mathematics

MDPI

Article

Separable Reversible Data Hiding in Encrypted Image Based on Two-Dimensional Permutation and Exploiting Modification Direction

Chunqiang Yu [1], Chenmei Ye [1], Xianquan Zhang [1,*], Zhenjun Tang [1,*] and Shanhua Zhan [2]

[1] Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin 541004, China; yu_chunqiang@126.com (C.Y.); mit2007@mailbox.gxnu.edu.cn (C.Y.)
[2] Department of Information and Management, Guangdong Justice Police Vocational College, Guangzhou 510520, China; shz0606@foxmail.com
* Correspondence: zxq6622@163.com or zxq@mailbox.gxnu.edu.cn (X.Z.); tangzj230@163.com or zjtang@gxnu.edu.cn (Z.T.); Tel.: +86-135-9733-6671 (X.Z.); +86-152-9599-0968 (Z.T.)

Received: 11 September 2019; Accepted: 11 October 2019; Published: 15 October 2019

Abstract: In this paper, we propose a separable reversible data hiding method in encrypted image (RDHEI) based on two-dimensional permutation and exploiting modification direction (EMD). The content owner uses two-dimensional permutation to encrypt original image through encryption key, which provides confidentiality for the original image. Then the data hider divides the encrypted image into a series of non-overlapping blocks and constructs histogram of adjacent encrypted pixel errors. Secret bits are embedded into a series of peak points of the histogram through EMD. Direct decryption, data extraction and image recovery can be performed separately by the receiver according to the availability of encryption key and data-hiding key. Different from some state-of-the-art RDHEI methods, visual quality of the directly decrypted image can be further improved by the receiver holding the encryption key. Experimental results demonstrate that the proposed method outperforms some state-of-the-art methods in embedding capacity and visual quality.

Keywords: reversible data hiding; encrypted image; two-dimensional permutation; exploiting modification direction (EMD); image recovery

1. Introduction

In the era of big data, data transmission and exchange have become easier than ever before. However, meanwhile, data security is confronted with more risks [1,2]. Image is a common form of data and also suffers the same affliction. At present, there are two effective methods to protect image data, which are image encryption and image data hiding. Image encryption converts a meaningful original image into an unrecognized one to prevent information leakage [3,4]. Different from image encryption, image data hiding modifies the pixel values of a cover image imperceptibly to embed secret data into the cover image so that the transmission of stego-image does not attract attacker's interest [5–9].

In general, the cover image will suffer distortion inevitably due to the modifying operation. However, for some image applications, e.g., military or medical images, the distortion is unacceptable even if it is very small. For the demand of recovering original image losslessly, reversible data hiding (RDH) was proposed, which can both recover the original image and extract secret data perfectly. In the literature, the existing RDH methods are generally classified into four categories: (1) The methods based on lossless compression [10,11], (2) the methods based on difference expansion (DE) [12,13], (3) the methods based on histogram shifting (HS) [14,15], and (4) the methods based on prediction-error

expansion (PEE) [16–18]. Nowadays, PEE-based methods have attracted considerable research interest due to their large embedding capacity and high fidelity.

Recently, as cloud computing and cloud storage develop rapidly, RDH in encrypted image (RDHEI) draws more and more interest [19–40]. RDHEI can be applied to the applications of ciphertext management of multimedia data, copyright protection and privacy protection. There are three users in RDHEI: the content owner, the data hider and the receiver. The content owner encrypts the original image according to encryption key and uploads it to the server. Then the data hider embeds some secret data into the encrypted image according to data-hiding key and cannot access the original image content without an encryption key. The receiver can obtain the original image, secret data or both under specific authority. Generally, there are mainly three RDHEI frameworks, namely reserving room before encryption (RRBE) [19–25], vacating room after encryption (VRAE) [26–32] and vacating room by encryption (VRBE) [33–40].

RRBE-based methods adopt RDH methods in plaintext domain or high pixel correlation to obtain embedding room before image encryption. For example, Ma et al. [19] exploited the traditional RDH method to achieve self-embedding before encryption so as to obtain room for data embedding. In Reference [20], some pixels are estimated before image encryption and secret data is embedded into the estimated errors. In Reference [21], small image blocks are compressed by sparse representation to vacate room. Shiu et al. [22] exploited DE to transform two adjacent pixels into two odd or even pixels, and then adopted Paillier encryption [41] to encrypt pixels. The homomorphism of Paillier encryption assists data extraction and image recovery. In Reference [23], the mean values are preserved before encryption. Due to mean value preservation, data extraction and image recovery can be performed perfectly. Yu et al. [24] encrypted the pre-processed image generated by prediction and conducted data hiding with additive homomorphism. Nguyen et al. [25] selected smooth pixels according to a given threshold and four neighboring pixels before encryption. The middle bit-planes of the selected smooth pixels of encrypted image are accommodated for secret data. The RRBE-based RDHEI schemes can achieve good embedding performance, but they require extra pre-processing before image encryption, which increases computational cost for the content owner.

VRAE based methods use standard Advanced Encryption Standard (AES), Rivest Cipher 4 (RC4) encryptions to encrypt the original image directly and do not require extra pre-processing before image encryption. Thus, the VRAE based methods have more widespread application than the RRBE-based methods. In References [26,27], the original image is encrypted directly by using the stream cipher and the encrypted image is divided into several blocks. Three least significant bits (LSB) of half of the pixels in a block are flipped to embed a secret bit. In the receiver side, data extraction and image recovery are conducted by designing a fluctuation function. Liao and Shu [28] adopted data hiding scheme [26] to embed a secret bit into a block. To improve the accuracy of extracted data and recovered image, the locations of different pixels of a block are considered to design an evaluation of block complexity in their method. Qin and Zhang [29] elaborated a pixel selection strategy so that data hiding is performed by flipping the LSBs of fewer pixels and the visual quality of decrypted image can be improved. In the above methods, the receiver is required to perform data extraction and image recovery simultaneously. It means that, for these VRAE based methods [26–29], if the receiver only holds data-hiding key, he or she cannot extract secret data. To separate data extraction from image recovery, Zhang [30] first proposed separable RDHEI method. In this method, the stream cipher is exploited to directly encrypt the original image and the room for accommodating secret data is vacated by compressing the LSBs of the encrypted image. Qian et al. [31] adopted distributed source coding to compress the encrypted image to improve embedding performance. Wu and Sun [32] embedded secret bits into the encrypted image by using most significant bit (MSB) replacement technique. At the receiver's side, a prediction technique is utilized to perform data extraction and image recovery.

Since pixel spatial correlation is disorganized by the stream cipher which leads to the maximum entropy of the image, it is extraordinary difficult to vacate room from the encrypted image by the stream cipher. Consequently, VRAE based methods achieve low embedding rate. To achieve high

embedding rate, some researchers have designed VRBE based methods. In these methods, some specific encryption algorithms provide confidentiality for the image while keeping spatial redundancy in the encrypted image. Room for data hiding can be vacated from the encrypted image. Because the encrypted image has spatial redundancy, some RDH methods proposed in the plain domain can be introduced into RDHEI by VRBE, which can achieve better embedding performance than BBRE and VRAE. For example, Yin et al. [33] encrypted image by changing pixel locations and embedded secret data by the HS method. This method exists security loophole in the encrypted image due to only changing pixel locations. Yi et al. [34] encrypted the original image by two stages involving block permutation and stream cipher. The data hider embeds secret bits into the permuted blocks by using PEE after performing stream decipher. Di et al. [35] divided an original image into a series of blocks and encrypted all the pixels of a block by using the same key. Every encrypted pixel is divided into two components and the HS is adopted for data hiding in these two components. Xiao et al. [36] divided original image into blocks sized 2×2 and encrypted all the pixels of each block with same key and additive homomorphism. The pixel value ordering (PVO) strategy realizes data hiding in each block. Yu et al. [37] adopted key transmission for image encryption and shifted the histogram of two-layer encrypted pixel errors to embed secret data reversibly. Tang et al. [38] divided the original image into several blocks and encrypted all the pixels in a block using the same number so that pixel spatial correlation in a block is preserved. The encrypted image is compressed to vacate room for data hiding. In References [39,40], redundant space is transferred from the original image to the encrypted image. Secret bits are embedded into vacated room by sparse matrix encoding. These two methods achieve high embedding rate.

In this paper, an RDHEI method based on two-dimensional permutation and EMD is proposed. It is a VRBE method that keeps pixel spatial correlation so that data hiding can be achieved by EMD to improve embedding capacity significantly. The rest of the paper is organized as follows. Section 2 presents our separable RDHEI method. Section 3 provides the experimental results and evaluates the performance of the proposed method. Finally, the paper is concluded in Section 4.

