mathematics

MDPI

Article

High-Capacity Reversible Data Hiding in Encrypted Images Based on Adaptive Predictor and Compression of Prediction Errors

Bin Huang, Chun Wan * and Kaimeng Chen

Computer Engineering College, Jimei University, Xiamen 361021, China; huangbin@jmu.edu.cn (B.H.); chenkaimeng@jmu.edu.cn (K.C.)

* Correspondence: wanchun@jmu.edu.cn

Abstract: Reversible data hiding in encrypted images (RDHEI) is a technology which embeds secret data into encrypted images in a reversible way. In this paper, we proposed a novel high-capacity RDHEI method which is based on the compression of prediction errors. Before image encryption, an adaptive linear regression predictor is trained from the original image. Then, the predictor is used to obtain the prediction errors of the pixels in the original image, and the prediction errors are compressed by Huffman coding. The compressed prediction errors are used to vacate additional room with no loss. After image encryption, the vacated room is reserved for data embedding. The receiver can extract the secret data and recover the image with no errors. Compared with existing approaches, the proposed method efficiently improves the embedding capacity.

Keywords: reversible data hiding; image encryption; linear regression; Huffman coding

Citation: Huang, B.; Wan, C.; Chen, K. High-Capacity Reversible Data Hiding in Encrypted Images Based on Adaptive Predictor and Compression of Prediction Errors. *Mathematics* **2021**, *9*, 2166. https://doi.org/10.3390/math9172166

Academic Editors: Ioana Boureanu and Liqun Chen

Received: 1 August 2021
Accepted: 2 September 2021
Published: 5 September 2021

1. Introduction

Reversible data hiding (RDH) is a technology that allows the reversible embedding of secret data into various carriers (such as digital images, texts, and videos) with no obvious distortion [1,2]. With the development of cloud services, more and more images are being stored and addressed in the cloud instead of user terminals. This brings the problem of privacy protection for the users. Image encryption is the most widely used technology to ensure the content security of image content. Currently, image encryption schemes are usually based on stream cipher [3], public key cryptosystem [4], or chaotic system [5]. Therefore, methods have been proposed that allow reversible data hiding in encrypted images, which allows the cloud (data hider) to embed secret data into encrypted images reversibly without image decryption. To date, the proposed RDHEI methods can be classified into three categories, i.e., (1) reserving room before image encryption (RRBE), (2) creating room by encryption (CRBE), and (3) vacating room after image encryption (VRAE).

In the RRBE method, the original image is pre-processed to vacate additional room before the image is encrypted. The vacated room is reserved after the image is encrypted, and this room can be used by the data hider for embedding data. Ma et al. [6] proposed the first RRBE method, and it divided the original image into a smooth area and a complex area. The least significant bits (LSBs) of the complex area are embedded into the smooth area using RDH methods for plaintext images, so that the spare LSBs can be used for embedding data after the image is encrypted. Based on Ma et al.'s method, several improved methods have been proposed [7–10]. In [7], the image was divided into three parts, and a more efficient RDH scheme was used for embedding LSBs. In [8], bicubic interpolation and partitioned local histogram shift were used for embedding LSBs. In [9], a reversible contrast mapping scheme was used for image partition, and the GRCM algorithm was used for embedding LSBs. In [10], the Paillier cryptosystem was used for image encryption, and a mirroring ciphertext group scheme was used so that the secret data could be extracted

from encrypted images or decrypted images. In [11], some pixels were selected from the original image, and they were estimated by their surrounding pixels to generate estimation errors. Histogram shift was used to modify these estimation errors for embedding data. In [12], an interpolation technique in [13] was used to generate interpolation error of pixels, and a designed histogram shift scheme was used for embedding data. In [14], the image was divided into patches, and each patch was encoded by the patches-level sparse coding to vacate additional room. In [15], the most significant bit (MSB) planes of the image were divided into non-overlapping blocks, and each block was encoded by a designed sparse matrix coding scheme to vacate room. In [16], an MSB predictor was designed to vacate room in the MSB plane of the original image. In [17], an MSB plane rearrangement scheme and extended run-length coding were used jointly to compress the MSB planes for additional room. In [18], the median edge detector (MED) was used to predict multi-MSBs of each pixel, and a set of Huffman coding-based labels was used to label the prediction result of each pixel for additional room. In [19], the image was divided into non-overlapping blocks, and an adaptive reversible integer transformation was used for each block to vacate embedding room.

In the CRBE methods, embedding room was created based on the designed image encryption scheme. At present, the idea that is used most extensively is that the image encryption scheme partially keeps the redundancy of the original image in the encrypted image. In [20–31], different block-level image encryption schemes were used to maintain the spatial correlation inside the blocks of the encrypted image. In [20], the image was encrypted by a stream encryption scheme that consisted of block-level permutation and block-level bit-XOR. The proposed histogram shifting-based RDH methods can be used directly on the encrypted image. In [21], a cross division image encryption scheme was used to keep spatial correlation in cross blocks, and a difference histogram shifting scheme was used for hiding data. In [22], a pixel value ordering scheme was designed to embed data into the 2×2 encrypted blocks. In [23], a block-level prediction-error expansion scheme was used to embed data into the 2×2 encrypted blocks. In [24], a run-length coding compression and a matrix compression were used to compress 2×2 encrypted blocks to create room for embedding data. In [25], a multi-level histogram shifting scheme was used for each encrypted block to embed data. In [26–28], different encoding schemes were used to compress the MSB planes of each encrypted block. In [29,30], for each encrypted block, one pixel was used to label the other pixels, so that additional bits could be spared for embedding data in the other pixels. In [31], a block histogram modification scheme was used to embed data into the LSB planes of each encrypted block. In [32,33], specific image encryption schemes were designed to transfer the redundancy of the original image into the encrypted image. In [32], a reversible image transformation scheme was used to encrypt images. After the images were encrypted, they were transformed other plaintext images as encrypted images, and traditional RDH methods can be used directly on the encrypted images. In [33], a reversible image reconstruction scheme was used to encrypt images. The original image was reconstructed into a meaningless redundancy image on which the traditional RDH methods were available.

