mathematics

MDPI

Article

High Capacity Reversible Data Hiding Based on the Compression of Pixel Differences

Kai-Meng Chen

Computer Engineering College, Jimei University, Xiamen 361021, China;
chenkaimeng@jmu.edu.cn

Received: 31 July 2020; Accepted: 24 August 2020; Published: 26 August 2020

Abstract: In this paper, we proposed a novel reversible data hiding method in encrypted image (RDHEI), which is based on the compression of pixel differences. In the proposed method, at the content owner' side the image is divided into non-overlapping blocks, and a block-level image encryption scheme is used to generate the encrypted image, which partially retains spatial correlation in the blocks. Due to the spatial correlation, in each block the pixels are highly likely to be similar. Therefore, the pixel differences in all blocks are concentrated in a small range and can be compressed. By the compression of pixel differences, the data hider can vacate the room to accommodate secret data in the encrypted image without losing information. At the receiver's side, the receiver can obtain secret data or retrieve the original image using different keys with no error. The experimental results demonstrate that, compared with existing methods, the proposed method can achieve a higher capacity and visual quality.

Keywords: reversible data hiding; image encryption; compression

1. Introduction

Reversible data hiding (RDH) is a technology which embeds secret data into cover media (such as text, image, and video) imperceptibly and reversibly. The embedded data can be extracted and the cover media can be completely recovered. Because the digital images are used widely in various fields, a series of RDH methods based on images have been proposed [1]. These methods can be classified into two categories: RDH in plaintext images [2–9] and RDH in encrypted images (RDHEI) [10–33].

RDHEI methods embed secret data into an encrypted image in a reversible way without decryption or being aware of the image content. RDHEI technology is useful for applications in which the data managers have no access to the contents of the images for privacy or other reasons. For example, in the application of cloud storage, to protect privacy, the original images are encrypted before storing them in the cloud servers. In RDHEI, there are three parties: the content owner, the data hider, and the receiver. The content owner encrypts the original image and transfers it to the data hider. The data hider embeds secret data into the encrypted image. After data embedding, the receiver can extract the secret data or retrieve the original image from the encrypted image which contains secret data. According to how the data hiding room is vacated, the existing RDHEI methods can be classified into the vacating room after encryption (VRAE) [10–27] and the reserving room before encryption (RRBE) [28–33].

In the VRAE methods, the original image is encrypted with no preprocessing by the content owner. The data hider should vacate room in the encrypted image for data hiding. In Zhang's method [10], the encrypted image is divided into non-overlapping blocks, and one secret bit is embedded into one block by flipping the three least-significant bits (LSBs) of half the pixels in the block. To extract the secret data and recover the flipped block, a smoothness estimator is used to judge the flipped pixels in each block after image decryption. The methods in [11–13] improve Zhang's method in terms of the reversibility and visual quality. In Hong et al.'s method [11] and Liao et al.'s method [12], to reduce errors in data extraction and image recovery, different improved smoothness estimators and side matching strategies are used. In Qin et al.'s method [13], an elaborate pixel selection scheme is used to reduce the distortion of the directly decrypted image, and a more precise smoothness estimator is used to reduce errors in data extraction and image recovery. In Wu et al.'s method [14], pixels are pseudo-randomly selected from the encrypted image and divided into same-size groups, and one secret bit is embedded into one group by flipping the i-th ($1 \leq i \leq 6$) LSBs of all pixels in the group. After image encryption, multiple designed content-aware pixel value predictors are used for detecting the bit flipping, so that the embedded bits can be extracted and the image can be recovered. Dragoi et al.'s method [15] extends Wu et al.'s method. In Dragoi et al.'s method, more pixels in the encrypted image can be selected for data hiding so that the embedding rate is improved, and Bose–Chaudhuri–Hocquenghem (BCH) coding is used to reduce errors in data extraction and image recovery. In Zhou et al.'s method [16], the encrypted image is divided into non-overlapping blocks, and a public key modulation scheme is used to embed a certain number of bits into each block. To recover the image and extract the embedded bits, a classifier based on the support vector machine (SVM) is used to judge the embedded bits. In [17], a designed sparse matrix is used to compress the LSB planes of the encrypted image so that the spared bits in the LSB planes can be used to accommodate the secret data. The receiver extracts the embedded data directly from the LSB planes, and uses a specific pixel predictor to recover the compressed LSBs. The methods in [18,19] improve the method in [17]. In Qian et al.'s method [18], the pixels of the encrypted image are divided into three subsets. In the different subsets, the LSBs of the pixels are compressed by different sparse matrices, and are recovered by different pixel predictors. In Qin et al.'s method [19], a specific image encryption scheme is used, so that the encrypted image can be decomposed into smooth regions and complex regions. Only the LSBs of the smooth regions are compressed for data hiding. In [20,21], Low-Density Parity Check (LDPC) code is used to compress the specific bit planes for data hiding. In Zhang et al.'s method [20], the bits of the fourth LSB plane of the encrypted image are compressed. In Qian et al.'s method [21], three quarters of the most significant bits (MSBs) are compressed. To recover the image, the log-likelihood ratio (LLR) and iterative belief propagation algorithm (BPA) decoding algorithm are used to recover the compressed bits.