2. Proposed Method

In this section, an effective separable RDHEI method is presented. Figure 1 illustrates the block diagram of the proposed method. The content owner encrypts the original image by using two-dimensional permutation, which consists of bit-plane permutation and block permutation according to the encryption key. The bit-plane permutation is achieved by Arnold transformation. According to the data-hiding key, the data hider embeds secret bits into the encrypted image by using EMD to generate a marked encrypted image. After receiving the marked encrypted image, the receiver can perform different operations according to the availability of encryption key and data-hiding key. If only data hiding key is available, the receiver can only perform data extraction. If only encryption key is available, the receiver can directly perform decryption to obtain a directly decrypted image, which is analogous to a noisy image. It is noted that the quality of the directly decrypted image can be further improved by using noise removal method [42]. If both keys are available, the receiver extracts the secret bits and recovers the original image perfectly.

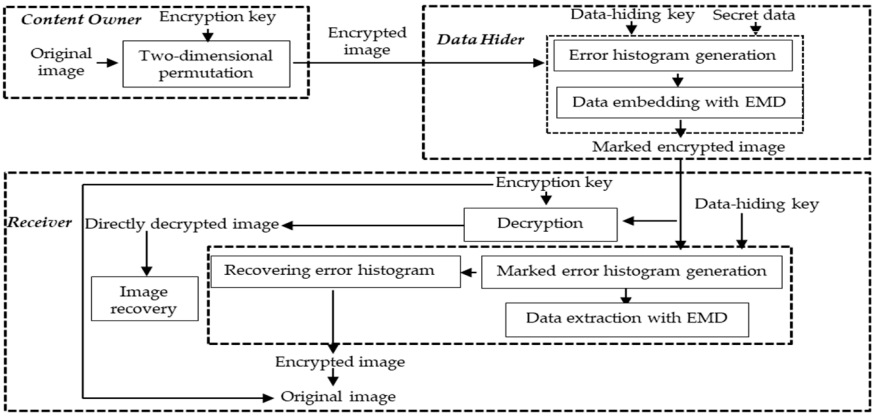


Figure 1. Block diagram of the proposed method.

2.1. Image Encryption

In this stage, the content owner encrypts an original image by using two-dimensional permutation, which consists of bit-plane permutation and block permutation. Note that bit-plane permutation and block permutation aim to scramble the original pixel values and the original pixel positions, respectively.

2.1.1. Bit-Plane Permutation

Suppose that **I** is an 8-bit original image sized $H \times W$ and $I_{i,j}$ denote one pixel value of **I**, where $0 \le I_{i,j} \le 255, 1 \le I \le H$, and $1 \le j \le W$. Since a pixel consists of 8 bits, the content owner divides $I_{i,j}$ into 4 Most Significant Bits (MSB) and 4 Least Significant Bits (LSB), and then converts the 4 MSBs and 4 LSBs into two decimal numbers denoted by $x_{i,j}$ and $y_{i,j}$, respectively. The numbers $x_{i,j}$ and $y_{i,j}$ can be obtained by

$$\begin{cases} y_{i,j} = I_{i,j} \bmod 16 \\ x_{i,j} = \left(I_{i,j} - y_{i,j}\right) \bmod 16 \end{cases} \tag{1}$$

where the "mod" denotes the modulo operation and $0 \le x_{i,j} \le 15$, $0 \le y_{i,j} \le 15$. Then Arnold transform is employed to scramble $x_{i,j}$ and $y_{i,j}$ as follows.

$$\begin{pmatrix} x'_{i,j} \\ y'_{i,j} \end{pmatrix} = \begin{pmatrix} 1\,1 \\ 1\,2 \end{pmatrix} \begin{pmatrix} x_{i,j} \\ y_{i,j} \end{pmatrix} (\bmod 16) \tag{2}$$

where $x'_{i,j}$ and $y'_{i,j}$ represent the transformed values of $x_{i,j}$ and $y_{i,j}$, respectively. Let $x^k_{i,j}$ and $y^k_{i,j}$ be transformed values of $x_{i,j}$ and $y_{i,j}$ after the Equation (2) is iterated k times. Thus, the permutated pixel value $I^k_{i,j}$ can be calculated via

$$I^k_{i,j} = 16 \times x^k_{i,j} + y^k_{i,j} \tag{3}$$

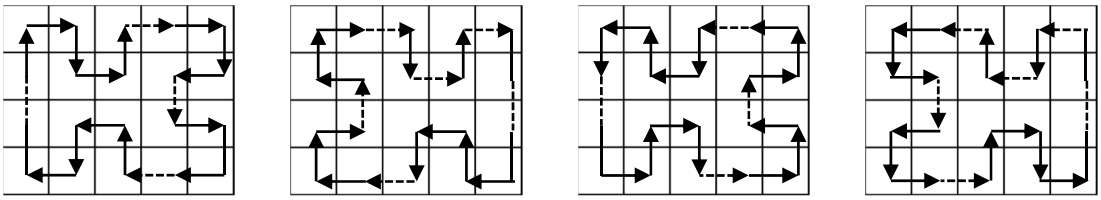
It is well-known that the Arnold transform has periodicity. Specifically, the original image will reappear after a certain number of iterations. Additionally, Arnold transform period has been studied by Reference [43] and the period of Equation (2) is 12 from Reference [43]. Since the period of Equation (2) is 12, there are 11 permutated images, which have different scrambled effects due to different iterations. To achieve the best scrambled effect for image encryption, correlation coefficient is adopted to evaluate correlation between the original image and its bit-plane permutated version with different k ($1 \le k \le 11$). It is defined as

$$\rho_k = \frac{Cov(\mathbf{X}, \mathbf{Y}_k)}{\sqrt{D(\mathbf{X})}\sqrt{D(\mathbf{Y}_k)}} \tag{4}$$

where $\mathbf{X}$ and $\mathbf{Y}_k$ denote the matrices of the original image and its bit-plane permutated version after k times iteration, respectively. $Cov(\mathbf{X}, \mathbf{Y}_k)$ represents the covariance between $\mathbf{X}$ and $\mathbf{Y}_k$. Additionally, $D(\mathrm{X})$ and $D(\mathbf{Y}_k)$ represents the variances of $\mathbf{X}$ and $\mathbf{Y}_k$, respectively. Suppose that $|\rho_n|$ is the smallest correlation coefficient among $\{|\rho_1|, |\rho_2|,.., |\rho_{11}|\}$ ($|\cdot|$ is the absolute function) and its corresponding index is n, which is the optimal iteration. Consequently, a bit-plane permutated image is generated after all the original pixels are conducted by using the Equation (2) with n iterations.

2.1.2. Block Permutation

To further enhance security of image encryption, block permutation is adopted to scramble the positions of all pixels of the bit-plane permutated image. The block permutation consists of a coarse-grained and a fine-grained permutation. The coarse-grained permutation is exploited to permute blocks within the image according to a random key K_c and the fine-grained permutation is exploited to permute pixels within each block. Specially, the bit-plane permutated image is divided into a series of non-overlapping blocks sized $t \times (t+1)$, and the number of blocks is $N = \lfloor H/t \rfloor \times \lfloor W/(t+1) \rfloor$, where $\lfloor \cdot \rfloor$ denotes the rounding down operation. Considering key space and encryption security, the value of t should not be too large. The appropriate value of t may be 4, 8 or 16, thus block sizes are 4×5, 8×9 or 16×17. For fine-grained permutation, we design four kinds of closed Hilbert orders C_1, C_2, C_3, C_4 as shown in Figure 2a–d, respectively.



(**a**) (**b**) (**c**) (**d**)

Figure 2. Four kinds of closed Hilbert orders. (**a**) Closed Hilbert order C_1. (**b**) Closed Hilbert order C_2. (**c**) Closed Hilbert order C_3. (**d**) Closed Hilbert order C_4.