In VRAE methods, the encrypted image has no spatial correlation or redundancy. The reversibility relies on the spatial correlation of the decrypted image. In [34], the image was divided into non-overlapping blocks. To embed one bit into one block, the three LSBs of half of the pixels in the block were flipped. At the receiver's side, first, the image was decrypted and then a smoothness estimator was used for each block to identify the flipped pixels so that the image could be recovered and the embedded bits could be extracted. This method was improved in [35–38]. In [35], an improved smoothness estimator and a side match scheme were used to reduce errors in the extraction of data and the recovery of images. In [36], a more precise estimator and a smoothness difference ordering scheme were used. In [37], the data embedding scheme was improved to reduce distortion, and a content-based predictor was used to estimate the smoothness. In [38], the data embedding scheme was improved, and an estimation scheme that used multiple judgements was used

to reduce the error rate. In [39], an LSB-swap scheme was used for embedding data to improve capacity. In [40], a public key modulation mechanism was used for embedding data, and a support vector machine (SVM) classifier was used for the extraction of data and to recover the image. In [41], pixels were divided into same-size groups for embedding data, and a designed predictor was used to recover the images. The method was extended in [42,43] for higher capacity. In [44], the LSBs of the image were compressed by using a compression matrix for embedding room, and an LSB predictor was used for the recovery of the compressed bits. The methods in [45,46] improved the method in [44]. In [45], the LSB planes of the pixels were divided into three subsets and compressed by three different compression matrices, and three different predictors were used in three rounds to recover the images. In [46], the images were divided into smooth blocks and complex blocks, and only the LSBs of the smooth blocks were compressed. In [47,48], LDPC encoding was used to directly compress the encrypted bits, and the uncompressed bits were used to recover the images.

Because these three types of the RDHEI methods embed secret data and achieve reversibility in completely different ways, the RRBE, CRBE, and VRAE methods cannot be replaced with each other. In RRBE methods, so far, the existing methods use the redundancy and spatial correlation of the original image to vacate room for embedding data. Some methods vacate room by lossless compression [14–18]. In these methods, the key point is how to efficiently encode the redundancy information by fewer bits.

Aiming to design a more efficient encoding scheme to vacate larger capacity, in this paper, a novel RRBE RDHEI method was proposed based on linear regression and Huffman coding. Due to the lightweight and fast training of linear regression models, the cost of training a specific linear regression model for an original image is acceptable. The linear regression model can be used as a pixel predictor. Based on the predictor, most of the prediction errors of the pixels are concentrated in a small range centered on 0. These prediction errors can be encoded efficiently by using Huffman coding. By using the predictor, the Huffman codeword table, and the encoded prediction errors, a significant amount of room can be vacated in the original image for embedding data without losing information. The main contributions of the proposed method are as follows:

(1) A scheme is proposed for vacating high-capacity data hiding room in the original image, and the scheme is based on linear regression and Huffman coding. The scheme can work effectively for images that contain large complex regions.
(2) A novel RRBE RDHEI method is proposed based on the scheme of vacating room. Compared with the existing RDHEI methods, the proposed method can use redundancy of the original image more efficiently to vacate larger room for data embedding. The experimental results show that the proposed method can achieve a higher embedding rate and better visual quality than the related methods. The extraction of data extraction and the recovery of images are separable and error-free.

The rest of this paper is organized as follows. Section 2 presents the details of the proposed method. Section 3 shows the experimental results and provides a comparison of the proposed method with existing methods. Section 4 presents our conclusions concerning the proposed method.

2. Proposed Method

In this section, the details of the proposed RDHEI method are presented. Figure 1 shows the framework of the proposed method. The content owner first trains a linear regression model from the original image. By using the linear regression model and Huffman coding, embedding room is vacated in the original image. Then, the image is encrypted and sent to the data hider. The data hider retrieves the embedding room and embeds secret data into it. At the receiver's side, the data hiding key is used to retrieve the secret data, and image encryption key is used to recover the original image.

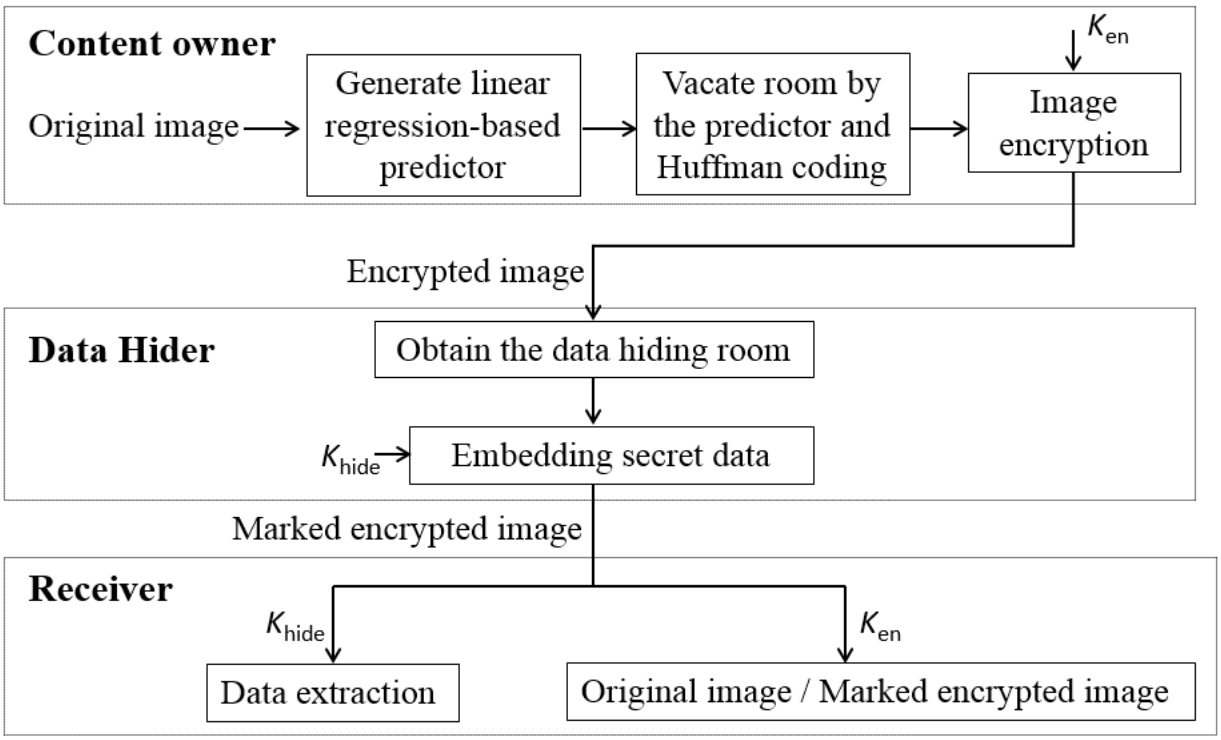


Figure 1. This is a figure. Schemes follow the same formatting.