Some VRAE methods use special encryption schemes to partially retain the spatial correlation in encrypted images, and the data hider uses variants of the different traditional RDH methods (such as histogram shifting, prediction error expansion, and pixel value ordering) to embed secret data. In Huang et al.'s method [22], various histogram shifting-based RDH methods are accomplished in the encrypted domain. In Xiao et al.'s method [23], a variant pixel value ordering scheme is used to embed secret bits into non-overlapping blocks of the encrypted image. In Yi et al.'s method [24], a designed pixel value predictor is used to generate prediction errors, and a variant prediction error expansion scheme is used for data embedding. In Qin et al.'s method [25], the MSBs of each block of the encrypted image are compressed by a sparse matrix compression scheme. In Li et al.'s method [26], based on a different histogram which is generated from the encrypted image, a variant histogram shifting scheme is used to embed secret data. In Ge et al.'s method [27], a multi-level histogram shifting scheme is used to embed secret data into each block of the encrypted image.

In the RRBE methods, before image encryption, the content owner preprocesses the original image to vacate room for accommodating secret data. The vacated room is retained after image encryption, and the data hider can embed secret data into the room directly. In Ma et al.'s method [28], the original image is divided into the smooth regions and the complex regions. To reserve the LSBs of the encrypted image for data hiding, the LSBs of the complex regions are embedded into the smooth regions using a traditional RDH method, such as difference expansion. Image encryption is performed after embedding the LSBs. At the data hider's side, the secret data can be placed into the original positions of the embedded LSBs directly. In Zhang et al.'s method [29], before image encryption a pixel estimator is used to predict the original values of a portion of pixels, then these pixels are substituted with their prediction errors. After image encryption, based on the histogram of the prediction errors, the data hider uses a variant histogram shifting scheme to embed secret data. In Cao et al.'s method [30], the original image is divided into patches, and each patch is encoded using less bits by a sparse coding technology with an over-complete dictionary. The spared bits in each patch can be used for data embedding after image encryption. In Yi et al.'s method [31], each MSB plane of the original image is divided into non-overlapping blocks and compressed by a designed sparse matrix coding scheme. Then, the LSBs of the image are embedded into the compressed MSB planes to vacate room for data hiding. In Chen et al.'s method [32], a bit plane rearrangement strategy is used to rearrange the MSBs of the original images, then the rearranged MSBs are compressed by an extended run-length coding scheme to vacate room for data hiding. In Qiu et al.'s method [33], the LSBs of the original image are removed for data hiding by using a reversible integer transformation scheme.

Benefitting from the use of the original image's spatial correlation, the RRBE methods can achieve a much higher capacity than the VRAE methods. However, the RRBE methods require the content owner to handle extra image processing work. In some cases, the content owner may not be able to process images. Therefore, the VRAE methods are more feasible for different applications.

In this paper, we propose a novel VRAE RDHEI method which is based on the compression of pixel differences. In the proposed method, at the content owner's side a specific image encryption scheme is used to partially retain the spatial correlation in non-overlapping 2×2 blocks of the encrypted image. At the data hider's side, for each block the four pixels are divided into two parts—one mark pixel and three replaceable pixels—and three pixel differences between the three replaceable pixels and the mark pixel are collected. Due to the spatial correlation, the pixel differences are highly likely to be concentrated and can be compressed efficiently. By replacing the information of the replaceable pixels with their compressed pixel differences, the data hiding room is vacated in the encrypted image without losing any information. At the receiver's side, the receiver can extract the secret data with no error and recover the image to its original version.

The rest of the paper is organized as follows. In Section 2, we present the detailed introduction to the proposed method. In Section 3, the experimental results and comparisons are provided. The conclusions are given in Section 4.

2. The Proposed Method

In this section, we introduce the details of the proposed method. Figure 1 illustrates the framework of the proposed method. At the content owner's side, the content owner encrypts the original image with no preprocessing using two encryption keys, and sends the encrypted image to the data hider. The data hider vacates room in the encrypted image, then uses a data hiding key to embed secret data into the room. The receiver extracts the embedded data from the encrypted image by the data hiding key K_{en}, and recovers the original image or generates the marked decrypted image containing the secret data by the encryption keys.

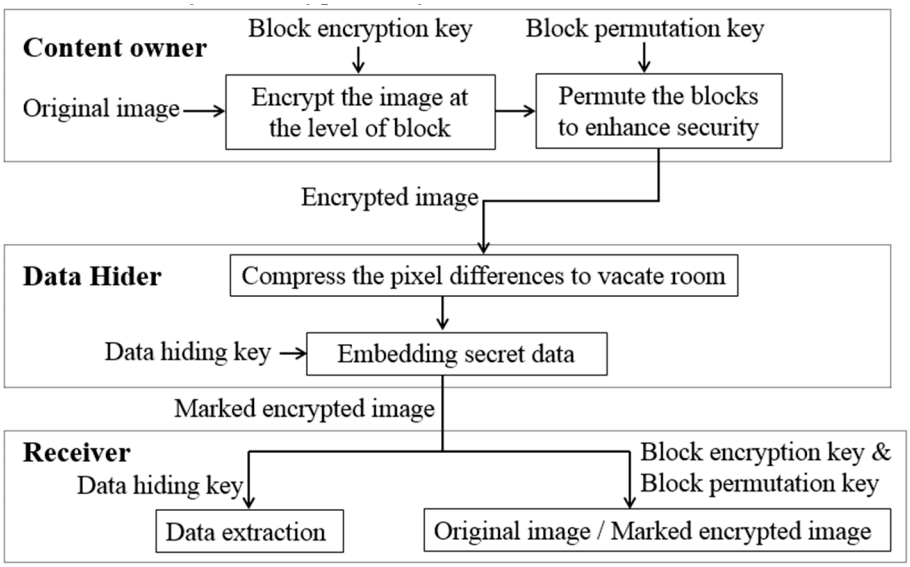

Figure 1. The framework of the proposed method.