Let $\mathbf{B}_i$ ($1 \le I \le N$) be a block after coarse-grained permutation and $b_{i,x,y}$be one pixel of $\mathbf{B}_i$, where x and y denote the coordinates of $b_{i,x,y}$ ($1 \le x \le t$, $1 \le y \le t+1$) in $\mathbf{B}_i$. As Figure 2 illustrated, there are four kinds of closed Hilbert orders for $\mathbf{B}_i$, thus we select a kind of closed Hilbert order for $\mathbf{B}_i$ by the logistic chaotic map which is defined as follows

$$c_{i+1} = rc_i(1\text{-}c_i),\ 0 \le r \le 4,\ c_i \in (0,1) \tag{5}$$

where r is a control parameter, and the logistic map is in the chaotic state when $3.569945972 < r < 4$. In our paper, r is set as 3.689945977 and the initial value c_0 is regarded as key. The Equation (5) is calculated calculate $2N$ times repeatedly to generated a chaotic sequence $\{c_1, c_2, \ldots, c_{2N}\}$. Since $0 < c_i < 1$, a binary sequence $\{\lfloor c_1 \rfloor, \lfloor c_2 \rfloor, \ldots, \lfloor c_{2N} \rfloor\}$ can be obtained. This binary sequence is transformed into a sequence of digits in 4-ary notational system, denoted by $\{q_1, q_2, \ldots, q_N\}$. Then we select a kind of closed Hilbert order for $\mathbf{B}_i$ from Figure 2 according to q_i, where $0 \le q_i \le 3$ and $1 \le I \le N$. Note that if $q_i = 0$, C_1 is selected for $\mathbf{B}_i$. If $q_i = 1$, C_2 is selected for $\mathbf{B}_i$. If $q_i = 2$, C_3 is selected for $\mathbf{B}_i$. If $q_i = 3$, C_4 is selected for $\mathbf{B}_i$. After selecting a closed Hilbert order, a step size f_i is generated for $\mathbf{B}_i$ according to a random key K_f, where $0 \le f_i \le t \times (t+1)$. Then all the pixels of $\mathbf{B}_i$ are put forward f_i step beginning with $b_{i,1,1}$ according to the selected closed Hilbert order, which realizes fine-grained permutation. Suppose that the image blocks after block permutation are $\left\{\mathbf{B}'_1, \mathbf{B}'_2, \ldots, \mathbf{B}'_N\right\}$. Consequently, the final encrypted image $\mathbf{I}_\mathrm{e}$ can be constructed by $\left\{\mathbf{B}'_1, \mathbf{B}'_2, \ldots, \mathbf{B}'_N\right\}$. The encryption keys consist of iteration number n, K_c, c_0 and K_f. Figure 3 shows the final encrypted version with two-dimensional permutation.

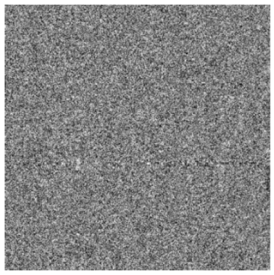


Figure 3. Final encrypted image.

2.2. Data Hiding in Encrypted Image

The data hider embeds secret bits into $\mathbf{I}_e$ according to pixel spatial correlation within blocks of the encrypted image. Firstly, $\mathbf{I}_e$ is divided into N non-overlapping blocks sized $t \times (t+1)$ by the data hider, denoted by $\left\{\mathbf{B}'_1, \mathbf{B}'_2, \ldots, \mathbf{B}'_N\right\}$. Note that bit-plane permutation and block permutation are able to preserve spatial correlation and relative position between two adjacent original pixels with the block, the data hider can calculate adjacent encrypted pixel errors to construct an error histogram which is used for data hiding with EMD. Similar with most reversible data hiding, some auxiliary information is also generated in the proposed method, which is described in the Section 2.2.1 in detail. The data hider divides $\left\{\mathbf{B}'_1, \mathbf{B}'_2, \ldots, \mathbf{B}'_N\right\}$ into two parts $\left\{\mathbf{B}'_1, \mathbf{B}'_2, \ldots, \mathbf{B}'_r\right\}$ and $\left\{\mathbf{B}'_{r+1}, \mathbf{B}'_2, \ldots, \mathbf{B}'_N\right\}$. The LSBs of $\left\{\mathbf{B}'_1, \mathbf{B}'_2, \ldots, \mathbf{B}'_r\right\}$ is extracted to obtain a binary sequence S_{LSB} and replaced by the auxiliary information. Then S_{LSB} and secret bits are concatenated to form the embedded data to be embedded into $\left\{\mathbf{B}'_{r+1}, \mathbf{B}'_2, \ldots, \mathbf{B}'_N\right\}$.

2.2.1. Error Histogram Generation

In this part, an error histogram is generated by calculating adjacent encrypted pixel errors of each block. We take $\mathbf{B}'_i$ ($r+1 \le I \le N$) as an example to describe the error histogram generation. The data hider randomly selects a pixel $b'_{i,x,y}$ from $\mathbf{B}'_i$ according to a random key K_h. Then the data hider scans $\mathbf{B}'_i$ beginning with $b'_{i,x,y}$ in a selected closed Hilbert order determined by the sharing key c_0 to generate a pixel sequence denoted by P_i={$p_{i,1}$, $p_{i,2}$, … , $p_{i,m}$}, $m = t \times (t+1)$. Then the adjacent pixel errors of P_i are calculated by.

$$e_{i,j} = \begin{cases} p_{i,j}, & j = 1 \\ p_{i,j-1} - p_{i,j}, & 2 \le j \le m \end{cases} \tag{6}$$

where $-255 \le e_{i,j} \le 255$. By the same way, the adjacent pixel errors of $\left\{\mathbf{B}'_{r+1}, \mathbf{B}'_2, \ldots, \mathbf{B}'_N\right\}$ are calculated and are concatenated to generate an error sequence denoted by $E = \{e_1, e_2, \ldots, e_d\}$, where $d = m \times (N-1)$. Note that $e_{i,1}$ ($r+1 \le I \le N$) is excluded for concatenating and data hiding. Consequently, the histogram of E is generated. Figure 4 shows the error histogram of the encrypted Lena.

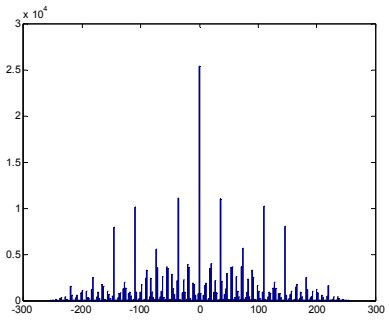
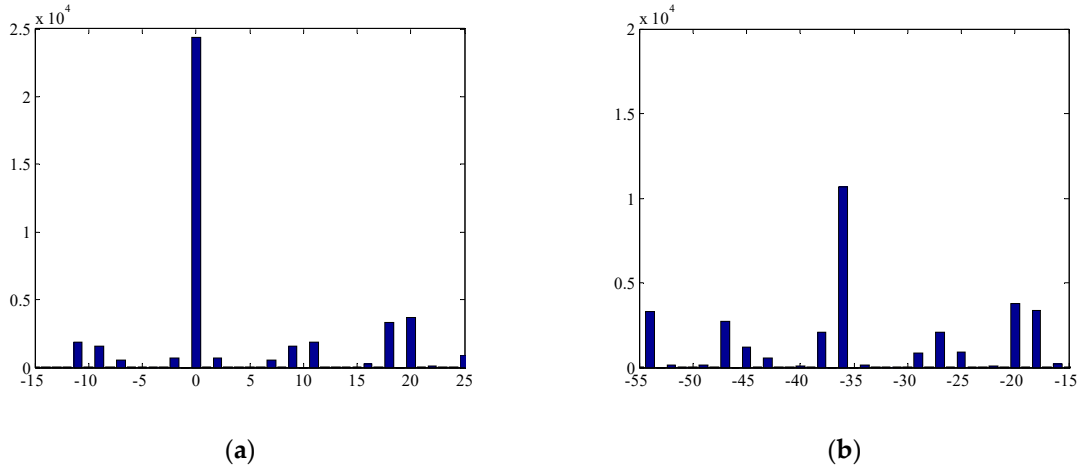


Figure 4. Error histogram of the encrypted Lena.

It can be observed that there are a wide variety of peak bins and zero bins in the histogram from Figure 4. Assume that the histogram bins and their numbers are denoted by $\{b_{-255}, \ldots, b_0, \ldots, b_{255}\}$ and $\{n_{-255}, \ldots, n_0, \ldots, n_{255}\}$, respectively. For convenient observation, we magnify some parts of Figure 4 to exhibit the bins and their numbers as shown in Figure 5. It is observed from Figure 5a that b_0 is a peak bin, while b_{-1} and b_1 are two zeros bin adjacent with bin b_0. The b_{-2} and b_2 are two peak bins, while b_{-3} and b_3 are two zero bins which are adjacent with bin b_{-2} and b_2, respectively. Similarly, it is observed from Figure 5b that b_{-36} is a peak bin, while b_{-37} and b_{-35} are two zeros bin adjacent with bin b_{-36}. The b_{-38} is a peak bin, while b_{-39} is a zero bin which is adjacent with bin b_{-38}. The other bins of Figure 4 have same properties. It is concluded that there is one zero bin between two peak bins at least and there is one zero bin adjacent with one peak bin in the error histogram in most cases. Peak bins and zero bins occur alternatively. Consequently, these peak bins are accommodated for data hiding. According to the property of the histogram, secret bits can be embedded into the encrypted image via EMD without histogram shifting.



