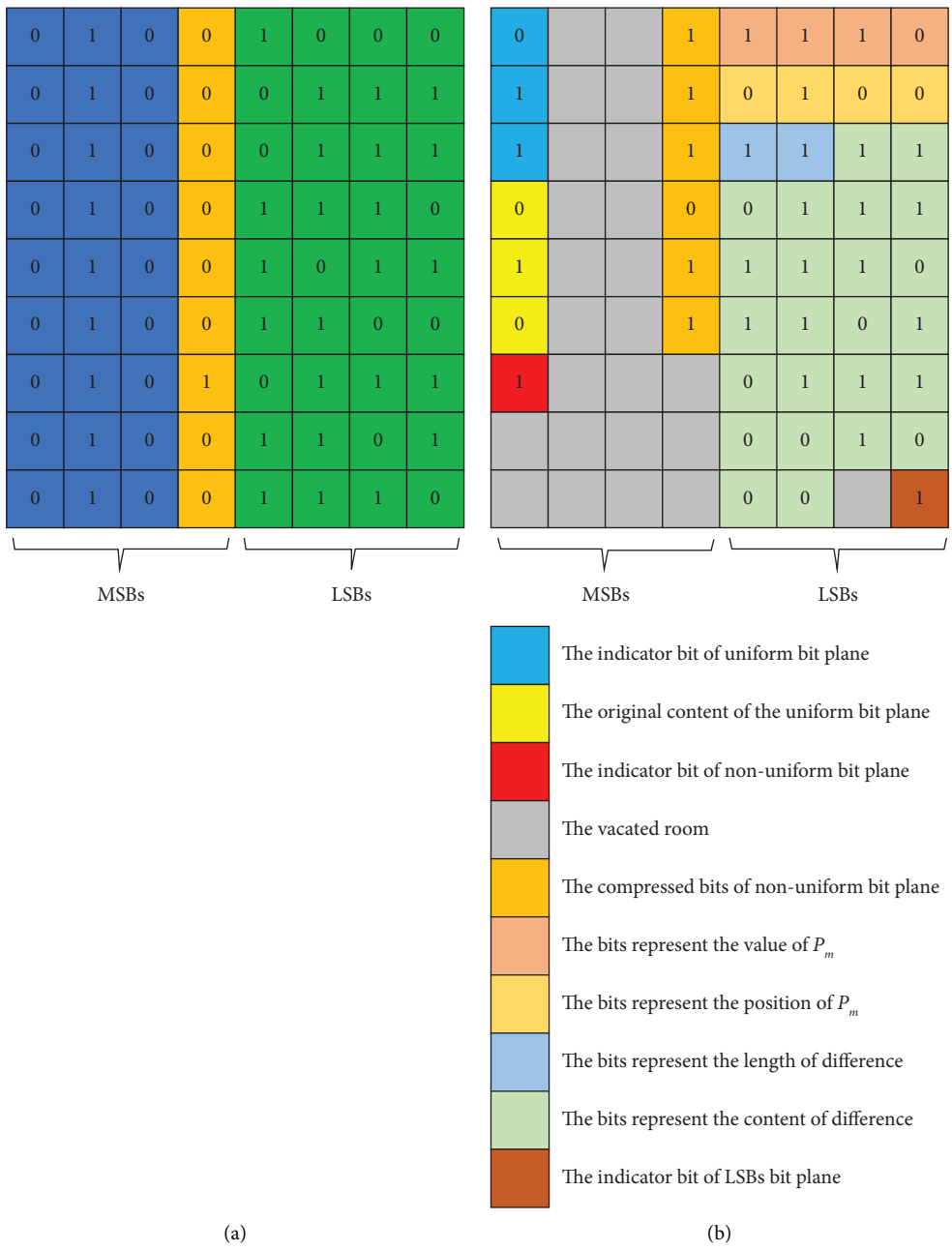


Figure 8: An example of multiple MSB encoding and pixel difference encoding. (a) Encrypted block. (b) Compressed encrypted block with two encoding methods.

achieved through the encryption key K_e. (2) Embedded additional data can be extracted with no errors by using the data-hiding key K_h. (3) If the data hiding key K_h and the encryption key K_e are both available, the original image can be recovered losslessly and additional data can be extracted without distortions simultaneously. Separability makes image decoding and data extraction more flexible, and it helps to achieve the hierarchical management of data, for example, receivers with different permissions can only obtain different information.