2.1. Content Owner's Work

2.1.1. Generating of the Linear Regression-Based Predictor

In the proposed method, to vacate a large amount of embedding room, an accurate pixel predictor is required to generate concentrated prediction errors. To generate the predictor, a linear regression model is trained based on the original image. Figure 2 shows that the linear regression model predicts one pixel by using its three neighboring pixels, and the predicted value is calculated as follows:

$$p_x = w_0 + w_1 p_1 + w_2 p_2 + w_3 p_3. \quad (1)$$

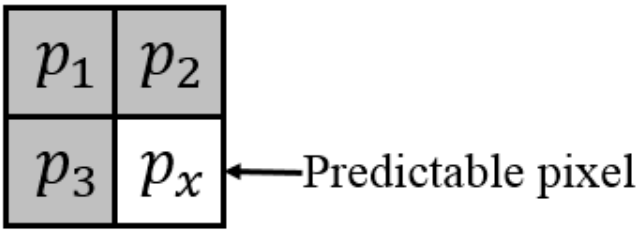


Figure 2. Predictable pixel and its three neighboring pixels.

To acquire the applicable four coefficients, i.e., w_0, w_1, w_2, and w_3, for the original image, the linear regression model in Equation (1) is trained using the training dataset that was constructed from the original image. The training dataset was constructed as follows. Figure 3 shows that the original image is divided into two parts, i.e., (1) the reference pixels that contain all pixels in the first row and the first column of the original image and (2) the predictable pixels that contain the other pixels of the original image. Denoting the training set as $\boldsymbol{D} = \{\boldsymbol{X}; \boldsymbol{Y}\}$, Figure 2 shows that the target set $\boldsymbol{Y}$ consists of all predictable pixels,

and the feature set $\boldsymbol{X}$ consists of the three neighbor pixels of each predictable pixel. For an original image that has the size of $H \times W$, the training set $\boldsymbol{D} = \{\boldsymbol{X}; \boldsymbol{Y}\}$ is constructed as:

$$\boldsymbol{D} = \left\{ p_{i,j}, p_{i,j+1}, p_{i+1,j}; p_{i+1,j+1} \middle| 1 \le i \le H-1, 1 \le j \le W-1 \right\}, \quad (2)$$

where $p_{i,j}$ is the pixel at the coordinates (i, j) of the original image.

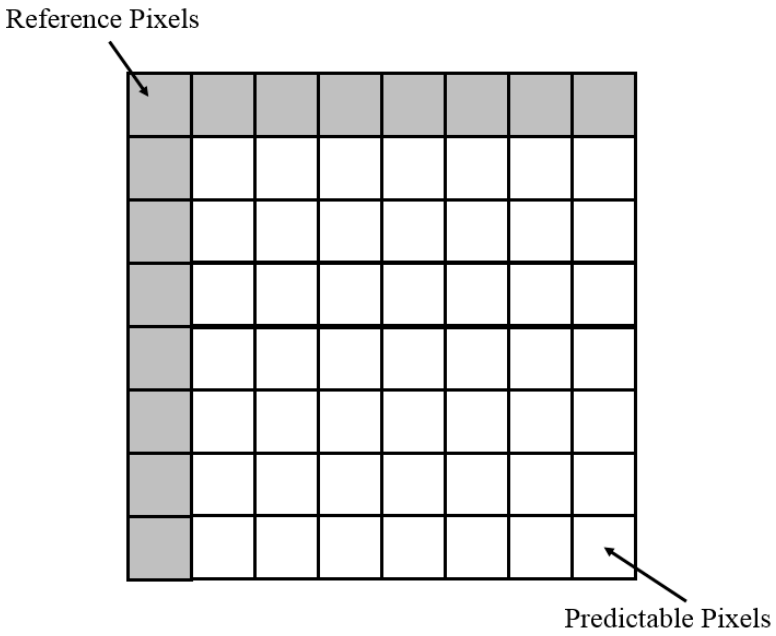


Figure 3. Reference pixels and predictable pixels.

Based on the training set $\boldsymbol{D}$, a linear regression model $\boldsymbol{LM} = \{w_0, w_1, w_2, w_3\}$ can be trained by any linear regression algorithm. Using the model $\boldsymbol{LM}$ as the predictor, the prediction errors of all predictable pixels can be calculated and encoded to vacate room in the original image. The details are provided in Section 2.1.2.

2.1.2. Vacating Room for Hiding Data

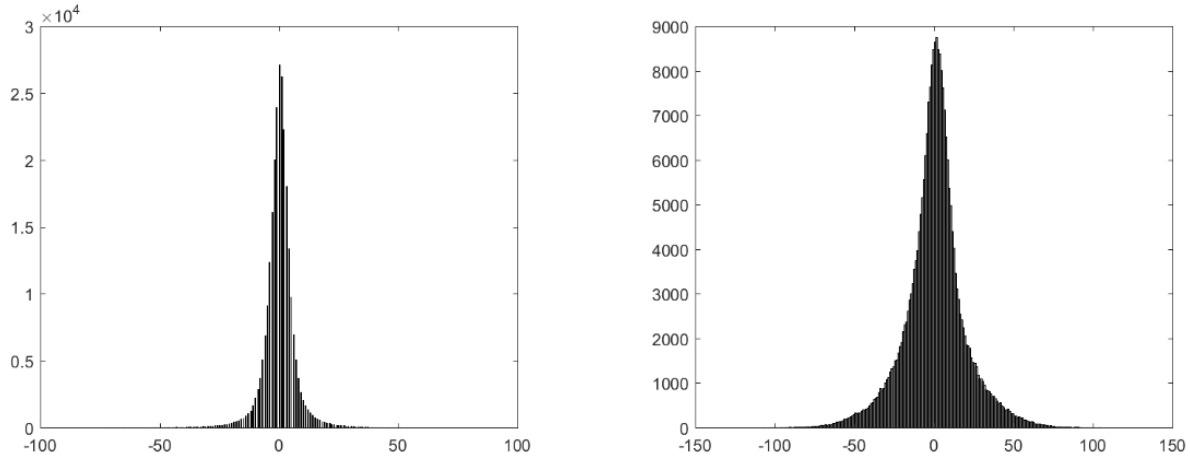
Using the linear regression predictor $\boldsymbol{LM} = \{w_0, w_1, w_2, w_3\}$, which is trained from the original image, the predicted value $p^e_{i,j}$ of the predictable pixel $p_{i,j}$ is calculated by:

$$p^e_{i,j} = \text{round}(w_0 + w_1 p_{i-1,j-1} + w_2 p_{i-1,j} + w_3 p_{i,j-1}) \text{ , where } 2 \le i \le H, 2 \le j \le W. \quad (3)$$