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

Figure 5. Magnified versions of some parts of Figure 4. (**a**) Bins around b_0. (**b**) Bins around b_{-36}.

2.2.2. Data Hiding with EMD

For clarity, we construct a histogram—as shown in Figure 6—the property of which is analogous with that of Figure 5 to illustrate our data hiding. In this histogram, the bins b_2, b_4, b_6, b_8 are four peak bins which are expanded for data hiding, while the bins b_1, b_3, b_5, b_7, b_9 are five zero bins which are vacated room for reversible data hiding. Since every peak bin can be expanded rightward and leftward, there are a series of expanding schemes for these peak bins. Due to page limitation, we only give two expanding schemes E_1 and E_2 as shown in Figures 6a and 6b, respectively. In Figure 6a, the peak bins b_4 and b_8 can be both expanded rightward and leftward for data hiding and the bin b_2 can be only expanded leftward. The peak bin b_6 is unchanged to avoid ambiguity. In Figure 6b, the peak bin b_8 can be both expanded rightward and leftward for data hiding and the peak bins b_2, b_4, b_6 are only expanded leftward for data hiding. No matter which scheme, the zero bins are marked by a location map for avoiding ambiguity.

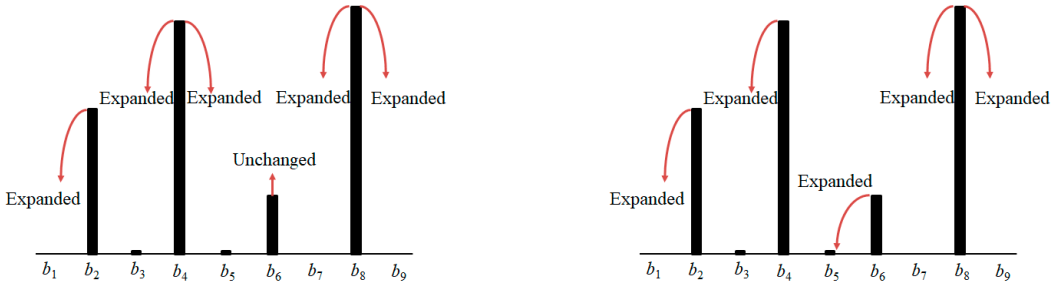


Figure 6. Two Expanding schemes. (**a**) Expanding scheme: E1. (**b**) Expanding scheme: E2.

Based on the above analysis, every peak bin has two directions or only one direction to be expanded. Consequently, secret bits can be embedded into the peak bins using EMD which uses directional modification of data hiding. In the EMD method, a group of n pixels has different possible directions to embed secret bits according to designed extraction function. Then, data hiding can be performed as follows.

For $E = \{e_1, e_2, \ldots, e_d\}$, let the bin b_k be a candidate peak bin for data hiding. We extract all the errors the values of which are equal with b_k from E to embed secret bits. All these errors are permuted into a series of error-pairs according to a random K_d. Suppose that e_i and e_j are a pair of errors and their marked errors are e_i' and e_j', respectively. Since there are three expanding cases for embedding bin b_k which are only expanding leftward, only expanding rightward and expanding both leftward and rightward, there are three corresponding embedding methods as follows.

1. Only expanding leftward

If b_k is only expanded leftward, it means the errors whose values are equal with b_k can be decreased by 1 or unchanged during data hiding. For a pair (e_i, e_j), $e_i = e_j = b_k$, we embed two secret bits into (e_i, e_j). Two secret bits are transformed into a secret digit s $(0 \le s \le 3)$. In this case, the extraction function f_1 is defined as

$$f_1(a,b)=(a+2\times b) \bmod 4 \tag{7}$$

If e_i' and e_j' satisfy the conditions $s=f_1(e_i', e_j')$ and $\begin{cases} b_k - 1 \le e_i' \le b_k \\ b_k - 1 \le e_j' \le b_k \end{cases}$, s can be embedded by $f_1(a,b)$. Consequently, (e_i', e_j') can be obtained as follows.

① If $s = f_1(e_i', e_j')$, $e_i' = b_k$ and $e_j' = b_k$; ② If $s = f_1(e_i'-1, e_j')$, $e_i' = b_k-1$ and $e_j' = b_k$;
③ If $s = f_1(e_i', e_j'-1)$, $e_i' = b_k$ and $e_j' = b_k-1$; ④ If $s = f_1(e_i'-1, e_j'-1)$, $e_i' = b_k-1$ and $e_j' = b_k-1$;

By the same way, other secret bits can be embedded into other pairs.

2. Only expanding rightward

If b_k is only expanded rightward, it means the errors whose values are equal with b_k can be increased by 1 or unchanged during data hiding. For a pair of errors (e_i, e_j), $e_i = e_j = b_k$, we embed two secret bits into (e_i, e_j). Two secret bits are transformed into a secret digit $s(0 \le s \le 3)$. In this case, Equation (7) is exploited as the extraction function. If e_i' and e_j' satisfy the conditions $s = f_1(e_i', e_j')$ and $\begin{cases} b_k \le e_i' \le b_k + 1 \\ b_k \le e_j' \le b_k + 1 \end{cases}$, s can be embedded by $f_1(a,b)$. Consequently, (e_i', e_j') can be obtained as follows.

① If $s = f_1(e_i', e_j')$, $e_i' = b_k$ and $e_j' = b_k$; ② If $s = f_1(e_i'+1, e_j')$, $e_i' = b_k+1$ and $e_j' = b_k$;
③ If $s = f_1(e_i', e_j'+1)$, $e_i' = b_k$ and $e_j' = b_k+1$; ④ If $s = f_1(e_i'+1, e_j'+1)$, $e_i' = b_k+1$ and $e_j' = b_k+1$;

By the same way, other secret bits can be embedded into other pairs.

3. Expanding both leftward and rightward

If b_k can be both expanded leftward and rightward, it means the errors whose values are equal with b_k can be increased by 1, decreased by 1 or unchanged during data hiding. For a pair (e_i, e_j),

e_i=e_j=b_k, we embed three secret bits into (e_i, e_j). Three secret bits are transformed into a secret digit s ($0 \le s \le 7$). According to the algorithm [7], the extraction function f_2 is defined as

$$f_2(a,b)=(a+3\times b) \bmod 8 \tag{8}$$

If e_i' and e_j' satisfy the conditions $s=f_2(e_i', e_j')$ and $\begin{cases} b_k - 1 \le e_i' \le b_k + 1 \\ b_k - 1 \le e_j' \le b_k + 1 \end{cases}$, s can be embedded by $f_2(a,b)$. Consequently, (e_i',e_j') can be obtained as follows.

① If $s=f_2(e_i', e_j')$, $e_i'=b_k$ and $e_j'=b_k$; ② If $s=f_2(e_i'+1, e_j')$, $e_i'=b_k+1$ and $e_j'=b_k$;
③ If $s=f_2(e_i'-1, e_j')$, $e_i'=b_k-1$ and $e_j'=b_k$; ④ If $s=f_2(e_i', e_j'+1)$, $e_i'=b_k$ and $e_j'=b_k+1$;
⑤ If $s=f_1(e_i', e_j'-1)$, $e_i'=b_k$ and $e_j'=b_k-1$; ⑥ If $s=f_1(e_i'+1, e_j'+1)$, $e_i'=b_k+1$ and $e_j'=b_k+1$;
⑦ If $s=f_1(e_i'+1, e_j'-1)$, $e_i'=b_k+1$ and $e_j'=b_k-1$; ⑧ If $s=f_1(e_i'-1, e_j'+1)$, $e_i'=b_k-1$ and $e_j'=b_k+1$;
By the same way, secret data can be embedded into other pairs.