2.4.1. Data Extraction. In order to extract the embedded additional data, the receiver first divides the marked encrypted image I_m into nonoverlapping blocks sized $s \times s$. Second, according to the indicator bits in the uniform bit plane, the block types and the number of LSBs bit plane L_a can be obtained. For each block, divide the 8-bit planes into $8-L_a$ MSBs bit plane and L_a LSBs bit plane. Third, extract the additional data according to the inverse of the embedding process. Finally, the original additional data can be obtained by XOR decryption with the data-hiding key K_h.

2.4.2. Image Decryption. Since the encryption process and the data embedding process do not interfere with each other, image decryption and data extraction can be separably implemented with the data-hiding key K_h and the encryption key K_e. Divide the marked encrypted image I_m into blocks according to the method described in the previous subsection, and then use the encryption key K_e to perform the inverse process of the encryption process to obtain the decrypted image.

2.4.3. Image Recovery. If both keys K_h and K_e are available, the additional data and decrypted image can be obtained as shown in Sections 2.4.1 and 2.4.2. Then, the original bits can be recovered according to the additional data, and a reversible image can also be obtained.

3. Experimental Results and Analyses

In order to demonstrate the superior performance of the proposed scheme, a series of experiments were conducted on the six standard grayscale images sized 512×512 in Figure 9 and a large number of randomly selected images from BOSSBase [52] and BOWS-2 [53]. The experiment environment was based on a personal computer with a 16.00 GB memory, 4.20 GHz intel i7 processor, and Matlab R2018b. The experimental setup is as follows: (1) parameter settings and experimental results of the proposed scheme, (2) security analysis, and (3) comparisons with state-of-the-art schemes.

3.1. Results of Our Scheme. In the embedding process, the block size $s \times s$ greatly affects the embedding rate of our scheme; thus, a large number of experiments about the choosing of the optimal block size $s \times s$ were conducted. As shown in Table 1, we first calculated the average embedding rate for the six images in Figure 9 and an image dataset with large number of images, i.e., BOSSBase [52], under different block sizes to achieve the optimal value of s, i.e., $s = 4$ is the optimal value. Since the block size s affects the correlation of the encrypted block, too large or too small a block will result in a smaller embedding rate. Detailedly, if s is too large, the correlation between pixels in the block will become small; thus, the compression efficiency will become lower. On the contrary, if s is too small, the auxiliary information is required in each block while the number of blocks increases; thus, the embedding rate will reduce due to the large number of auxiliary information. For the image dataset [52], its average embedding rate is greater than that for the six standard test images in Figure 9 because the complexity of images in [52] is lower than that of the test images in Figure 9, and more redundancy space can be compressed for the images in [52].

Besides, the experimental results of the embedding rate performance for the three kinds of bit planes are given in Table 2, where τ represents the total embedding rate, and τ_1, τ_2, and τ_3 represent the embedding rate of the uniform bit plane, nonuniform bit plane, and LSBs bit plane, respectively. The auxiliary information lengths of the three bit planes are different. On average, the uniform bit plane has the largest correlations, the nonuniform bit plane is second, and the LSBs bit plane has the least correlations. In addition, the lengths of auxiliary information for the three kinds of bit planes are also different. So, the embedding rates τ_1, τ_2, and τ_3 achieve their maximums under different block sizes. When the blocks' side length is 4, the sum of these three embedding rates is the highest. Therefore, the subsequent experiments of our scheme are carried out on the basis of $s = 4$.