2.1. Image Encryption

For an 8-bit standard gray image sized $H \times W$, we denote the k-th bit of the pixel $p_{i,j}$ as $b_{i,j,k}$ ($1 \le i \le H$, $1 \le j \le W$). The $b_{i,j,k}$ is defined as follows:

$$b_{i,j,k} = \left\lfloor p_{i,j}/2^{k-1} \right\rfloor \bmod 2, \text{ where } 1 \le k \le 8. \tag{1}$$

To partially retain the spatial correlation in the encrypted image for data hiding, a specific image encryption scheme is used to encrypt the original image at the level of the block. The detailed steps of the image encryption scheme are as follows:

Step 1: Divide the original image into $N = \lfloor H/2 \rfloor \times \lfloor W/2 \rfloor$ non-overlapping 2×2 blocks $\boldsymbol{B}_r$ ($r = 1, 2, \ldots N$).

Step 2: For each block $\boldsymbol{B}_r$, an 8-bit pseudo-random bit sequence S_r is generated using the block encryption key K_{en}. Denoting the four pixels in $\boldsymbol{B}_r$ as $p_r^{(1)}$, $p_r^{(2)}$, $p_r^{(3)}$, and $p_r^{(4)}$, each pixel $p_r^{(i)}$ is encrypted into $pe_r^{(i)}$ by S_r as follows:

$$e_{r,k}^{(i)} = b_{r,k}^{(i)} \oplus bs_{r,k},\ i = 1,2,3,4,\ k = 1,2,\ldots,8, \tag{2}$$

$$pe_r^{(i)} = \sum_{k=1}^{8} e_{r,k}^{(i)} \times 2^{k-1},\ i = 1,2,\ldots,8, k = 1,2,3,4, \tag{3}$$

where $b_{r,k}^{(i)}$ is the k-th bit of $p_r^{(i)}$, and $bs_{r,k}$ is the k-th bit of S_r.

If H or W is odd, after all the blocks are encrypted, the last row or column should be encrypted at the level of the pixel. For each pixel which does not belong to any block, an 8-bit pseudo-random bit sequence is generated by K_{en} to encrypt the pixel. The encryption process of each pixel is the same as that shown in Equations (2) and (3).

Step 3: After all the blocks have been encrypted, to enhance the encryption strength the content owner uses the block permutation key K_{bp} to pseudo-randomly permute all blocks $\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_N$ into $\boldsymbol{B}'_1, \boldsymbol{B}'_2, \ldots, \boldsymbol{B}'_N$ inside the encrypted image.

After image encryption, the encrypted image is sent to the data hider for embedding secret data. Because the four pixels in the block are encrypted by the same bit sequence, the encrypted pixels are still highly likely to be similar. Therefore, the spatial correlation of the original image can be partially

retained in each block. That allows the data hider to vacate room in each block for accommodating secret data.

2.2. Data Embedding

When the data hider receives the encrypted image, first the data hider divides the encrypted image to retrieve the 2×2 blocks $\boldsymbol{B}'_1, \boldsymbol{B}'_2, \ldots, \boldsymbol{B}'_N$. For each block $\boldsymbol{B}'_r$, the four pixels $pe_r^{(1)}$, $pe_r^{(2)}$, $pe_r^{(3)}$, and $pe_r^{(4)}$ are divided into one mark pixel and three replaceable pixels. As shown in Figure 2, to simplify the statement $pe_r^{(4)}$ is assigned to the mark pixel, and the other three pixels are assigned to the replaceable pixels. Then, three pixel differences $d_r^{(1)}$, $d_r^{(2)}$, and $d_r^{(3)}$ of block $\boldsymbol{B}'_r$ are calculated as:

$$d_r^{(i)} = pe_r^{(i)} - pe_r^{(4)}, \text{ where } i = 1, 2, 3. \tag{4}$$

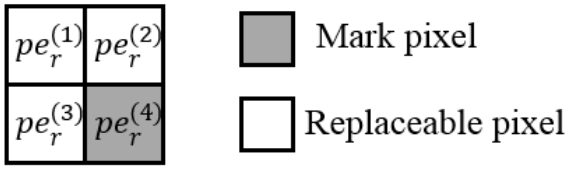

Figure 2. The mark pixel and replaceable pixels.

The range of $d_r^{(i)}$ is $[-255, 255]$. However, since the spatial correlation is retained in each block, there is a high probability that the four pixels inside the block are close to each other. Thus, $d_r^{(i)}$ is highly likely to be close to 0.

After all the pixel differences $\left\{d_1^{(1)}, d_1^{(2)}, d_1^{(3)}, d_2^{(1)}, d_2^{(2)}, d_2^{(3)}, \ldots, d_N^{(1)}, d_N^{(2)}, d_N^{(3)}\right\}$ are collected from all the blocks, these pixel differences are highly concentrated. Figure 3 shows the histogram of the pixel differences of Lena (in Figure 5). As shown in the figure, most of the pixel differences are concentrated in a small range around 0. Therefore, these pixel differences can be compressed efficiently by Huffman coding. According to the mark pixel and the pixel differences, the three replaceable pixels in each block can be recovered completely. Therefore, the data hider can replace the replaceable pixels with the compressed pixel differences to vacate room in the encrypted image for accommodating secret data.

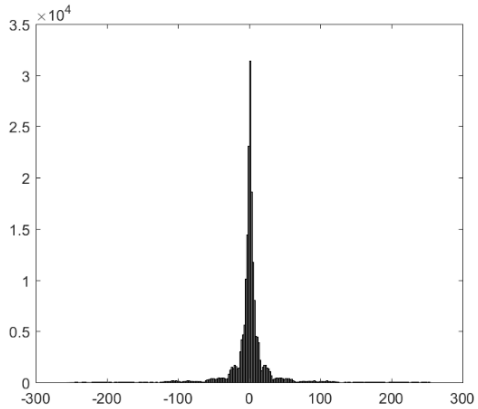

Figure 3. The histogram of the pixel differences of Lena.