Overall, if b_k is only expanded leftward or rightward, two secret bits can be embedded into a pair of errors. Otherwise, three secret bits can be embedded into a pair of errors. Consequently, we prefer to select the peak peaks with more numbers to be both expanded leftward and rightward for obtaining more embedding capacity as follows. The bin numbers $\{n_{-255}, \ldots, n_0, \ldots, n_{255}\}$ are sorted in stable descending order to obtain $\{n_{\sigma(1)}, n_{\sigma(2)}, \ldots, n_{\sigma(511)}\}$ and their corresponding bins are $\{b_{\sigma(1)}, b_{\sigma(2)}, \ldots, b_{\sigma(511)}\}$, where $\sigma(i) \in \{-255,-254, \ldots, 254,255\}$ and σ:$\{1,2, \ldots, 511\} \to \{1,2, \ldots, 511\}$ is the unique one-to-one mapping such that: $n_{\sigma(1)} \ge n_{\sigma(2)} \ge \ldots \ge n_{\sigma(511)}$. In order to achieve high payload, the bins with more numbers among $\{b_{\sigma(1)}, b_{\sigma(2)}, \ldots, b_{\sigma(511)}\}$ are selected to be both expanded leftward and rightward in priority. Let the embedding capacity be l_{EC}. First an empty set S is initiated and the expanded bins are stored in S which can be obtained by the following details. A counter c is used for calculating the payload and its initial value is 0. Beginning with $i = 1$, it is clear that the bin the bin $b_{\sigma(1)}$ can be both expanded leftward and rightward and $b_{\sigma(1)}$ is appended with S. $c = c+3\times n_{\sigma(1)}/2$ and $i = i+1$. If $i > 1$, the expanding rules of $b_{\sigma(i)}$ are determined by $| b_{\sigma(i)} - b_{\sigma(j)} |$ and can be classified into three cases, where $j = 1,2, \ldots i$-1.

1) If $(\forall j)| b_{\sigma(i)}-b_{\sigma(j)} | > 2$, the bin $b_{\sigma(i)}$ can be both expanded leftward and rightward and $b_{\sigma(i)}$ is appended with S. Additionally, $c = c +3\times n_{\sigma(i)}/2$.

2) If $(\exists j) | b_{\sigma(i)}-b_{\sigma(j)} | = 2$ and $(\forall k) | b_{\sigma(i)}-b_{\sigma(k)} | > 2$, where $k = 1,2, \ldots i$ and $k \ne j$. The bin $b_{\sigma(i)}$ can be expanded leftward or rightward. If $b_{\sigma(i)} > b_{\sigma(j)}$, the bin $b_{\sigma(i)}$ can be expanded rightward. Otherwise, the bin $b_{\sigma(i)}$ can be expanded leftward. $b_{\sigma(i)}$ is appended with S and $c = c+n_{\sigma(i)}$.

3) If $b_{\sigma(i)}$ does not satisfy the above condition, $b_{\sigma(i)}$ is skipped.

Repeat the above process until $c \ge l_{EC}$. We take Figure 6a as an example to illustrate the generation of S. It is clear that $\{n_8, n_4, n_2, n_6\}$ and $\{n_3, n_5, n_1, n_7, n_9\}$ are the peak bins and zero bins, respectively. The sorted bin numbers are $\{n_8, n_4, n_2, n_6, n_3, n_5, n_1, n_7, n_9\}$ and the corresponding bins are $\{b_8, b_4, b_2, b_6, b_3, b_5, b_1, b_7, b_9\}$. According to the above expanding rules, the first peak bin b_8 can be both expanded leftward and rightward and b_8 is appended with S. For the second peak bin b_4, it is clear that $b_8 - b_4 > 2$, thus b_4 can be both expanded leftward and rightward and b_4 is appended with S. For the third peak bin b_2, there is a bin b_4 which satisfies $b_4 - b_2 = 2$. Thus, b_2 can be only expanded leftward and b_2 is appended with S. For the fourth bin b_6, there are b_4 and b_8 which satisfy the condition $b_8 - b_6 = 2$ and $b_4 - b_6 = -2$. Thus, b_6 remains unchanged. Consequently, S consists of b_8, b_4, b_2. After obtaining S, secret bits are embedded into the errors of E by expanding the bins from the set S in order according to the embedding method and suppose that the marked error sequence is $E'=\{e_1', e_2', \ldots, e_d'\}$. Then these marked errors can be mapped to the marked errors in each block. Consequently, the marked pixels in $\left\{\mathbf{B}_{r+1}'', \mathbf{B}_2'', \ldots, \mathbf{B}_N''\right\}$ are generated by

$$p_{i,j}' = \begin{cases} p_{i,j}, & j = 1 \\ p_{i,j-1} - e_{i,j}', & 2 \le j \le m \end{cases} \tag{9}$$

where r+1 ≤ i ≤ N. In order to blind extraction, some auxiliary information is required to be embedded into the encrypted and it is described as follows. The overflow/underflow problem may occur since the pixel value 0 or 255 may be modified to −1 or 256, respectively. To address this problem, we utilize a location map **LM** with the same size of the encrypted image to mark those pixels with value 0 or 255. For each encrypted pixel, if its value is 0 or 255, it is marked with 1 in **LM** and skipped for data hiding. Otherwise, it is marked with 0 in **LM**. Meanwhile, the pixels with the zero bins are also marked with 1 in **LM**. To reduce the size of **LM, LM** is losslessly compressed and its compressed version is denoted as L_{clm} and the size of L_{clm} is l_{clm}. Since the peak bins vary from −255 to 255, each peak bin can be represented 9 bits and the first bit is a sign bit. The bins are almost evenly divided into the peak bins and zero bins as shown in Figure 6, thus the maximum size L_s of S is 9 × 255. Since most pixels of encrypted image are used for data hiding and three bits are embedded into two pixels, $3/2 \times \log_2 HW$ bits are enough to represent the embedding capacity l_{EC}. In summary, the auxiliary information consists of three sections:

* The compressed location map L_{clm} (l_{clm} bits)
* The expanded bin set S (L_s bits)
* The embedding capacity l_{EC} ($3/2 \times \log_2 HW$ bits)

The auxiliary information is embedded into the LSBs of $\left\{\mathbf{B}'_1, \mathbf{B}'_2, \ldots, \mathbf{B}'_r\right\}$ by using LSB replacement technique to generate the marked blocks $\left\{\mathbf{B}''_1, \mathbf{B}''_2, \ldots, \mathbf{B}''_r\right\}$. Finally, a marked encrypted image $\mathbf{I}_{\mathrm{w}}$ can be obtained by $\left\{\mathbf{B}''_1, \mathbf{B}''_2, \ldots, \mathbf{B}''_N\right\}$.

2.3. Data Extraction and Image Recovery

With a marked encrypted image $\mathbf{I}_{\mathrm{w}}$, the receiver can perform different operations involving data extraction, image decryption and image recovery according to the availability of the encryption key and the data-hiding key.

If only encryption key is available for the receiver, he can directly decrypt $\mathbf{I}_{\mathrm{w}}$ to obtain a noise-like image which can be filtered to generate an image with remarkable quality. Specifically, the receiver divides $\mathbf{I}_{\mathrm{w}}$ into a series of non-overlapping blocks with $t \times (t+1)$ size. Since each original pixel is encrypted by iterating n times using Equation 2, the marked encrypted pixels can be decrypted by iterating 12-n times using Equation 2 to generate the directly decrypted pixels. After directly decrypting pixels, inverse coarse-grained permutation and fine-grained permutation are performed to recover the positions of those pixels. Consequently, a directly decrypted image is generated. Note that the encrypted pixels may increase by 1, decrease by 1 or keep unchanged during data hiding. Those marked encrypted pixels which keep unchanged can be decrypted to obtain corresponding original pixels without errors. While, the decrypted pixels by decrypting those marked, encrypted pixels directly which increase by 1 or decrease by 1 may be far away from the corresponding original pixels. We take an original pixel value 95 as an example for illustrating decryption. The original pixel value 95 is iterated 7 times using Arnold transform to generate the encrypted pixel value 75. Note that 75 may be modified as 74, 75 or 76 after data embedding. In direct decryption process, the marked pixel value 74, 75 or 76 is decrypted by iterating 5 times using Equation 2 to obtain 230, 95 or 200, respectively. It is clear that 95 is a desired decryption value, while 230 and 200 can be regarded as the noises which will bring large distortion for image. That is to say, our directly decrypted image may contain noises. Let the numbers of expanded bins with one direction in S be $\left\{n_1^1, n_2^1, \ldots, n_u^1\right\}$ and the numbers of expanded bins with two directions be $\left\{n_1^2, n_2^2, \ldots, n_v^2\right\}$. The number of noises in the directly decrypted image is estimated as

$$Q \approx \frac{1}{2}\sum_{i=1}^{u} n_i^1 + \frac{5}{8}\sum_{i=1}^{v} n_i^2 \tag{10}$$

where $\frac{1}{2}\sum_{i=1}^{u} n_i^1$ and $\frac{5}{8}\sum_{i=1}^{v} n_i^2$ are the noise estimations of expanded bins with one direction and two directions respectively according to our data hiding method. Our noise mode can be regarded as uniform impulse noise and can be detected with high detection rate [44]. For a directly decrypted image, different from the previous methods, we adopt an outstanding filter method [42] to improve visual quality of the directly decrypted image significantly.