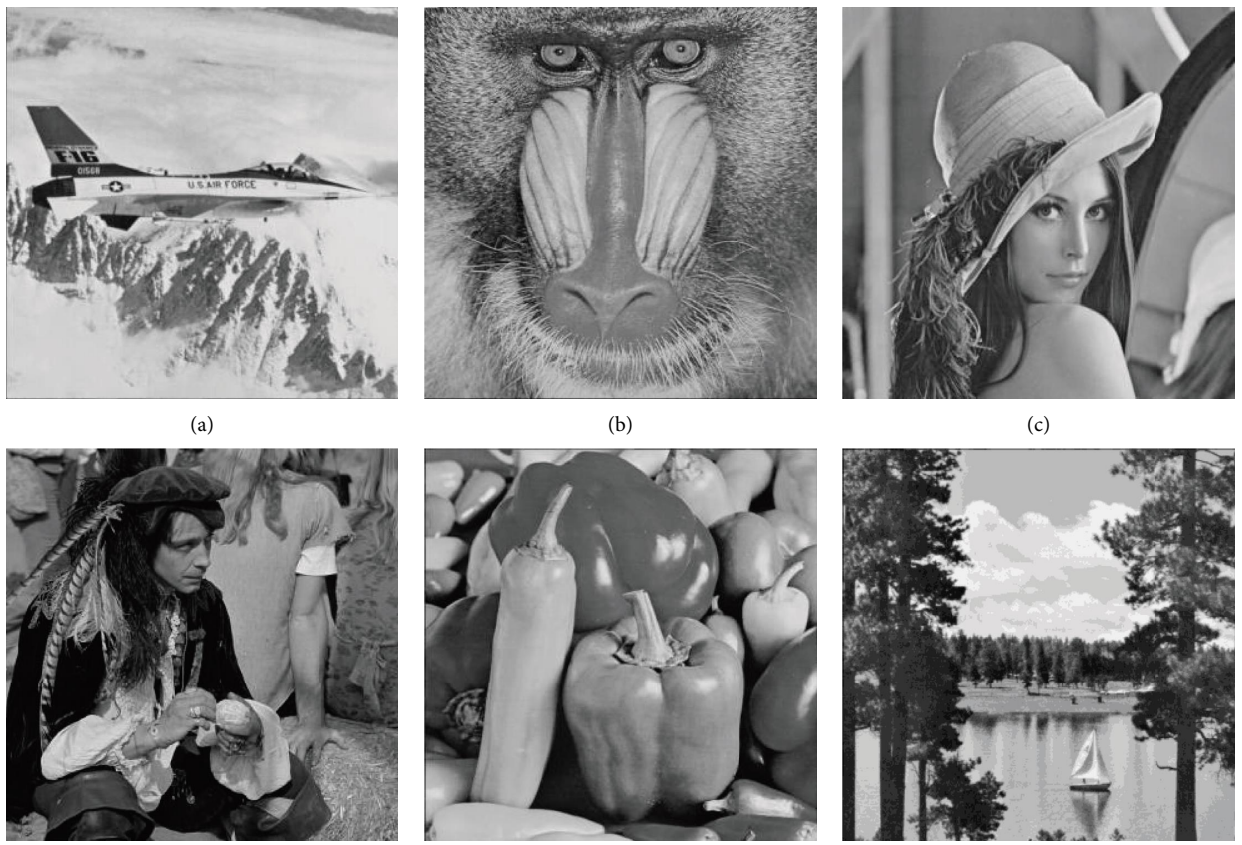
3.2. Security Analysis. Figure 10 gives an example of the image encryption ($s = 4$) for *Lena* sized 512×512. Figure 10(a) shows the original image of *Lena*. Figures 10(b)–10(d) show the encrypted results after the bit plane XOR for blocks, blocks scrambling, and pixel scrambling, respectively. Figures 10(e)–10(h) are the histograms of the original image, the intermediate encrypted images, and the final encrypted image, respectively. It can be observed that the encryption method in our scheme can protect the plaintext image contents and statistical information well.

In addition, the information entropy of the encrypted image was also evaluated. Entropy can be used to represent the degree of randomness for an image, which can be calculated as follows:

$$H(I) = -\sum_{i=1}^{255} \rho(H_i) \log_2 \rho(H_i), \tag{14}$$

where I denotes a grayscale image with 256 grey levels H_i, and ρ represents the probability of H_i. According to the information theory, the higher the randomness of the encrypted image is, the greater and the closer to 8 bits the entropy value is. The entropy values of the six images in Figure 9 before and after encryption are listed in Table 3. We can find that all the encrypted images have the entropy values that are much greater than that of the original images and closer to 8. Therefore, the image encryption method in our scheme can ensure the security of image contents effectively.

3.3. Comparisons with State-of-the-Art Schemes. In order to demonstrate the superiority of the proposed method, some state-of-art schemes, including Zhang's scheme [28, 31], and the schemes proposed by Liu and Pun [36], Qin et al. [37], Fu et al. [38], Yi and Zhou et al. [48], Chen and Chang et al. [39], and Ma et al. [42] were used for performance comparisons. For the fair comparison, the abovementioned schemes all adopted the optimal parameters to obtain their best performances. Specifically, the schemes in [36–39, 48] chose the block size as 4, three LSBs bit planes were adopted in [42, 48], four LSBs bit plane were applied in [37], and five MSB bit layers were used in [38]. Four test images shown in Figure 9 were adopted to compare the embedding rate and the image



(a) (b) (c)

(d) (e) (f)

Figure 9: Standard test images. (a) Airplane. (b) Baboon. (c) Lena. (d) Man. (e) Peppers. (f) Sailboat.

Table 1: Embedding rates under different block sizes $s \times s$ (bpp).