The procedure of room vacating and data embedding is as follows:

Step 1: For each block, calculate the three pixel differences according to Equation (4).

Step 2: For all the pixel differences $d_r^{(i)}$ $(r = 1, 2, \ldots, N, i = 1, 2, 3)$, calculate the distribution of the difference values and use Huffman coding to encode all $d_r^{(i)}$ into $ed_r^{(i)}$ $(r = 1, 2, \ldots, N, i = 1, 2, 3)$.

Step 3: Compose all the encoded pixel differences $ed_1^{(1)}, ed_1^{(2)}, ed_1^{(3)}, \ldots, ed_N^{(1)}, ed_N^{(2)}, ed_N^{(3)}$ into one bit sequence BS_d, which is the bitstream of the compressed pixel differences.

Step 4: Beginning from the highest MSB plane, embed the Huffman codebook, BS_d, and their length information into the MSBs of all replaceable pixels by bit substitution. After this auxiliary information has been embedded, the rest bits of all the replaceable pixels can be used as the data hiding room.

Step 5**:** To embed secret data, beginning from the LSB plane the data hider uses the data hiding key K_h to pseudo-randomly select bits from the data hiding room and substitute them with the secret data.

Figure 4 shows an example of room vacating and data embedding. First, the encrypted image is divided into four 2×2 blocks: $\boldsymbol{B}'_1, \boldsymbol{B}'_2, \boldsymbol{B}'_3, \boldsymbol{B}'_4$. Then, three pixel differences in each block are calculated, and all the pixel differences are encoded by Huffman coding. All the encoded pixel differences compose the bitstream of the compressed pixel differences BS_d. Finally, the length information, the Huffman codebook, and the BS_d are embedded into the MSBs of the replaceable pixels, then the secret data are embedded into the rest bits of the replaceable pixels by the data hiding key.

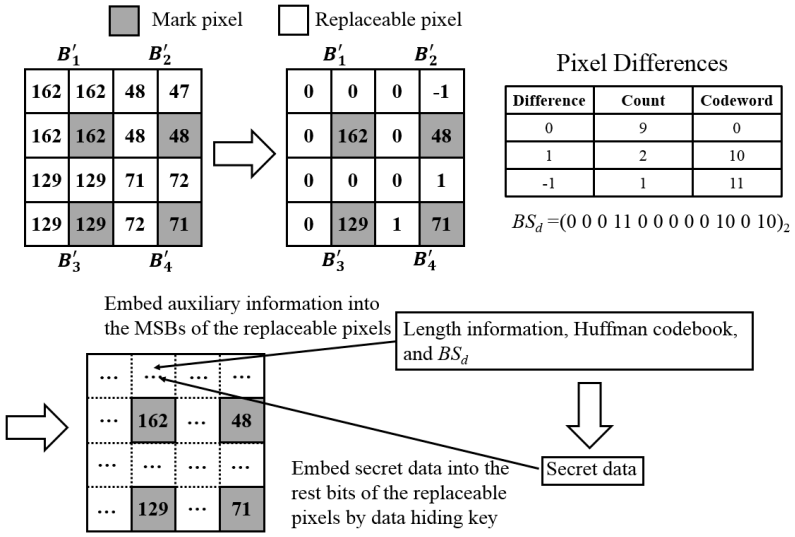

Figure 4. Example of room vacating and data embedding.

2.3. Data Extraction and Image Recovery

When the receiver gets the marked encrypted image containing secret data from the data hider, the receiver can obtain different data from the image using different keys.

Data extraction: When the receiver has the data hiding key K_h, the receiver can extract the secret data directly from the data hiding room. Beginning from the LSB planes of the replaceable pixels, the receiver uses the data hiding key K_h to retrieve the embedded secret bits from the data hiding room, so that the secret data are extracted.

Image Recovery: When the receiver has the block encryption key K_{en} and the block permutation key K_{bp}, the receiver can retrieve the original image from the marked encrypted image or generate the marked decrypted image, which still contains secret data and is highly similar to the original image. The procedure of image recovery is as follows:

Step 1: Extract the length information from the MSBs of the replaceable pixels. Then, extract the Huffman codebook and the bitstream of the compressed pixel differences BS_d according to the length information.

Step 2: According to the Huffman codebook, decompose BS_d into the encoded pixel differences $\left\{ed_1^{(1)}, ed_1^{(2)}, ed_1^{(3)}, ed_2^{(1)}, ed_2^{(2)}, ed_2^{(3)}, \ldots, ed_N^{(1)}, ed_N^{(2)}, ed_N^{(3)}\right\}$, and decode them into the original differences $\left\{d_1^{(1)}, d_1^{(2)}, d_1^{(3)}, d_2^{(1)}, d_2^{(2)}, d_2^{(3)}, \ldots, d_N^{(1)}, d_N^{(2)}, d_N^{(3)}\right\}$.

Step 3 : For each block $\boldsymbol{B}'_r$ containing the mark pixel $pe_r^{(4)}$, retrieve the original replaceable pixels $pe_r^{(1)}$, $pe_r^{(2)}$, and $pe_r^{(3)}$ as follows:

$$pe_r^{(i)} = pe_r^{(4)} + d_r^{(i)}, \ where \ i = 1, 2, 3, \ r = 1, 2, \ldots, N. \tag{5}$$

Step 4**:** After retrieving the original values of all the replaceable pixels, to recover the original image all the replaceable pixels in the marked encrypted image are substituted with their original values.