If only data-hiding key is available for the receiver, he can extract the error-free secret bits from $\mathbf{I}_w$ and recover the encrypted image losslessly. The receiver also divides $\mathbf{I}_w$ into a series of non-overlapping blocks $\left\{\mathbf{B}_1'', \mathbf{B}_2'', \ldots, \mathbf{B}_N''\right\}$ and retrieves the auxiliary information from the LSBs of the blocks $\left\{\mathbf{B}_1'', \mathbf{B}_2'', \ldots, \mathbf{B}_r''\right\}$ to obtain the compressed location map L_{clm}, the expanded bin set S and the embedding capacity l_{EC}. L_{clm} is decompressed to generate **LM**. We also take $\mathbf{B}_i''$ (r+1 ≤ i≤N) as an example to describe the marked error generation and data extraction. The receiver scans $\mathbf{B}_i''$ beginning with $b_{i,x,y}''$ in a selected closed Hilbert order determining by the sharing key c_0 to generate a pixel sequence denoted by $P_i' = \left\{p_{i,1}', p_{i,2}', \ldots, p_{i,m}'\right\}$. If $p_{i,j}'$ is marked with 1 in **LM**, $p_{i,j}'$ is skipped for data extraction and the original pixel can be recovered as $p_{i,j} = p_{i,j}'$. Otherwise $p_{i,j}'$ is used for data extraction. The marked errors in $\left\{\mathbf{B}_{r+1}'', \mathbf{B}_{r+2}'', \ldots, \mathbf{B}_N''\right\}$ are generated by

$$e_{i,j}' = \begin{cases} p_{i,j}, & j = 1 \\ p_{i,j-1} - p_{i,j}', & 2 \le j \le m \end{cases} \tag{11}$$

Then the expanded bins are selected from S to assist data extraction. The expanding method for the bin $S_{(k)}$ can be decided by the expanding rules, where $1 \le k \le u$+v. If the bin $S_{(k)}$ can be only expanded leftward or expanded rightward and $e_{i,j}' = \left\{S_{(k)}, S_{(k)} - 1\right\}$ or $e_{i,j}' = \left\{S_{(k)}, S_{(k)} + 1\right\}$, the original error can be recovered as $e_{i,j} = S_{(k)}$. If the bin $S_{(k)}$ can be both expanded leftward and expanded rightward and $e_{i,j}' = \left\{S_{(k)}, S_{(k)} - 1, S_{(k)} + 1\right\}$, the original error can be recovered as $e_{i,j} = S_{(k)}$. Then the original encrypted pixel can be recovered as

$$p_{i,j} = p_{i,j\text{-}1} - e_{i,j} \tag{12}$$

Thus, the encrypted pixels in $\left\{\mathbf{B}_{r+1}', \mathbf{B}_{r+2}', \ldots, \mathbf{B}_N'\right\}$ can be recovered losslessly. The marked error sequence $E' = \{e_1', e_2', \ldots, e_d'\}$ for the $\left\{\mathbf{B}_{r+1}', \mathbf{B}_{r+2}', \ldots, \mathbf{B}_N'\right\}$ can be obtained using Equations (11–12). For a pair of extracted marked errors in this sequence, if the corresponding bin can be only expanded leftward or expanded rightward, a secret digit can be extracted by Equation (7). Otherwise, a secret digit can be extracted by Equation (8). After data extraction, the embedded data is split into S_{LSB} and the secret bits. The LSBs of $\left\{\mathbf{B}_1'', \mathbf{B}_2'', \ldots, \mathbf{B}_r''\right\}$ is replaced by S_{LSB} to recover $\left\{\mathbf{B}_1', \mathbf{B}_2', \ldots, \mathbf{B}_r'\right\}$. Consequently, the encrypted image can be recovered by $\left\{\mathbf{B}_1', \mathbf{B}_2', \ldots, \mathbf{B}_N'\right\}$.

If the data-hiding key and the encryption key are both available for the receiver, he or she extracts secret data from the marked encrypted image and recovers the encrypted image losslessly according to the data-hiding key. Then, with the encryption key, he or she is able to decrypt the encrypted image to obtain the original image.

3. Experimental Results

In this section, six 512 × 512 standard images as shown in Figure 7 are used to validate our encryption and embedding performances. The encryption performance is evaluated by key space and the statistical security. Additionally, the superiority of the directly decrypted image quality and embedding capacity are illustrated. Section 3.1 analyzes our image encryption performance and we set t=8 for illustrating it. Section 3.2 provides comparison results with other methods to evaluate our embedding performance.

(**a**) Lena (**b**) Boats (**c**) Peppers

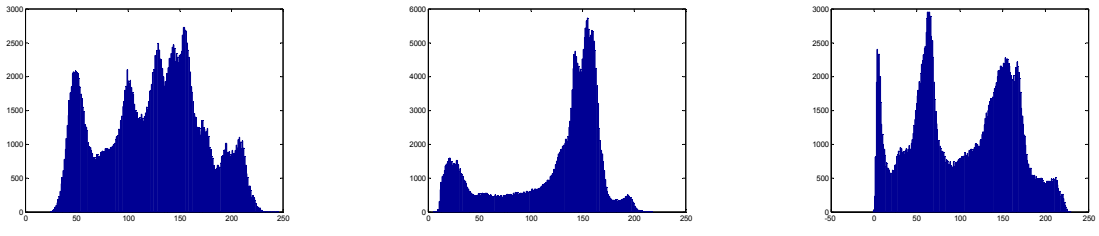
(**d**) Airplane (**e**) Airplane (**f**) Airplane

Figure 7. Six test images.

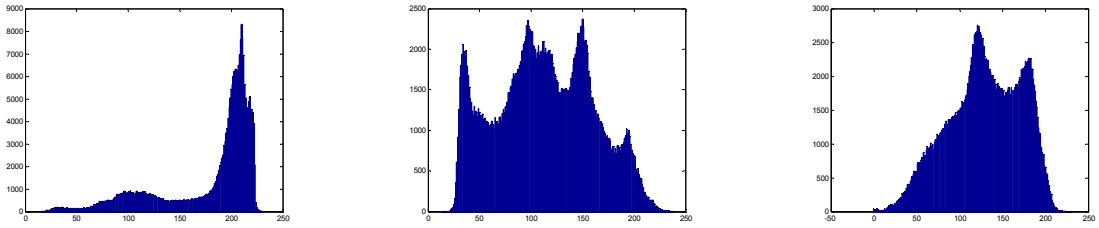
3.1. Encryption Performance

The encryption security is decided by the size of key space. Generally, the larger the size of key space is, the more security the algorithm has. Our encryption key consists of iteration time n, K_p, c_0 and K_f, where K_p and K_f are two random keys and their precisions are both 64. It is obvious that the key space of iteration time is 11 in our paper. In coarse-grained permutation, the image blocks are permuted by the key K_p with 64 bits storage and the number of possible scenarios for block permutation is $N!$. If the block number $N > 21$, $N! > 2^{64}$ and the key space is 2^{64}. Otherwise, the key space is $N!$. Thus, the valid key space is min $(2^{64}, N!)$ for coarse-grained permutation. Since c_0 is a floating number with 64 bits storage, the key space of c_0 is 2^{64}. In fine-grained permutation, there are four closed Hilbert orders for a block and the number of possible scenarios of closed Hilbert orders is 4^N, thus the valid key space is min $(2^{64}, 2^N)$. Furthermore, all the pixels of one block are put forward f_i step decided by K_f, the number of selected step sizes is $(t \times (t+1))^N$, the valid key space is min $(2^{64}, (t \times (t+1))^N)$. Overall, the whole key space of the proposed encryption method is $11 \times (2^{64}, N!) \times \min(2^{64}, 4^N) \times \min(2^{64}, (t \times (t+1))^N)$. For Lena in Figure 2, if $t = 8$, the number of blocks is $N = 3584$, the key space is $11 \times 2^{64} \times 2^{64} \times 2^{64} = 11 \times 2^{192}$, which is able to resist brute-force attacks [4].

Histogram analysis for our encryption method is presented. An image histogram illustrates the distribution of the pixel values. The histograms of the original images and their encrypted versions are exhibited in Figures 8 and 9, respectively. It is observed that the histogram of each encrypted version is entirely different from the histogram of corresponding original image. Consequently, our encryption method can withstand statistical attacks.