Images	Embedding rate τ					
	$s=3$	$s=4$	$s=5$	$s=6$	$s=7$	$s=8$
BOSSBase [47]	2.6470	2.7880	2.7593	2.6683	2.6387	2.5499
Airplane	2.4018	2.4666	2.4288	2.3116	2.2738	2.1014
Baboon	0.6690	0.7322	0.7406	0.6651	0.6438	0.5851
Lena	2.0047	2.0921	2.0279	1.9254	1.8831	1.7795
Man	1.7544	1.8145	1.7614	1.6561	1.6164	1.5262
Peppers	1.9167	2.0052	1.9776	1.8695	1.8278	1.7190
Sailboat	1.5942	1.6785	1.6484	1.5517	1.5047	1.4366

Table 2: Embedding rates of the two embedding methods under different block sizes $s \times s$ (bpp).

Images	s	Embedding rate τ	Embedding rate τ_1	Embedding rate τ_2	Embedding rate τ_3
Airplane	$s=3$	2.3474	2.2523	0.0659	0.0292
	$s=4$	2.4666	2.1774	0.1550	0.1342
	$s=5$	2.4288	2.0410	0.1846	0.2031
	$s=6$	2.3116	1.9143	0.1767	0.2206
Baboon	$s=3$	0.6130	0.5441	0.0570	0.0119
	$s=4$	0.7322	0.4700	0.1365	0.1257
	$s=5$	0.7406	0.3976	0.1570	0.1861
	$s=6$	0.6651	0.3378	0.1361	0.1913

Table 2: Continued.

Images	s	Embedding rate τ	Embedding rate τ_1	Embedding rate τ_2	Embedding rate τ_3
Lena	$s=3$	1.9523	1.8446	0.0640	0.0437
	$s=4$	2.0921	1.7756	0.1534	0.1630
	$s=5$	2.0279	1.6127	0.1869	0.2284
	$s=6$	1.9254	1.4987	0.1726	0.2541
Man	$s=3$	1.8265	1.7443	0.0746	0.0076
	$s=4$	1.9139	1.6661	0.1376	0.1102
	$s=5$	1.8915	1.5487	0.1845	0.1583
	$s=6$	1.7972	1.4368	0.1837	0.1766
Peppers	$s=3$	1.8676	1.7531	0.0733	0.0413
	$s=4$	2.0052	1.6943	0.1606	0.1504
	$s=5$	1.9776	1.5661	0.1954	0.2162
	$s=6$	1.8695	1.4485	0.1857	0.2353
Sailboat	$s=3$	1.5406	1.4581	0.0668	0.0158
	$s=4$	1.6785	1.4017	0.1571	0.1197
	$s=5$	1.6484	1.2937	0.1906	0.1642
	$s=6$	1.5517	1.1681	0.1896	0.1941