To generate the marked decrypted image, for each replaceable pixel the MSBs which are substituted with the auxiliary information are recovered according to the original value, and the LSBs which belong to the data hiding room stay the same.

Step 5: Use the block permutation key K_{bp} and the block encryption key K_{en} to decrypt the processed encrypted image into the original image or the marked decrypted image, which is highly similar to the original image.

3. Experimental Results

In this section, we evaluate the performance of the proposed method in terms of its security, embedding rate, and visual quality. Figure 5 shows the six standard grayscale images which are used to validate the performance of the proposed method [34].

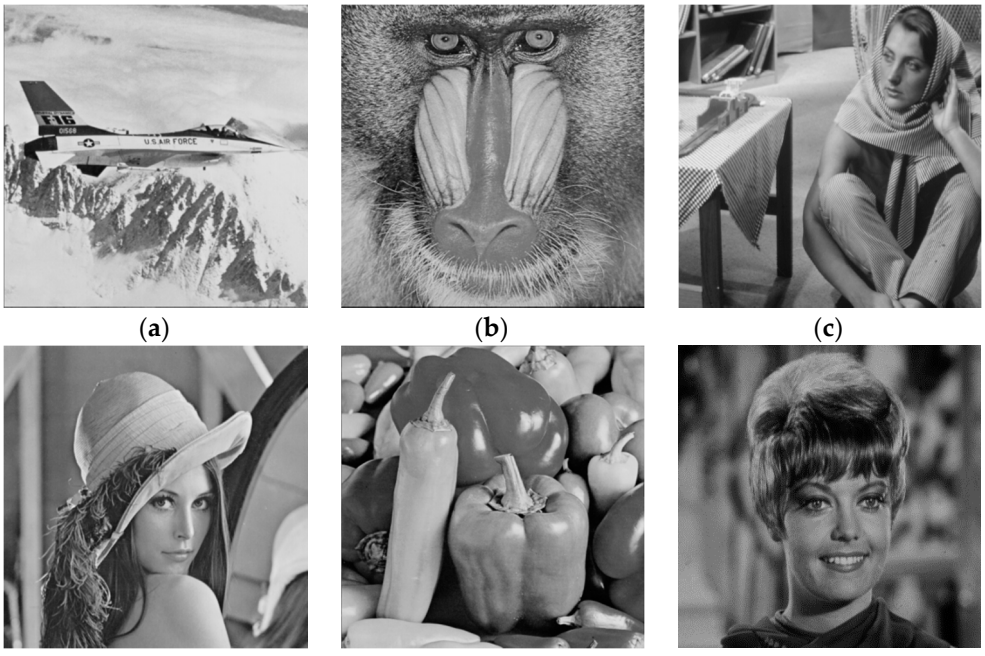

Figure 5. Six test images. (**a**) Airplane; (**b**) Baboon; (**c**) Barbara; (**d**) Lena; (**e**) Peppers; (**f**) Zelda.

3.1. Security Analysis

In the proposed method, the original image is encrypted by two keys: the block encryption key K_{en} which is used to encrypt the pixels at the level of a 2×2 block, and the block permutation key K_{bp} which is used to pseudo-randomly permute the blocks. For an image sized $H \times W$, $\lfloor H/2 \rfloor \times \lfloor W/2 \rfloor \times 8$ bits are required to encrypt the $\lfloor H/2 \rfloor \times \lfloor W/2 \rfloor$ blocks of the original image. After encrypting the pixels inside blocks, there are $(\lfloor H/2 \rfloor \times \lfloor W/2 \rfloor)!$ possible situations of block permutation. Therefore, there are $2^{8 \times \lfloor H/2 \rfloor \times \lfloor W/2 \rfloor} \times (\lfloor H/2 \rfloor \times \lfloor W/2 \rfloor)!$ possible situations of image encryption. For a standard 512×512 grayscale encrypted image, the number of its possible decryption results is $2^{524288} \times 65536! \approx 2^{1478325}$, which is very large number to ensure computational security. Without the block encryption key and the block permutation key, it is almost impossible to obtain the original image from the encrypted image.

Figures 6 and 7 show the histograms of the six original image and their encrypted versions, respectively. As shown in the figures, the pixel value distribution of each encrypted image is uniform and completely different from that of the original image. Therefore, it is almost impossible to obtain the information of the original image from the encrypted image by statistical attack.