The prediction error $e_{i,j}$ of $p_{i,j}$ is calculated by:

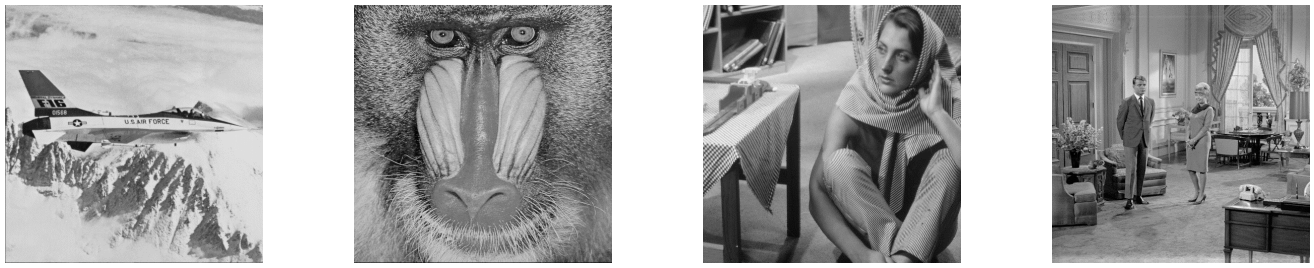
$$e_{i,j} = p_{i,j} - p^e_{i,j}, \text{ where } 2 \le i \le H, 2 \le j \le W. \quad (4)$$

For a standard 8-bit grayscale image, the range of $e_{i,j}$ is $[-255, 255]$. However, in most cases, $p^e_{i,j}$ is close to $p_{i,j}$, so the most prediction errors are concentrated in a small range around 0. Figure 4 shows the prediction error histograms of Baboon and Lena (Figure 5b,f in Section 4). As shown in the figure, the prediction errors are highly concentrated. Therefore, all of the prediction error information of the image has a lot of redundancy and can be compressed efficiently. For each predictable pixel $p_{i,j}$, the original value can be calculated by the predicted value $p^e_{i,j}$, and the prediction error $e_{i,j}$. If all of the reference pixels maintain their original values, all of the predictable pixels can be recovered, row by row, using the predictor $\boldsymbol{LM}$ and the prediction errors. Therefore, all of the predictable pixels in the original image can be replaced by the predictor $\boldsymbol{LM}$ and the compressed prediction errors to vacate room for hiding data without losing any information.



(**a**) Lena (**b**) Baboon

Figure 4. Prediction error histograms of Lena and Baboon.



(**a**) *Airplane* (**b**) *Baboon* (**c**) *Barbara* (**d**) *Couple*

(**e**) *Crowd* (**f**) *Lena* (**g**) *Man* (**h**) *Peppers*

Figure 5. The eight test images.

In the proposed method, the prediction errors are compressed by Huffman coding. Huffman coding is a variable length code scheme which construct the codewords with the shortest average code length according to the probability of the appearance of each character. The predictor ***LM*** and the compressed prediction errors are stored in the MSBs of the predictable pixels, and the LSBs of the predictable pixels are used by the data hider for embedding secret data after the image has been encrypted.

For an original image I sized $H \times W$, the detail procedure of vacating room is as follows:

Step 1: Calculate all the prediction errors $e_{2,2}, e_{2,3}, \ldots, e_{H,W-1}, e_{H,W}$ according to Equations (1) and (2) using the predictor ***LM***.

Step 2: According to the value distribution of the prediction errors, encode all of the prediction errors using Huffman coding. Connect all the encoded prediction errors row by row to form the bitstream $\boldsymbol{BS}_{pe} = C(e_{2,2})C(e_{2,3}) \ldots C(e_{H,W-1})C(e_{H,W})$, where $C(e_{i,j})$ is the Huffman codeword of $e_{i,j}$.

Step 3: Construct the Huffman codebook of the encoded prediction errors. The codebook $\boldsymbol{CB} = \{N, (V_1, L_1, C_1), (V_2, L_2, C_2), \ldots, (V_N, L_N, C_N)\}$, where N is the number of Huffman codewords, V_i is the value of the prediction error, L_i is the length of the codeword, and C_i is the codeword of V_i.

Step 4: From the most significant bit (MSB) plane to the lower bit plane, embed the length information L, the predictor $\boldsymbol{LM}$, the Huffman codebook $\boldsymbol{CB}$, and the encoded prediction error bitstream $\boldsymbol{BS_{pe}}$ into the MSBs of all of the predictable pixels. The remnant bits of the predictable pixels are used as data hiding room at the data hider's side.

Figure 6 shows an example of vacating room in a 5×5 image. To simplify the statement, we assumed that the trained linear regression-based predictor $\boldsymbol{LM} = \{w_0 = 0, w_1 = 0, w_2 = 1, w_3 = 0\}$ and that its binary representation is $(0010)_2$. Using the predictor $\boldsymbol{LM}$, the prediction errors of the predictable pixels were calculated and compressed by Huffman coding. In the Huffman codebook, 2 bits are used for N, 3 bits are used for V_i (the first bit is the sign bit, and the last two bits are the absolute value), and 2 bits are used for L_i. After encoding the prediction errors and constructing the Huffman codebook, the length information, the codebook, and the encoded prediction error bitstream are embedded into 8th, 7th, and 6th MSB planes of the predictable pixels, and the rest of the bits of the predictable pixels can be used for embedding data.