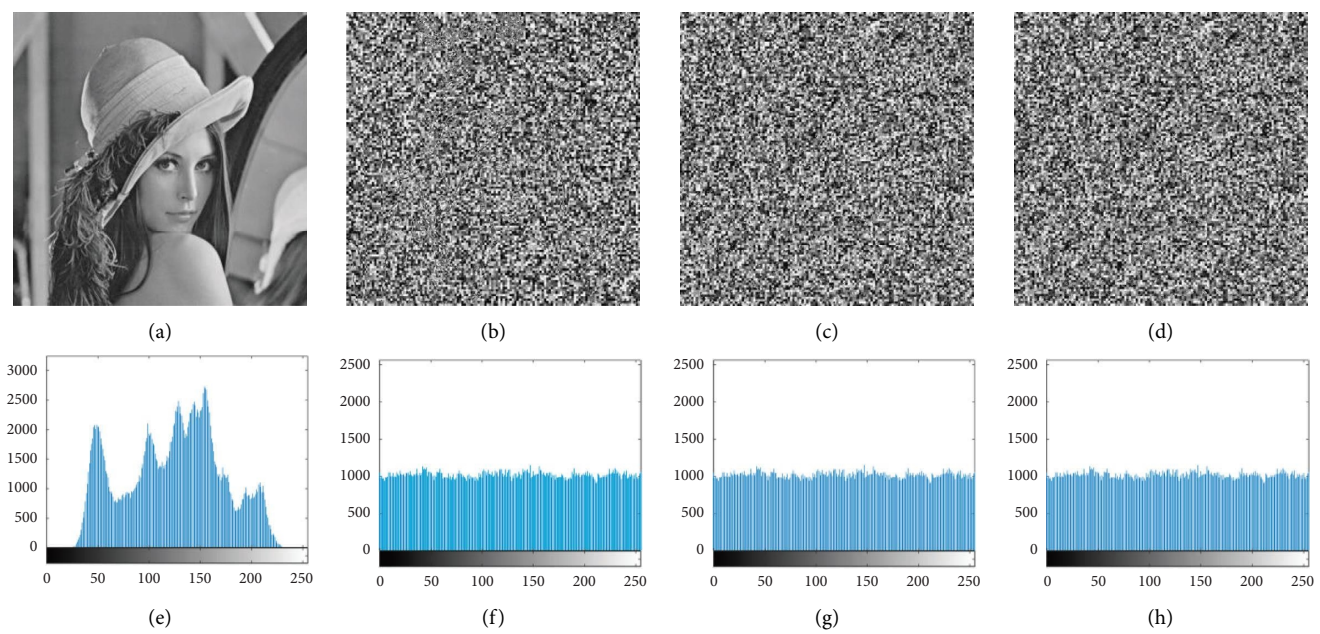


Figure 10: Results of image encryption for *Lena*. (a) Original image, (b) the result after the block-based bit plane XOR, (c) the result after the block-based bit plane XOR and blocks scrambling, (d) the final encrypted image, and (e–h) the histograms of images in (a–d).

Table 3: Information entropy for different images (bits).

Images	Original image	Encrypted image
Airplane	6.7059	7.9982
Baboon	7.3585	7.9991
Lena	7.4455	7.9982
Man	7.5237	7.9996
Peppers	7.5944	7.9986
Sailboat	7.4842	7.9991

quality of the proposed scheme and the schemes in [28, 31, 36–38, 42]. The results in Figure 11 show that our scheme can achieve a higher embedding rate than the schemes in [28, 31, 36–38, 42] and satisfactory decrypted-image quality can also be obtained.

Table 4 compares the highest embedding rates among the six typical RDHEI schemes [36–40, 48] and our scheme. It can be seen that the embedding rate of the proposed scheme is greater than that of other schemes in [36–39, 48] and only slightly below the scheme proposed in [40]. Because the scheme proposed in [40] is based on the adaptive strategy, it needs to be continuously optimized to select the optimal parameters; therefore, the computational complexity is higher. Meanwhile, we only compare the optimal performance of each scheme for 6 images in Table 4, and in other images, our scheme can achieve better performance, as shown in Figure 12. Considering the influence of image complexity, two image databases [52, 53] that contain 10000 images were tested to illustrate the general embedding rate. As shown in