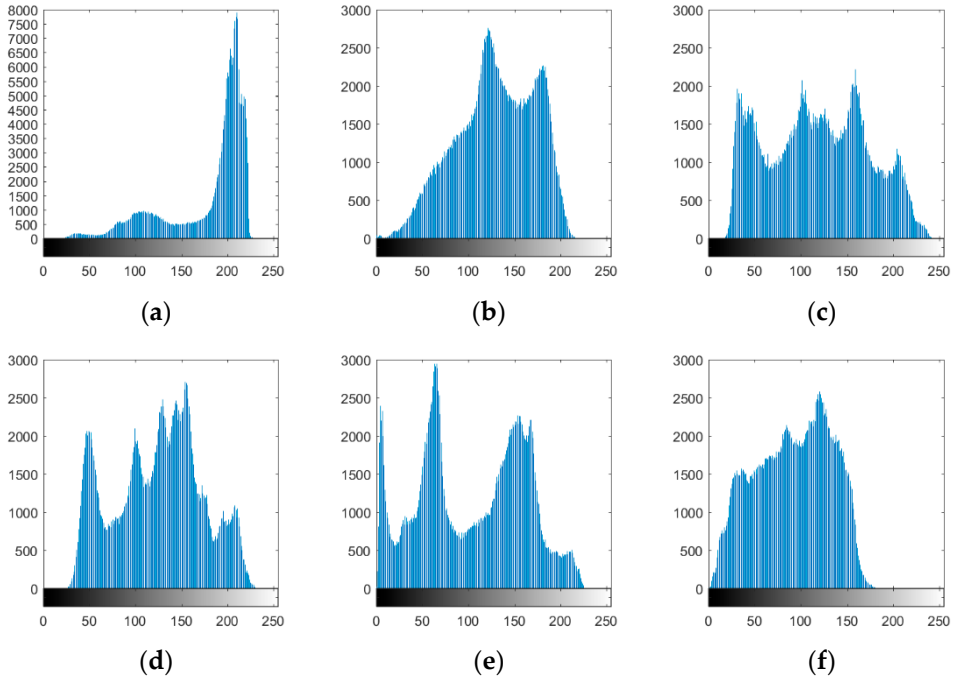

Figure 6. Histograms of the original image. (**a**) Airplane; (**b**) Baboon; (**c**) Barbara; (**d**) Lena; (**e**) Peppers; (**f**) Zelda.

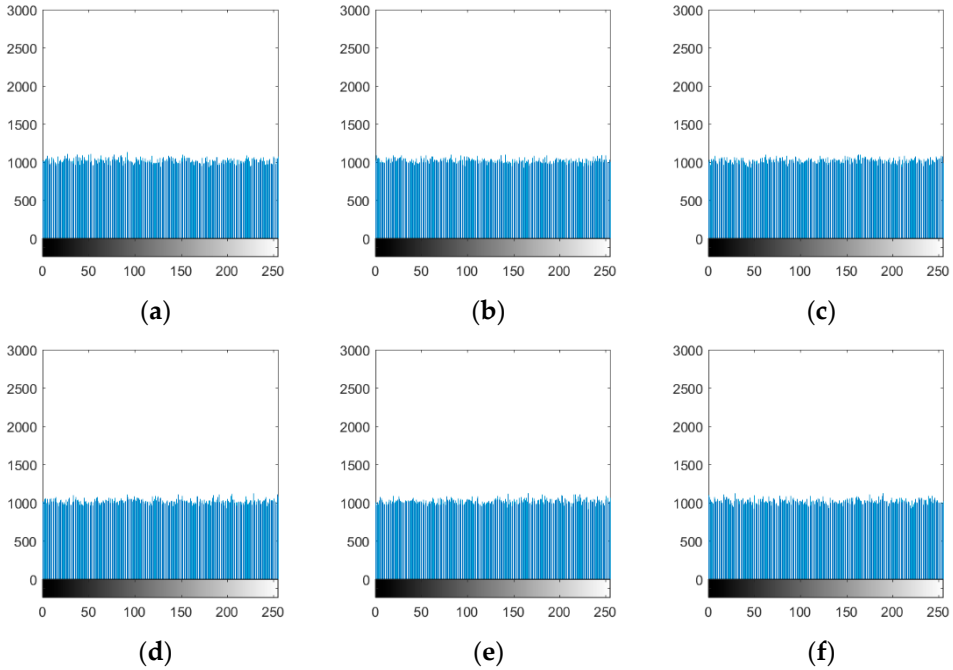

Figure 7. Histograms of the encrypted image. (**a**) Airplane; (**b**) Baboon; (**c**) Barbara; (**d**) Lena; (**e**) Peppers; (**f**) Zelda.

In the proposed method, though the image encryption scheme is based on stream cipher which cannot withstand differential attacks, it is very difficult for an attacker to perform differential attacks in the field of RDHEI. In the application scenarios of RDHEI, different content owners use their own image encryption keys to protect their own privacies, and only share the encryption keys to the receivers who can access the image content. The data hiders or other parties have no access to the image encryption keys and the content owners' original images, and they cannot provide similar plain images to the content owners or receivers to obtain the corresponding encrypted images for comparisons and differential attacks. Therefore, in general it is not feasible to perform differential attacks in RDHEI scenarios.

3.2. Performance Comparison

In this subsection, we conduct experiments to evaluate the embedding rate and visual quality of the proposed method, and perform a comparison between the proposed method and the related works in [10,12,14,21,23,26].

Table 1 shows the embedding rates of the proposed method and the existing RDHEI methods in [10,12,14,21,23,26]. As shown in the table, in most cases the proposed method can achieve relatively higher embedding rates compared with the other methods. For the smooth images such as Airplane, Lena, Peppers, and Zelda, the embedding rates of the proposed method can exceed 1.3 bpp, which is much higher than that of the other methods. For the complex images such as Baboon and Barbara, the proposed method can still achieve a relatively higher capacity than the other methods (except for Qian et al. [21] on Baboon). According to the experimental results, the data hiding room can be efficiently vacated in each block by Huffman coding.

Table 1. Comparison of the embedding rates (bpp).