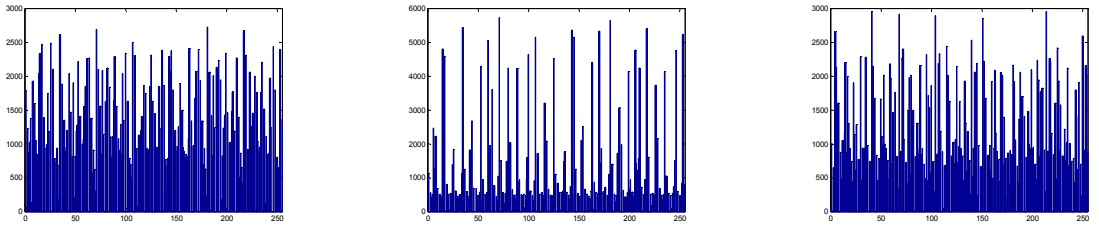


(**a**) Lena (**b**) Boats (**c**) Peppers

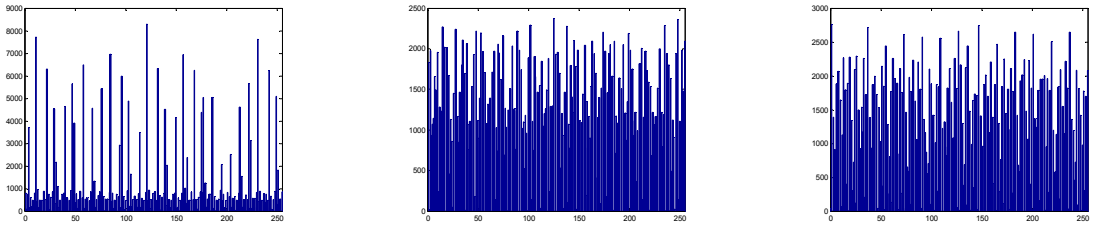


(**d**) Airplane (**e**) Barbara (**f**) Baboon

Figure 8. Histograms of original images.



(**a**) Lena (**b**) Boats (**c**) Peppers



(**d**) Airplane (**e**) Barbara (**f**) Baboon

Figure 9. Histograms of encrypted images.

We use correlation coefficient and peak signal-to-noise ratio (PSNR) between the original image and the encrypted image to evaluate the quality of the encrypted image. Correlation coefficient is defined as Equation (4) and PNSR is defined as follows.

$$\text{PSNR} = 10 \times log_{10}\left(\frac{255^2}{\text{MSE}}\right) \tag{13}$$

Where MSE represents the mean square error between the original image and its encrypted version. The evaluated results are shown in Table 1. Low PSNR demonstrates that no content of the original image can be retrieved from its encrypted version. The correlation coefficients are all closed to 0, which clearly demonstrate that the encrypted images are rarely correlated with the original images

and no information leakage from the encrypted images in case of statistical attacks. Overall, these analyses demonstrate that the proposed encryption securely protects the content of the original image.

Table 1. Correlation coefficient, PSNR values between original and encrypted images using proposed scheme.

Image	PSNR	Correlation Coefficient
Lena	9.2405	−0.0039835
Boats	9.1853	−0.0068682
Peppers	8.4256	−0.003959
Airplane	7.9554	−0.000215
Barbara	9.1159	0.00063109
Baboon	9.5364	0.0013209

3.2. Embedding Performance Comparisons

The embedding performance of the proposed method is demonstrated by comparing it with some state of art RDHEI methods [19,25,30,34–37], where Ma's method [19], Nguyen's method [25] are based on RRBE framework and Zhang's method [30] is based on VRAE framework. Yi's method [34], Di's method [35], Xiao's method [36] and Yu's method [37] are based on VRBE framework. The compared methods [19,25] use standard stream cipher to encrypt image after preprocess and the compared method [30] exploits standard stream cipher to directly encrypt image. The compared method [36] adopts standard homomorphic encryption to encrypt image. The other compared methods [34,35,37] designed specific encryption to encrypted image. Comparisons are performed from two aspects including the highest embedding capacity and the visual quality of directly decrypted image. If only the encryption key is available for the receiver, our method is able to improve the visual quality of directly decrypted image by filtering, while the other methods cannot. In aspect of comparison for the visual quality of directly decrypted image, we take our improved directly decrypted image for comparison.

In our paper, the block sizes have three cases, 4×5, 8×9, 16×17, Table 2 shows the pure embedding capacity of each image except the auxiliary information under different block sizes. It is observed that when block size is 16×17, the pure highest embedding capacity for each image can be obtained. This is because that one pixel of each encrypted block is excluded for data hiding. The smaller the block size is, the more pixels are excluded for data hiding. Consequently, the optimal size is 16×17 for embedding capacity. Moreover, the pure embedding capacity of an image is decided by the image content. The images with fewer textures (Lena, Boats et al.) will generate more peak bins and less zero bins which are recorded by the location map. The fewer the auxiliary information is, the higher the pure embedding capacity will be. The images with more textures (Baboon et al.) will generate more zero bins and more auxiliary information decreases the pure embedding capacity. Thus, the images with fewer textures achieve higher pure embedding capacities than the images with more textures (Baboon).

To demonstrate the embedding capacity advantage of the proposed method, we compare our highest embedding capacity with those of six methods and comparison results are shown in Table 3. It demonstrates that the embedding capacity of the proposed method is highest and the embedding capacity of Zhang [30] is lowest among the compared methods. In Reference [30], an original image is encrypted directly by standard stream cipher which disorganizes spatial pixel correlation. Thus, the method [30] achieves lowest embedding capacity. For all test images excluding Baboon, the highest pure embedding capacity of the proposed method exceeds 1 bpp. For example, for Lena, the maximum EC is 1.2711 bpp. However, the highest embedding capacities of the other compared methods are all less than 1 bpp. The embedding capacity of Ma [19] is similar with that of Yi [34]. The methods of Di [35] and Xiao [36] are both based on block encryption and all pixels in a block are encrypted with same key. In Xiao's method [36], PVO is adopted to calculate prediction errors and secret bits can be only embedded into the maximum value and the minimum value of a block. Thus, their embedding

capacity is limited. Di et al. calculated the pixel errors between one pixel and other pixels in a block, secret bits can be embedded by prediction-error histogram shifting (PEHS). Since most pixels of one block can accommodate secret bits, they achieve more EC than Xiao et al.

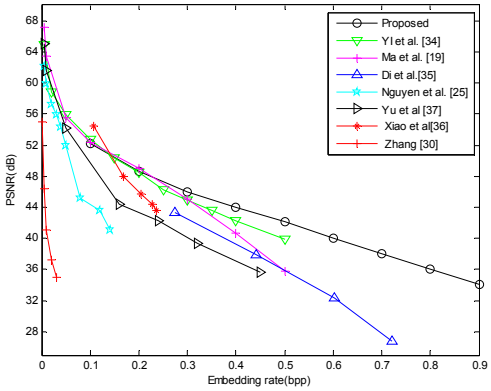
Table 2. Pure embedding capacity under different block sizes (bpp).

Image	4×5	8×9	16×17
Lena	1.2078	1.2406	1.2711
Boats	1.2032	1.2359	1.2625
Peppers	1.1754	1.2097	1.2397
Airplane	1.185	1.2214	1.253
Barbara	0.9723	1.009	1.0407
Baboon	0.7685	0.7534	0.8149

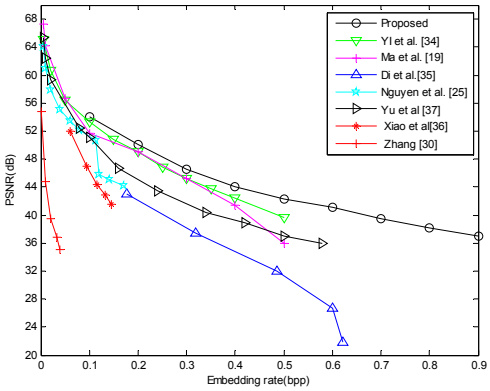
Table 3. Highest embedding capacity comparisons among different methods (bpp).