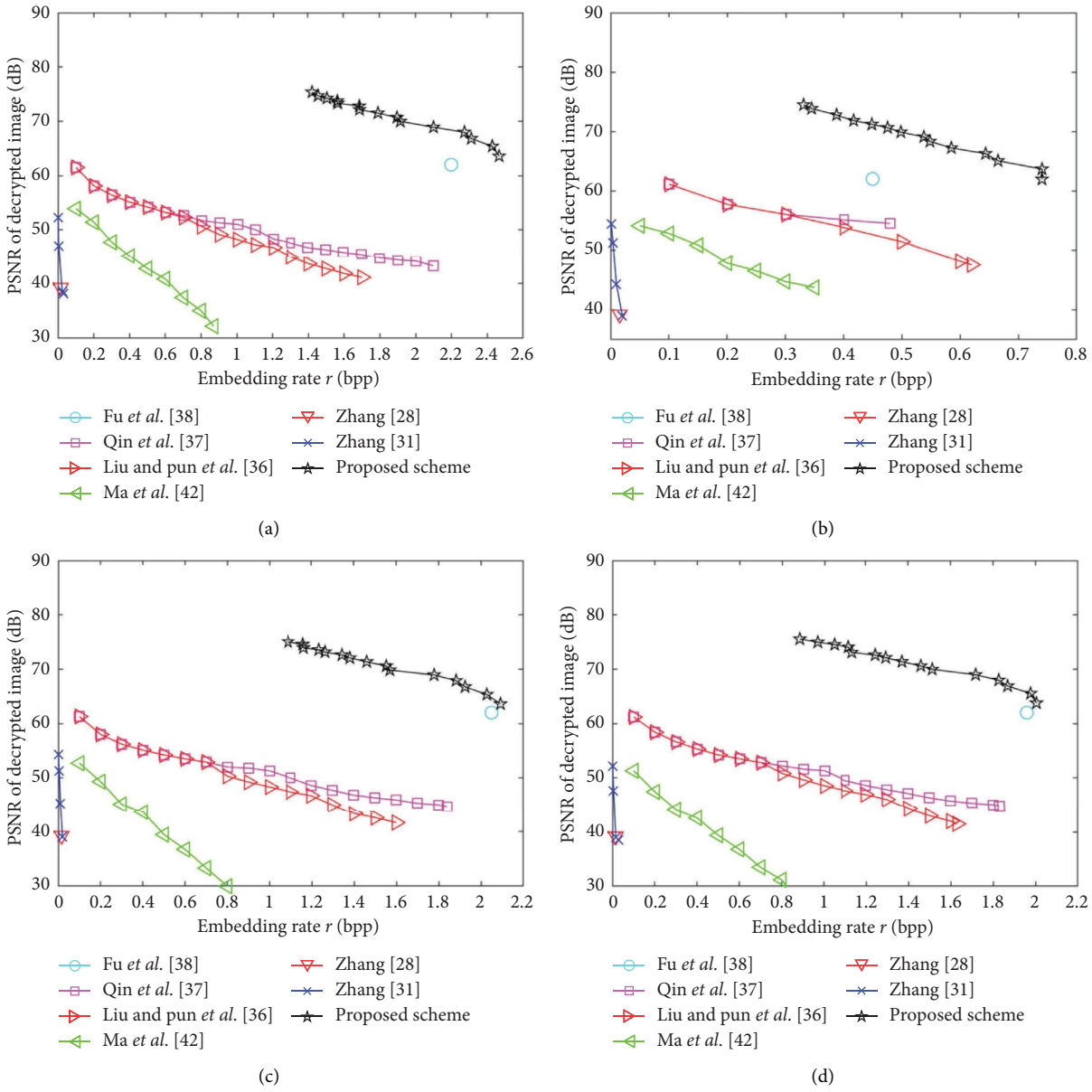


Figure 11: Comparisons of rate-distortion performance among our scheme andthe schemes in [28, 31, 36–38, 42]. (a) Airplane. (b) Baboon. (c) Lena. (d) Peppers.

Table 4: Comparisons of the embedding rate for the six standard images among our scheme and the schemes in [36–40, 48] (bpp).

		Airplane	Baboon	Lena	Man	Peppers	Sailboat
RRBE	Yi and Zhou [48]	2.1630	0.4025	1.7653	1.5747	1.7301	1.4809
	Chen and Chang [39]	2.3378	0.5352	1.7653	1.6787	1.8712	1.5857
VRAE	Liu and Pun [36]	1.7087	0.6215	1.6062	1.4509	1.6347	1.4293
	Qin et al. [37]	2.0896	0.4833	1.8387	1.602	1.8213	1.5532
	Fu et al. [38]	2.1652	0.4545	2.016	1.3111	1.9654	1.5718
	Wang et al. [40]	2.5031	0.7641	2.0853	1.8507	2.1002	1.8020
	Proposed scheme	2.4666	0.7406	2.0921	1.8145	2.0052	1.6785

Figure 12, the minimum, the maximum, and the average values of embedding rates on these two image databases were given. And it can be observed that our scheme achieves better performance because of the full use of the correlation between pixels. In addition to the compression of the MSBs bit plane with high correlation, the LSBs bit plane with low correlation are also compressed by the pixel difference block encoding.