Images	Zhang [10]	Wu et al. [14]	Liao et al. [12]	Qian et al. [21]	Xiao et al. [23]	Li et al. [26]	Proposed
Airplane	0.0191	0.0709	0.0345	0.2952	0.2799	0.6986	1.6965
Baboon	0.0047	0.0679	0.0096	0.2952	0.0467	0.2237	0.2902
Barbara	0.0138	0.0707	0.0239	0.2952	0.1637	0.4049	0.8555
Lena	0.0243	0.0709	0.0386	0.2952	0.2422	0.7701	1.4122
Peppers	0.0162	0.0709	0.0320	0.2952	0.2129	0.7418	1.3520
Zelda	0.0277	0.0709	0.0463	0.2952	0.2654	0.7797	1.6066

In Figure 8, the Peak Signal-to-Noise Ratio (PSNR) values of the marked decrypted images generated by the proposed method are compared with those of the marked decrypted images generated by the other methods. As shown in the figure, the proposed method can achieve higher PSNR values for different images and different embedding rates. That means that the proposed method outperforms the other methods in terms of the visual quality of the marked decrypted image. This is because the proposed method uses the LSB planes to embed secret data, and needs to modify only one bit to embed one secret bit. Compared with the proposed method, the other methods should use higher bit planes or modify more bits to embed one secret bit.

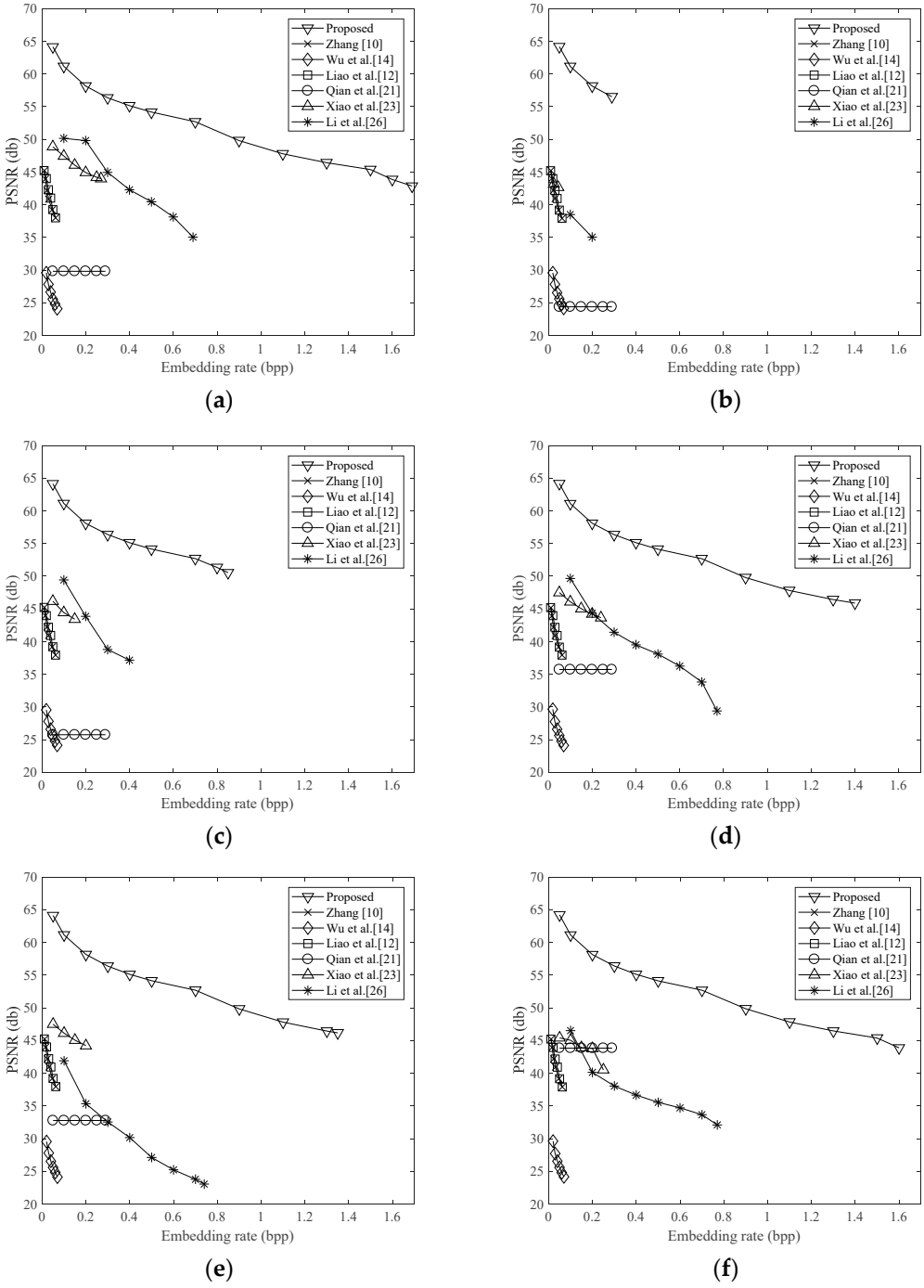

Figure 8. Comparison of the PSNR values: (**a**) Airplane; (**b**) Baboon; (**c**) Barbara; (**d**) Lena; (**e**) Peppers; (**f**) Zelda.

4. Conclusions

In this paper, a novel VRAE RDHEI method is proposed. In the proposed method, the original image is encrypted at the level of a 2×2 block to partially retain the spatial correlation inside the blocks. At the data hider's side, the pixels in each block are divided into one mark pixel and three replaceable pixels, and the replaceable pixels are replaced with their compressed pixel differences to vacate room in the encrypted image. At the receiver's side, the secret data can be extracted and the original image can be retrieved from the encrypted image with no errors. The experimental results

demonstrate that the proposed method can achieve higher embedding rates and better PSNR values compared to the existing RDHEI methods.

Funding: This paper was supported by the National Natural Science Foundation of China (U1936114, 61701191) and the Natural Science Foundation of Fujian Province, China (2019H0021).