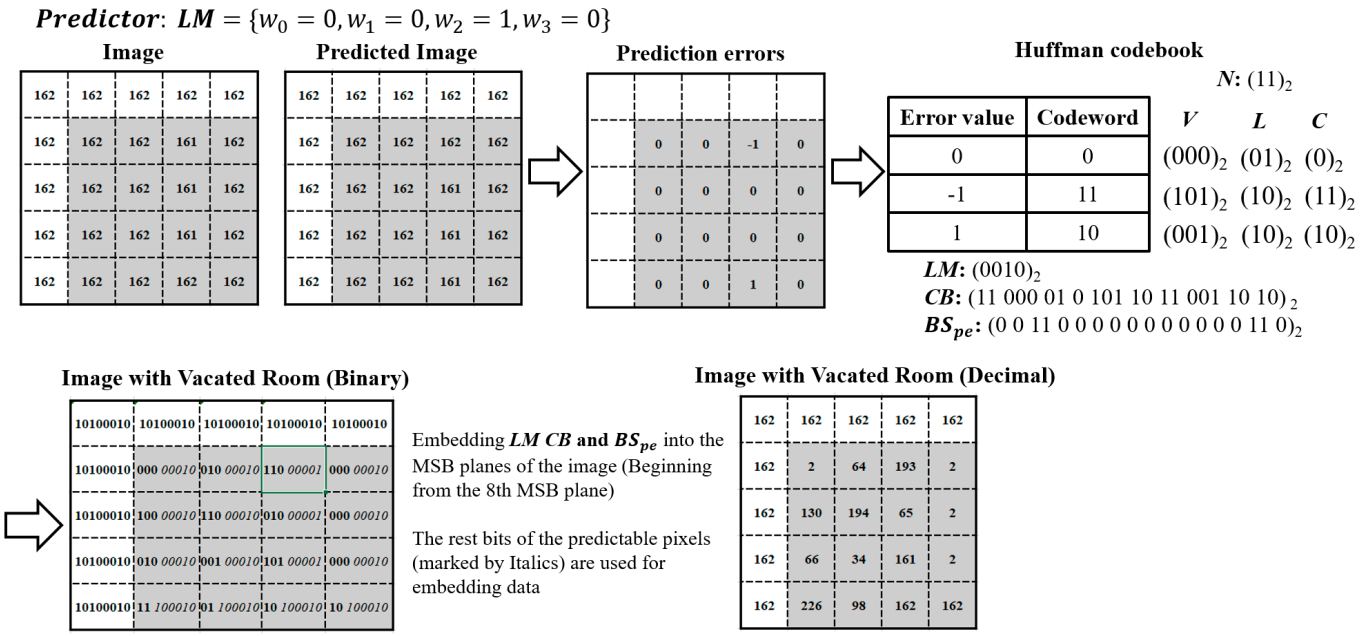


Figure 6. Example of the proposed method.

2.1.3. Image Encryption

After vacating room, the vacated image is encrypted by a stream cipher. Using the encryption key K_{en}, eight pseudo-random, $r_{i,j,1}, r_{i,j,2}, \ldots, r_{i,j,8}$, are generated for each pixel $p_{i,j}$ in the vacated image. Decompose $p_{i,j}$ into 8 bits $b_{i,j,1}, b_{i,j,2}, \ldots, b_{i,j,8}$ as follows:

$$b_{i,j,k} = p_{i,j}/2^{k-1} \bmod 2, \text{ where } k = 1, 2, \ldots, 8. \tag{5}$$

$p_{i,j}$ is encrypted into $E_{i,j}$ as follows:

$$e_{i,j,k} = b_{i,j,k} \oplus r_{i,j,k}, k = 1, 2, \ldots 8 \tag{6}$$

$$E_{i,j} = \sum_{k=1}^{8} e_{i,j,k} \times 2^{k-1} \tag{7}$$

After the image is encrypted, a certain number of the LSBs of the predictable pixels are replaced with capacity information to enable the data hider to obtain room for hiding data. Finally, the pre-processed encrypted image can be sent to the data hider for embedding the secret data.

In the proposed method, an image sized $H \times W$ is encrypted bit by bit with a pseudo-random binary sequence of length $H \times W \times 8$. In the binary sequence, each bit has almost the same possibility of being 0 or 1. Therefore, without the encryption key, the attacker should test $2^{H \times W \times 8}$ possible binary sequences to find out the correct decrypted image from the encrypted image. For the standard grayscale image sized 512×512, the number of possible binary sequences is $2^{2097152}$—it is almost impossible to test all possible sequences within an acceptable time. Therefore, the image encryption of the proposed method is strong enough to protect the content security.

2.2. Data Hider's Work

When the data hider receives the encrypted image from the content owner, the data hider can obtain room for hiding data by extracting the capacity information from the LSBs of the encrypted image. All of the bits in the data hiding room can be used for embedding data. The secret data are embedded into the data hiding room by bit substitution. Using the data hiding key K_{hide}, the data hider pseudo-randomly selects the bits in the data hiding room (from the LSB plane to the higher bit plane) and replaces these bits with the secret data. To enhance the security of the secret data, they also can be encrypted before being embedded into the data hiding room. The key used to encrypt the secret data must be shared in advance by the data hider and the data receiver.

2.3. Receiver's Work

The receiver acquires the marked encrypted image that contains the secret data from the data hider. Using different keys, the receiver can retrieve the embedded data from the image without decryption, recover the original image, or generate a marked decrypted image that contains the secret data and is highly similar to the original image.

(1) Data Extraction. When the receiver has the data hiding key K_{hide}, the receiver can extract the embedded data directly from the marked encrypted image. First, the receiver extracts the capacity information to obtain the room that is available for hiding data. Then, the receiver uses K_{hide} to extract the bits in the data hiding room to obtain the embedded data.

(2) Image Recovery. When the receiver has the image encryption key K_{en}, the receiver can reconstruct the original image or generate a marked decrypted image with the embedded data. The detailed procedure is as follows: Step 1: Decrypt the marked encrypted image by K_{en}. Step 2: Extract the predictor $\boldsymbol{LM} = \{w_0, w_1, w_2, w_3\}$, the Huffman codebook $\boldsymbol{CB}$, and the encoded prediction error bitstream $\boldsymbol{BS_{pe}}$ from the MSBs of the decrypted predictable pixels. Step 3: According to the Huffman codebook $\boldsymbol{CB}$, decode the bitstream $\boldsymbol{BS_{pe}}$, into the original prediction errors $\boldsymbol{PE} = \{e_{2,2}, e_{2,3}, \ldots\ldots, e_{H,W-1}, e_{H,W}\}$. Step 4: Use the predictor $\boldsymbol{LM}$, the prediction errors $\boldsymbol{PE}$ and the reference pixels to retrieve the original values of all predictable pixels row by row and column by column. Step 5: In the decrypted image, if all predictable pixels are recovered directly to their original values, the original image is reconstructed with no error; if only the MSBs of the predictable pixels that were modified for vacating room (Section 2.1.2) are recovered according to the original values, the marked decrypted image is generated, which is highly similar to the original image, and it still keeps the embedded data in the LSB planes.

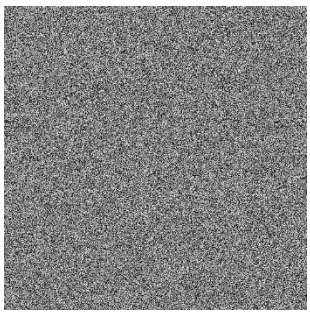
3. Experimental Results and Comparison

In this section, the experimental results and comparisons are provided to verify the effectiveness of the proposed method. The experiments were performed on eight standard

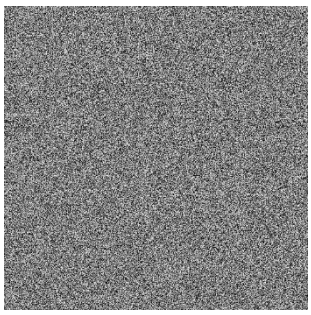
test images, as shown in Figure 5 [49]. The typical and state-of-the-art RDHEI methods in [15,17,18,21,34,41] were used as the competitors.