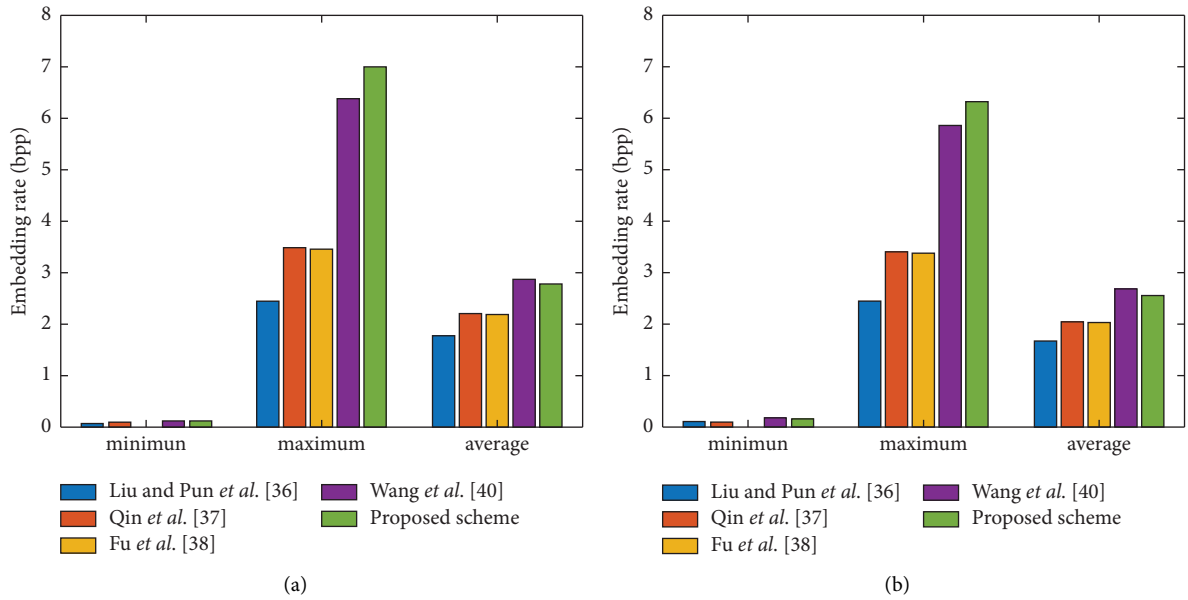


Figure 12: Comparisons of embedding rate on the two image datasets among our scheme and the schemes in [36–38, 40]. (a) BOSSBase [52]. (b) BOWS-2 [53].

4. Conclusion

In this work, we propose an RDHEI scheme based on the joint encoding of multiple MSB and pixel difference. The relevance of adjacent pixels is preserved by our block-based encryption method. In order to vacate more space to embed additional data, the bit planes are divided into different types of bit planes, i.e., the uniform bit plane, nonuniform bit plane, and LSBs bit plane, according to the redundancy degree. Generally, the MSBs bit plane consists of the uniform bit plane and the nonuniform bit plane. The uniform bit plane can be compressed based on its uniformity, and the nonuniform bit plane can be compressed by any binary compression method based on its sparsity. In addition, the LSBs bit plane can be represented by a maximum and several differences which occupy less space. Therefore, an efficient pixel difference encoding method is proposed to compress the LSBs bit plane. Experimental results demonstrate that our scheme can obtain the better embedding rate and visual quality of decrypted image than some reported schemes. Besides, the receiver can extract the additional data and decrypt the image separately.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Shanghai Pujiang Program (22PJD031), Academic Mentorship for Scientific Research Cadre Project (E3-0200-22-201007-6), the Foundation of National Key R&D Program of China (2020YFC2008700), the National Natural Science Foundation of China (82072228), the Foundation of Shanghai Municipal Commission of Economy and Informatization (202001007), Three-Year Action Plan for the Key Discipline Construction Project of Shanghai Public Health System Construction (GWV-10.1-XK05), the Natural Science Foundation of Shandong (ZR2020MF054), the Shandong Provincial Natural Science Foundation (ZR2019BF017), and the Major Scientific and Technological Innovation Projects of Shandong Province (2019JZZY010127, 2019JZZY010132, and 2019JZZY010201).

References

[1] L. Huang, Z. Y. Xiang, J. Li, H. Yao, and C. Qin, "New framework of self-embedding fragile watermarking based on reference sharing mechanism," *Security And Communication Networks*, vol. 2022, Article ID 2699802, 14 pages, 2022.

[2] X. R. Li, C. Qin, Z. C. Wang, Z. X. Qian, and X. P. Zhang, "Unified performance evaluation method for perceptual image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1404–1419, 2022.

[3] J. Li, X. L. Li, B. Yang, and X. M. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, 2015.

[4] H. Wu, B. Yi, F. Ding, G. Feng, and X. Zhang, "Linguistic steganalysis with graph neural networks," *IEEE Signal Processing Letters*, vol. 28, pp. 558–562, 2021.

[5] F. Ding, G. Zhu, Y. Li, X. Zhang, P. K. Atrey, and S. Lyu, "Anti-forensics for face swapping videos via adversarial training," *IEEE Transactions on Multimedia*, vol. 24, pp. 3429–3441, 2022.

[6] F. Ding, H. Wu, G. Zhu, and Y.-Q. Shi, "METEOR: measurable energy map toward the estimation of resampling rate via a convolutional neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 12, pp. 4715–4727, 2020.

[7] F. Ding, B. Fan, Z. Shen et al., "Securing facial bioinformation by eliminating adversarial perturbations," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2022.

[8] J. M. Barton, "Method and apparatus for embedding authentication information within digital data," *United States Patent*, vol. 5, pp. 646–997, 1997.

[9] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding—new paradigm in digital watermarking," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 2, pp. 185–196, 2002.

[10] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[11] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[12] D. Coltuc and J. M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Processing Letters*, vol. 14, no. 4, pp. 255–258, 2007.