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

Conflicts of Interest: The author declares no conflict of interest. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Shi, Y.Q.; Li, X.; Zhang, X.; Wu, H.-T.; Ma, B. Reversible Data Hiding: Advances in the Past Two Decades. *IEEE Access* **2016**, *4*, 3210–3237. [CrossRef]
2. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. [CrossRef]
3. Qiu, Y.; Qian, Z.; Yu, L. Adaptive reversible data hiding by extending the generalized integer transformation. *IEEE Signal Process. Lett.* **2016**, *23*, 130–134. [CrossRef]
4. Ni, Z.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–362. [CrossRef]
5. 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]
6. 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]
7. Qu, X.; Kim, H.J. Pixel-based pixel value ordering predictor for high-fidelity reversible data hiding. *Signal Process.* **2015**, *111*, 249–260. [CrossRef]
8. Hong, W.; Chen, T.; Shiu, C. Reversible data hiding for high quality images using modification of prediction errors. *J. Syst. Softw.* **2009**, *82*, 1833–1842. [CrossRef]
9. Carpentieri, B.; Castiglione, A.; Santis, A.D.; Palmieri, F.; Pizzolante, R. One-pass lossless data hiding and compression of remote sensing data. *Future Gener. Comput. Syst.* **2019**, *90*, 222–239. [CrossRef]
10. Zhang, X. Reversible data hiding in encrypted images. *IEEE Signal Process. Lett.* **2011**, *18*, 255–258. [CrossRef]
11. Hong, W.; Chen, T.; Wu, H. An improved reversible data hiding in encrypted images using side match. *IEEE Signal Process. Lett.* **2012**, *19*, 199–202. [CrossRef]
12. Liao, X.; Shu, C. Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. *J. Vis. Commun. Image Represent.* **2015**, *28*, 21–27. [CrossRef]
13. Qin, C.; Zhang, X. Effective reversible data hiding in encrypted image with privacy protection for image content. *J. Vis. Commun. Image Represent.* **2015**, *31*, 154–164. [CrossRef]
14. Wu, X.; Sun, W. High-capacity reversible data hiding in encrypted images by prediction error. *Signal Process.* **2014**, *104*, 387–400. [CrossRef]
15. Dragoi, I.C.; Coanda, H.G.; Coltuc, D. Improved Reversible Data Hiding in Encrypted Images Based on Reserving Room After Encryption and Pixel Prediction. In Proceedings of the 25th European Signal Processing Conference (EUSIPCO), Kos Island, Greece, 28 August–2 September 2017; pp. 2186–2190.
16. 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.* **2016**, *26*, 441–452. [CrossRef]
17. Zhang, X. Separable reversible data hiding in encrypted image. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 826–832. [CrossRef]
18. Qian, Z.; Dai, S.; Jiang, F.; Zhang, X. Reversible Data Hiding in Encrypted Images Based on Progressive Recovery. *IEEE Signal Process. Lett.* **2016**, *23*, 1672–1676. [CrossRef]
19. Qin, C.; Zhang, W.; Cao, F.; Zhang, X.; Chang, C.C. Separable reversible data hiding in encrypted images via adaptive embedding strategy with block selection. *Signal Process.* **2018**, *153*, 109–122. [CrossRef]
20. Zhang, X.; Qian, Z.; Feng, G.; Ren, Y. Efficient reversible data hiding in encrypted images. *J. Vis. Commun. Image Represent.* **2014**, *25*, 322–328. [CrossRef]

21. Qian, Z.; Zhang, X. Reversible data hiding in encrypted image with distributed source encoding. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 636–646. [CrossRef]
22. Huang, F.; Huang, J.; Shi, Y.Q. New Framework for Reversible Data Hiding in Encrypted Domain. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2777–2789. [CrossRef]
23. Xiao, D.; Xiang, Y.; Zheng, H.; Wang, Y. Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism. *J. Vis. Commun. Image Represent.* **2017**, *45*, 1–10. [CrossRef]
24. Yi, S.; Zhou, Y.; Hua, Z. Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion. *Signal Process. Image Commun.* **2018**, *64*, 78–88. [CrossRef]
25. Qin, C.; Qian, X.; Hong, W.; Zhang, X. An efficient coding scheme for reversible data hiding in encrypted image with redundancy transfer. *Inform. Sci.* **2019**, *487*, 176–192. [CrossRef]
26. 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]
27. Ge, H.; Chen, Y.; Qian, Z.; Wang, J. A High Capacity Multi-Level Approach for Reversible Data Hiding in Encrypted Images. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 2285–2295. [CrossRef]
28. 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]
29. Zhang, W.; Ma, K.; Yu, N. Reversibility improved data hiding in encrypted images. *Signal Process.* **2014**, *94*, 118–127. [CrossRef]
30. Cao, X.; Du, L.; Wei, X.; Meng, D.; Guo, X. High capacity reversible data hiding in encrypted images by patch-level sparse representation. *IEEE Trans. Cybern.* **2016**, *46*, 1132–1143. [CrossRef]
31. Yi, S.; Zhou, Y. Binary-block embedding for reversible data hiding in encrypted images. *Signal Process.* **2017**, *133*, 40–51. [CrossRef]
32. Chen, K.; Chang, C.C. High-capacity reversible data hiding in encrypted images based on extended Run-Length coding and block-based MSB plane rearrangement. *J. Vis. Commun. Image Represent.* **2019**, *58*, 334–344. [CrossRef]
33. Qiu, Y.; Qian, Z.; Zeng, H.; Lin, X.; Zhang, X. Reversible data hiding in encrypted images using adaptive reversible integer transformation. *Signal Process.* **2020**, *167*, 107288. [CrossRef]
34. Computer Vision Group Test Image Database. Available online: http://decsai.ugr.es/cvg/dbimagenes/g512.php (accessed on 15 June 2020).