Figure 7 shows the encrypted image, the marked encrypted image, the marked decrypted image, and the recovered image of *Lena* in the proposed method. Figure 7b is the encrypted image after the image has been preprocessed and encrypted. Figure 7c is the marked encrypted image with the embedding rate of 1 bpp after embedding the data. Figure 7d is the marked decrypted image generated from Figure 7c. The PSNR value of Figure 7d is 51.10 dB. Figure 7e is the recovered image, which is the same as Figure 7a.

(**a**) Original image



(**b**) Encrypted image



(**c**) Marked encrypted image

(**d**) Marked decrypted image

(**e**) Recovered image

Figure 7. Experimental images of *Lena* in the proposed method.

Table 1 shows the embedding rates of the proposed method and the methods in [15,17,18,21,34,41] on the eight test images in Figure 5. The embedding rates are measured by bit per pixel (bpp), i.e., the ratio of the number of secret bits to the number of image pixels. In the proposed method, 128 bits were used to represent the linear regression-based predictor (32 bits for each coefficient), and 9 bits and 5 bits were used to represent the prediction error value and the length of the Huffman codeword, respectively, in the Huffman codebook. In Zhang's method [34], the block size was set to 4×4. In Wu and Sun's method [41], the pixel group consisted of seven pixels, and the 6th LSB plane was used for embedding data. Because extracted-bit errors may occur in Zhang's method [34] and in Wu and Sun's method [41], the pure embedding rates in these methods are calculated by multiplying the original embedding rate by $(1 - H(p)) \times ER_{\mathrm{O}}$, where ER_{O} is the original embedding rate, and $H(p)$ is the binary entropy for the extract-bit error rate p [15]. As shown in the table, the proposed method for the eight images can achieve higher embedding rate than the other methods. For the images that contain large smooth regions, such as *Airplane*, *Crowd*, and *Lena*, the proposed method can achieve embedding rates higher than 3.4 bpp. For the images containing many complex regions, such as *Baboon*, the proposed method still works efficiently to achieve an embedding rate higher than 1.7 bpp, which is much higher than the other methods. Since the proposed method directly respectively trains the specific linear regression-based predictor for each original image, the predictor can be more accurate than the conventional predictor for all imagesm such as median edge detector. In smooth images such as *Airplane*, the predictor can be very precise. The prediction errors

can be highly concentrated in a small range so that they can be compressed efficiently. In complex images such as *Baboon*, it is hard to predict pixel values precisely, and the prediction errors cannot be as concentrated as those in smooth images. However, the predictor still reduces the errors as much as possible for higher capacity. Therefore, the proposed method can efficiently use the redundancy of the original image to vacate large room for data embedding.

Table 1. Comparison of embedding rates for the eight test images.

Images	Airplane	Baboon	Barbara	Couple	Crowd	Lena	Man	Peppers
Zhang [34]	0.034	0.005	0.017	0.017	0.029	0.030	0.014	0.019
Wu and Sun [41]	0.070	0.066	0.070	0.070	0.070	0.070	0.070	0.070
Li et al. [21]	0.698	0.223	0.404	0.732	0.761	0.770	0.558	0.741
BBE [15]	2.204	0.568	1.319	1.287	1.735	1.819	1.607	1.820
Chen et al. [17]	2.340	0.535	1.409	1.398	1.947	1.944	1.678	1.879
Yin et al. [18]	3.092	1.098	1.902	2.385	3.004	2.614	2.175	2.299
Proposed	3.711	1.745	2.408	3.021	3.545	3.413	2.733	3.088

Table 2 shows an average time of the proposed method in training linear regression-based predictor, vacating room, image encryption, and data hiding (1 bpp). The experimental environment is a computer with Intel i9-10920X 3.5GHz CPU, 32GB RAM, and MATLAB R2017a. As shown in the table, training a linear regression-based predictor on an original image for vacating room is quick and practical.

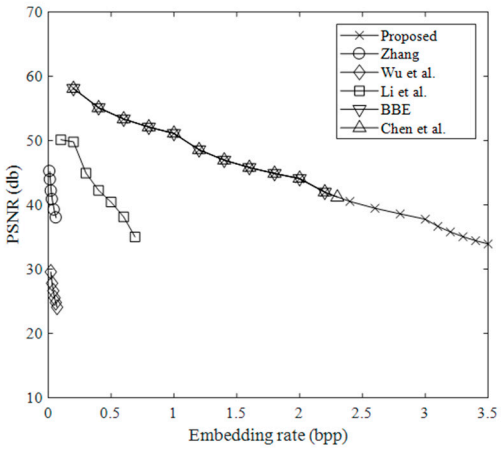
Table 2. Comparison of embedding rates for the eight test images (second).

Images	Airplane	Baboon	Barbara	Couple	Crowd	Lena	Man	Peppers
Linear regression	0.063	0.065	0.067	0.066	0.065	0.065	0.066	0.066
Vacating room	3.168	4.266	4.392	3.787	3.140	3.390	4.420	3.473
Image encryption	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008
Data hiding(1 bpp)	0.117	0.117	0.117	0.117	0.117	0.117	0.117	0.117

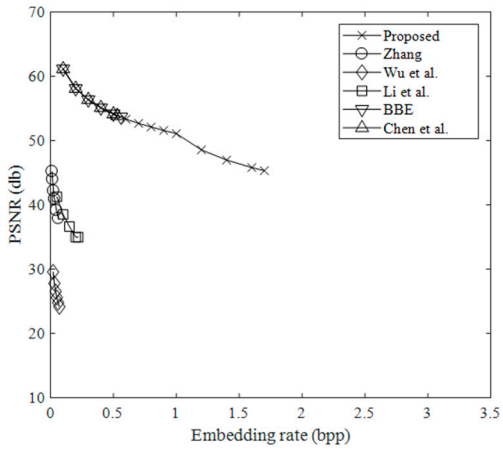
Figures 8 and 9 show the comparison of the marked decrypted image quality measured by peak signal to noise ratio (PSNR) in the proposed method and the methods in [15,17,18,21,34,41]. PSNR is a standard for image distortion evaluation. PSNR higher than 40 dB means that the quality of the marked decrypted image is very close to the original image. PSNR value is calculated by

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{(2^n - 1)^2}{MSE} \tag{8}$$