[13] S. Weng, Y. Zhao, J.-S. Pan, and R. Ni, "Reversible watermarking based on invariability and adjustment on pixel pairs," *IEEE Signal Processing Letters*, vol. 15, pp. 721–724, 2008.

[14] X. Wang, X. Li, B. Yang, and Z. Guo, "Efficient generalized integer transform for reversible watermarking," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 567–570, 2010.

[15] Y. Qiu, Z. Qian, and L. Yu, "Adaptive reversible data hiding by extending the generalized integer transformation," *IEEE Signal Processing Letters*, vol. 23, no. 1, pp. 130–134, 2016.

[16] F. Peng, X. Li, and B. Yang, "Adaptive reversible data hiding scheme based on integer transform," *Signal Processing*, vol. 92, no. 1, pp. 54–62, 2012.

[17] Y. Hu, H.-K. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 250–260, 2009.

[18] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989–999, 2009.

[19] W.-L. Tai, C.-M. Yeh, and C.-C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.

[20] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, "Reversible image watermarking using interpolation technique," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 187–193, 2010.

[21] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 873–882, 2011.

[22] X. Gao, L. An, Y. Yuan, D. Tao, and X. Li, "Lossless data embedding using generalized statistical quantity histogram," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1061–1070, 2011.

[23] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524–3533, 2011.

[24] J. Zhou and O. C. Au, "Determining the capacity parameters in PEE-based reversible image watermarking," *IEEE Signal Processing Letters*, vol. 19, no. 5, pp. 287–290, 2012.

[25] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, "Reversible watermarking based on invariant image classification and dynamic histogram shifting," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 111–120, 2013.

[26] C. Qin, C.-C. Chang, Y.-H. Huang, and L.-T. Liao, "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1109–1118, 2013.

[27] B. Ma and Y. Q. Shi, "A reversible data hiding scheme based on code division multiplexing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1914–1927, 2016.

[28] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

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

[30] C. Qin and X. Zhang, "Effective reversible data hiding in encrypted image with privacy protection for image content," *Journal of Visual Communication and Image Representation*, vol. 31, pp. 154–164, 2015.

[31] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.

[32] C. Qin, W. Zhang, F. Cao, X. P. Zhang, and C. C. Chang, "Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection," *Signal Processing*, vol. 153, pp. 109–122, 2018.

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

[34] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 441–452, 2016.

[35] F. Huang, J. Huang, and Y. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777–2789, 2016.

[36] Z. L. Liu and C. M. Pun, "Reversible data-hiding in encrypted images by redundant space transfer," *Information Sciences*, vol. 433-434, pp. 188–203, 2018.

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

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

[39] K. Chen and C. C. Chang, "High-capacity reversible data hiding in encrypted images based on extended run-length coding and block-based MSB plane rearrangement," *Journal of Visual Communication and Image Representation*, vol. 46, no. 5, pp. 1132–1143, 2015.

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

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

[42] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before

encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

[43] W. Zhang, K. Ma, and N. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, pp. 118–127, 2014.

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

[45] D. Xu and R. Wang, "Separable and error-free reversible data hiding in encrypted images," *Signal Processing*, vol. 123, pp. 9–21, 2016.

[46] Y. Chen, B. Yin, H. He, S. Yan, and F. Chen, "Reversible data hiding in classification- scrambling encrypted-image based on iterative recovery," *Computers, Materials & Continua*, vol. 56, no. 2, pp. 299–312, 2018.

[47] P. Puteaux and W. Puech, "An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1670–1681, 2018.

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

[49] F. Cao, Y. J. Fu, H. Yao, M. Zou, J. Li, and C. Qin, "Separable reversible data hiding in encrypted vq-encoded images," *Security and Communication Networks*, vol. 202216 pages, 2022, https://doi.org/10.1155/2022/1227926, Article ID 1227926.

[50] P. Kong, D. Fu, X. R. Li, and C. Qin, "Reversible data hiding in encrypted medical DICOM image," *Multimedia Systems*, vol. 27, no. 3, pp. 303–315, 2021.

[51] H. Wu, Y.-Q. Shi, H. Wang, and L. Zhou, "Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 8, pp. 1620–1631, 2017.

[52] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system: the ins and outs of organizing BOSS," in *Proceedings of the International Workshop on Information Hiding*, pp. 59–70, May 2011, http://dde.binghamton.edu/download/.

[53] P. Bas and T. Furon, *Image Database of BOWS-2*, 2017, http://bows2.ec-lille.fr/.