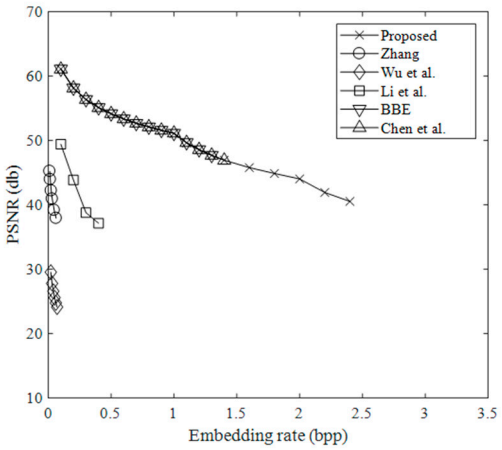
where *MSE* is the mean square error between all pixel values of the original image and the marked decrypted image. Yin et al.'s method [15] is not included in the comparison because, according to its original publication, the method directly embeds secret data into MSB planes. After image decryption, the method cannot generate a marked decrypted image. As shown in the figure, the proposed method can achieve a relatively high visual quality at high embedding rates. The average PSNR of the proposed method was 51 dB/44 dB when the embedding rate was 1 bpp/2 bpp. For different embedding rates, the PSNRs of the proposed method are higher than those of the methods in [21,34,41], and they are almost the same as the PSNRs of BBE [15] and Chen et al.'s method [17]. This is because the proposed method embeds the secret bits in the LSB planes by bit substitution, which is the same as BBE [15] and Chen et al.'s method [17].



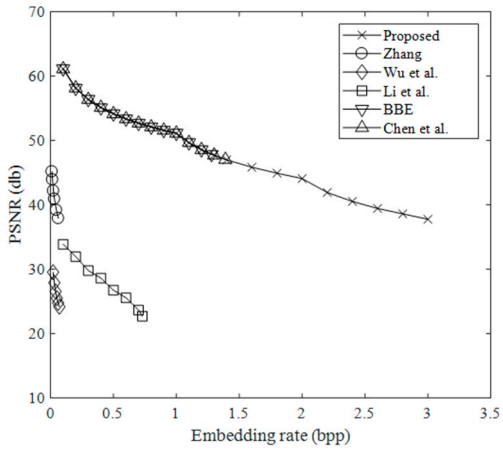
(**a**) *Airplane*



(**b**) *Baboon*

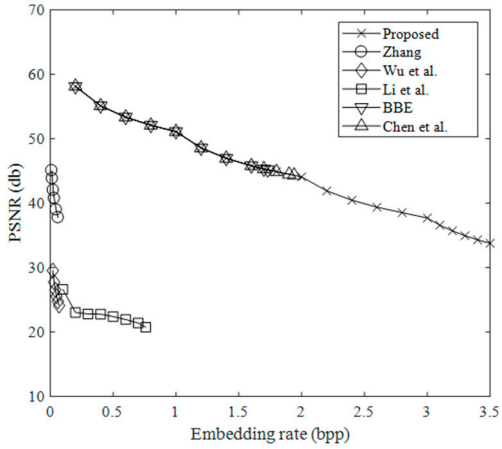


(**c**) *Barbara*

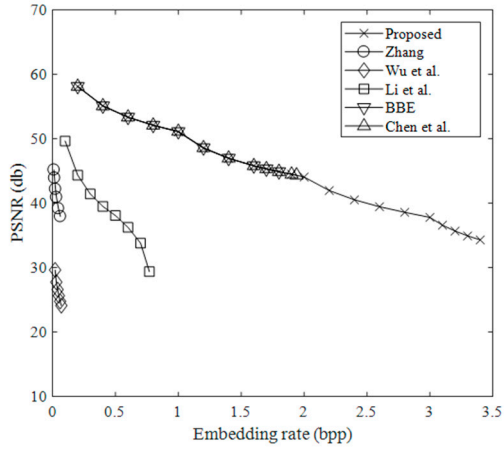


(**d**) *Couple*

Figure 8. Comparison of the marked decrypted image quality (*Airplane*, *Baboon*, *Barbara*, *Couple*).

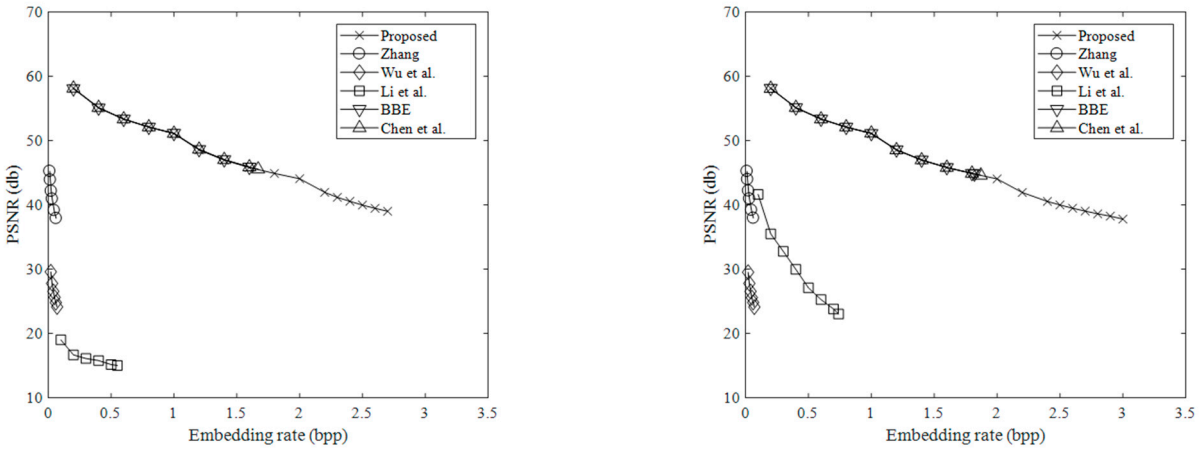


(**a**) *Crowd*



(**b**) *Lena*

Figure 9. *Cont*.



(**c**) *Man* (**d**) *Peppers*

Figure 9. Comparison of the marked decrypted image quality (*Crowd*, *Lena*, *Man*, *Peppers*).

4. Conclusions

In this paper, a new, high-capacity RRBE RDHEI method is proposed that is based on linear regression and Huffman coding. Before image encryption, a linear regression-based predictor is generated from the original image. Using the predictor, most prediction errors are concentrated in a small range, and they are compressed efficiently by Huffman coding. By substituting the original information of the predicted pixels with the compressed prediction errors and the auxiliary information, a large-capacity embedding room is made available. Compared with the related works, the proposed method can achieve higher embedding capacity, and vacate room efficiently for images with different contents.

In future works, we will consider designing more accurate predictors, and introducing these predictors into the RRBE RDHEI for higher capacity. Additionally, we will consider designed high-capacity CRBE RDHEI methods, which are based on prediction error compression schemes, specific image encryption schemes, and high accurate predictors.

Author Contributions: Conceptualization, B.H.; Funding acquisition, K.C.; Investigation, B.H.; Methodology, B.H. and C.W.; Project administration, K.C.; Software, B.H. and C.W.; Supervision, C.W.; Writing—original draft, B.H., C.W., and K.C.; Writing—review & editing, B.H., C.W. and K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Natural Science Foundation of Fujian Province, China grant number 2020J01698, 2019H0021.

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

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. Ni, Z.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–362.
3. Ding, Y.; Tan, F.; Qin, Z.; Cao, M.; Choo, K.; Qin, Z. DeepKeyGen: A Deep Learning-Based Stream Cipher Generator for Medical Image Encryption and Decryption. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–15. [CrossRef]
4. Ye, G.; Jiao, K.; Huang, X. Quantum logistic image encryption algorithm based on SHA-3 and RSA. *Nonlinear Dynam.* **2021**, *104*, 2807–2827. [CrossRef]
5. Guesmi, R.; Farah, M.A.B. A new efficient medical image cipher based on hybrid chaotic map and DNA code. *Multimed. Tools Appl.* **2021**, *80*, 1925–1944. [CrossRef]
6. 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]
7. Song, C.; Zhang, Y.; Lu, G. Reversible data hiding in encrypted images based on image partition and spatial correlation. *Int. Workshop Digit. Watermarking* **2018**, *11378*, 180–194.

8. Wang, X.; Han, X.; Xi, J.; Wang, S. Reversible data hiding in encrypted image with separable data extraction from image decryption. *Multimed. Tools Appl.* **2017**, *76*, 6127–6142. [CrossRef]
9. Qiu, Y.; Wang, H.; Wang, Z.; Qian, Z.; Feng, G.; Zhang, X. Reversible contrast mapping based reversible data hiding in encrypted images. In Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, 11–13 December 2018; pp. 1–7.
10. Xiang, S.; Luo, X. Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 3099–3110. [CrossRef]
11. Zhang, W.; Ma, K.; Yu, N. Reversibility improved data hiding in encrypted images. *Signal. Process.* **2014**, *94*, 118–127. [CrossRef]
12. Xu, D.; Wang, R. Separable and error-free reversible data hiding in encrypted images. *Signal. Process.* **2016**, *123*, 9–21. [CrossRef]
13. Luo, L.X.; Chen, Z.Y.; Chen, M. Reversible image watermarking using interpolation technique. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 187–193.
14. 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]
15. Yi, S.; Zhou, Y. Binary-block embedding for reversible data hiding in encrypted images. *Signal. Process.* **2017**, *133*, 40–51. [CrossRef]
16. 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]
17. 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]
18. Yin, Z.; Xiang, Y.; Zhang, X. Reversible data hiding in encrypted images based on multi-msb prediction and huffman coding. *IEEE Trans. Multimed.* **2020**, *22*, 874–884. [CrossRef]
19. 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]
20. 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]
21. 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]
22. 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]
23. 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]
24. Qin, C.; He, Z.; Luo, X.; Dong, J. Reversible data hiding in encrypted image with separable capability and high embedding capacity. *Inform. Sci.* **2018**, *465*, 285–304. [CrossRef]
25. 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]
26. Fu, Y.; Kong, P.; Yao, H.; Tang, Z.; Qin, C. Effective reversible data hiding in encrypted image with adaptive encoding strategy. *Inform. Sci.* **2019**, *494*, 21–36. [CrossRef]
27. Liu, Z.-L.; Pun, C.-M. Reversible data-hiding in encrypted images by redundant space transfer. *Inform. Sci.* **2018**, *433*, 188–203. [CrossRef]
28. 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]
29. Yi, S.; Zhou, Y. Parametric reversible data hiding in encrypted images using adaptive bit-level data embedding and checkerboard based prediction. *Signal. Process.* **2018**, *150*, 171–1828. [CrossRef]
30. Yi, S.; Zhou, Y. Separable and reversible data hiding in encrypted images using parametric binary tree labeling. *IEEE Trans. Multimed.* **2019**, *21*, 51–64. [CrossRef]
31. Yin, Z.; Abel, A.; Tang, J.; Zhang, X.; Luo, B. Reversible data hiding in encrypted images based on multi-level encryption and block histogram modification. *Multimed. Tools Appl.* **2017**, *76*, 3899–3920. [CrossRef]
32. Zhang, W.; Wang, H.; Hou, D.; Yu, N. Reversible data hiding in encrypted images by reversible image transformation. *IEEE Trans. Multimed.* **2016**, *18*, 1469–1479. [CrossRef]
33. Liu, Z.-L.; Pun, C.-M. Reversible image reconstruction for reversible data hiding in encrypted images. *Signal. Process.* **2019**, *161*, 50–62. [CrossRef]
34. Zhang, X. Reversible data hiding in encrypted images. *IEEE Signal. Process. Lett.* **2011**, *18*, 255–258. [CrossRef]
35. 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]
36. 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]
37. 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]

38. Pan, Z.; Wang, L.; Hu, S.; Ma, X. Reversible data hiding in encrypted image using new embedding pattern and multiple judgments. *Multimed. Tools Appl.* **2016**, *75*, 8595–8607. [CrossRef]
39. Qian, Z.; Dai, S.; Jiang, F.; Zhang, X. Improved joint reversible data hiding in encrypted images. *J. Vis. Commun. Image Represent.* **2016**, *40*, 732–738. [CrossRef]
40. 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]
41. Wu, X.; Sun, W. High-capacity reversible data hiding in encrypted images by prediction error. *Signal. Process.* **2014**, *104*, 387–400. [CrossRef]
42. 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 2017 25th European Signal Processing Conference (EUSIPCO), Kos, Greece, 28 August–2 September 2017; pp. 2186–2190.
43. Dragoi, I.C.; Coltuc, D. Reversible data hiding in encrypted images based on reserving room after encryption and multiple predictors. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2102–2105.
44. Zhang, X. Separable reversible data hiding in encrypted image. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 826–832. [CrossRef]
45. 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]
46. 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]
47. 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]
48. 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]
49. Computer Vision Group, Test Image Database. Available online: http://decsai.ugr.es/cvg/dbimagenes/g512.php (accessed on 31 July 2021).