Methods	Ma [19]	Nguyen [25]	Zhang [30]	Yi [34]	Di [35]	Xiao [36]	Yu [37]	Proposed
Lena	0.5	0.1570	0.033	0.5	0.8412	0.2368	0.5343	1.2711
Boats	0.5	0.1040	0.041	0.5	0.7515	0.1451	0.5875	1.2625
Peppers	0.5	0.1580	0.022	0.4	0.7250	0.2125	0.4587	1.2397
Airplane	0.4	0.2790	0.040	0.5	1.0349	0.2649	0.6914	1.2530
Barbara	0.5	0.1310	0.020	0.5	0.7268	0.1362	0.4990	1.0407
Baboon	0.5	0.0340	0.010	0.4	0.651	0.1012	0.2314	0.8149

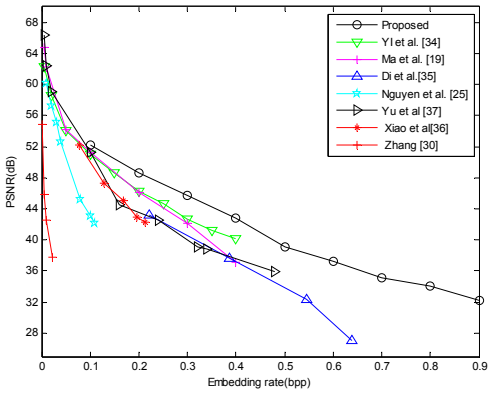
From the aspect of comparison for directly decrypted image, the capacity-distortion curves are adopted for embedding performance comparison, as shown in Figure 10. The distortion of directly decrypted image is evaluated by PNSR. It is observed from Figure 10 that the embedding performance of the method [30] underperforms the other methods due to the limitation of VRAE framework. The embedding performance of the proposed method outperforms those of the compared methods [19,25,35–37] for Boat, Peppers and Barbara when embedding capacity exceeds 0.1 bpp. Moreover, with embedding capacity increases, our method achieves more advantage due to remarkable image recovery efficiency. For Airplane, our capacity-distortion performance outperforms the other compared methods except for Yi et al. [34]. However, the embedding capacity of Yi et al. is limited under 0.5 bpp. For Lena, if embedding capacity is 0.1 bpp, the PSNR of Xiao [36] is higher than ours. When EC exceeds 0.2 bpp, our embedding performance outperforms those of all the compared methods. For Baboon, our embedding performance underperforms some compared methods [19,25,34,36], especial for low embedding capacity. However, when embedding capacity exceeds 0.3 bpp, our method achieves more advantage. Although RRBE-based methods [19,25] can achieve a relatively good performance, the content owner need to conduct an extra preprocess before encryption to vacate room for secret bits. In fact, the content owner may not want to perform this extra preprocess during encryption and/or may not know that secret bits will be embedded into the encrypted image. Furthermore, vacating room from the plain image is effortless than the encrypted image due to the high correlation within pixels of the plain image and the low correlation within pixels of the encrypted image. The proposed method is based on VRBE and the superiority of improved directly decrypted image in our paper is ascribed to remarkable filtering method [42]. In short, Figure 10 demonstrates that the proposed method is better than the compared methods in embedding performance in most cases especially for large embedding capacity.



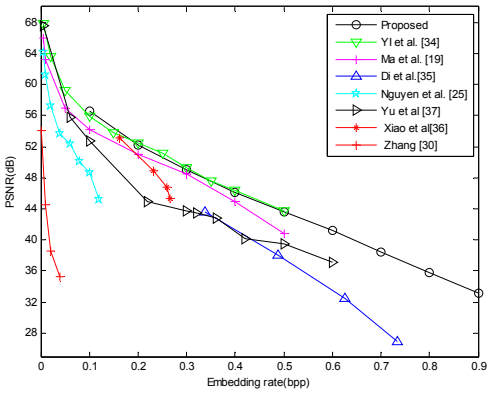
(**a**) Lena



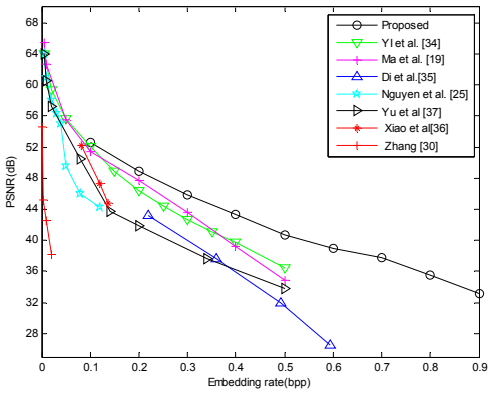
(**b**) Boats



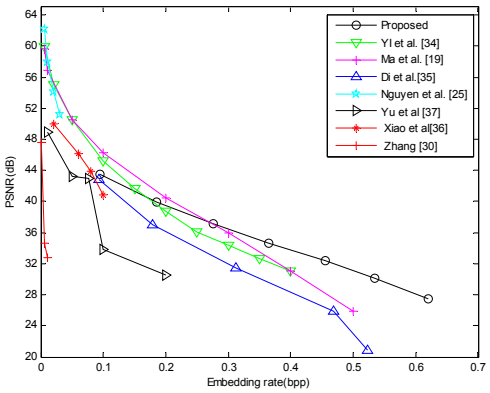
(**c**) Peppers



(**d**) Airplane



(**e**) Barbara



(**f**) Baboon

Figure 10. PSNR comparison among different methods.

4. Conclusions

In this paper, a separable RDHEI method based on two-dimensional permutation and EMD has been proposed. We first design an image encryption method which consists of block permutation and bit-plane permutation. It provides confidentiality for image while keeping pixel correlation in

the encrypted image. Then an error histogram with peak bins and zero bins occurring alternatively is constructed to accommodate for secret bits. Due to the property of the histogram, different EMD methods are adopted for hiding data without HS in order to improve embedding capacity significantly. Different from the previous methods, our method can significantly improve the quality of the directly decrypted image by using this remarkable filtering method. Experimental and comparison results show that the proposed method is better than the several compared methods in terms of the embedding rate and quality of the directly decrypted image.

Author Contributions: Conceptualization, C.Y. (Chunqiang Yu) and X.Z.; writing—original draft preparation, C.Y. (Chunqiang Yu) and Z.T.; validation, C.Y. (Chenmei Ye); writing—review and editing, S.Z.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61762017, Grant 61562007, Grant 61962008, and Grant 81701780, in part by the Guangxi Natural Science Foundation under Grant 2017GXNSFAA198222, and Grant 2017GXNSFBA198221, in part by the Project of Guangxi Science and Technology under Grant GuiKeAD17195062, in part by Science and Technology Planning Project of Guangdong Province, China under Grant 2019B020208001, in part by Educational Commission of Guangdong Province, China under Grant 2018GkQNCX076.

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

Conflicts of Interest: The authors declare no conflicts of interest regarding the publication of this research article.

References

1. Tang, Z.J.; Zhang, X.Q.; Li, X.X.; Zhang, S.C. Robust image hashing with ring partition and invariant vector distance. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 200–214. [CrossRef]
2. Qin, C.; Ji, P.; Zhang, X.P.; Dong, J.; Wang, J.W. Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. *Signal Process.* **2017**, *138*, 280–293. [CrossRef]
3. Tang, Z.J.; Wang, F.; Zhang, X.Q. Image encryption based on random projection partition and chaotic system. *Multimed. Tools Appl.* **2017**, *76*, 8257–8283. [CrossRef]
4. Zhang, G.J.; Liu, Q. A novel image encryption method based on total shuffling scheme. *Opt. Commun.* **2011**, *284*, 2775–2780. [CrossRef]
5. Peng, F.; Lin, Z.X.; Zhang, X.; Long, M. Reversible data hiding in encrypted 2D vector graphics based on reversible mapping model for real numbers. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2400–2411. [CrossRef]
6. Zhang, X.Q.; Sun, Z.R.; Tang, Z.J.; Yu, C.Q.; Wang, X.Y. High capacity data hiding based on interpolated image. *Multimed. Tools Appl.* **2017**, *76*, 9195–9218. [CrossRef]
7. Kuo, W.C.; Wuu, L.C.; Shyi, C.N.; Kuo, S.H. A Data Hiding Scheme with High Embedding Capacity Based on General Improving Exploiting Modification Direction Method. In Proceedings of the 9th International Conference on Hybrid Intelligent Systems (HIS 2009), Shenyang, China, 12–14 August 2009; IEEE: Piscataway, NJ, USA, 2009.
8. Li, S.; Zhang, X.P. Toward Construction-Based Data Hiding: From Secrets to Fingerprint Images. *IEEE Trans. Image Process.* **2019**, *28*, 1482–1497. [CrossRef]
9. Tao, J.; Li, S.; Zhang, X.P.; Wang, Z.C. Towards Robust Image Steganography. *IEEE Trans. Inf. Forensics Secur.* **2019**, *29*, 594–600. [CrossRef]
10. Fridrich, J.; Goljan, M.; Du, R. Lossless data embedding—New paradigm in digital watermarking. *EURASIP J. Adv. Signal Process.* **2002**, *2002*, 185–196. [CrossRef]
11. Celik, M.U.; Sharma, G.; Tekalp, A.M.; Saber, E. Lossless generalized-LSB data embedding. *IEEE Trans. Image Process.* **2005**, *14*, 253–266. [CrossRef]
12. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. [CrossRef]
13. Alattar, A.M. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156. [CrossRef] [PubMed]
14. Ni, Z.C.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–362.
15. Qin, C.; Chang, C.C.; Huang, Y.H.; Liao, L.T. An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 1109–1118. [CrossRef]

16. Ou, B.; Li, X.L.; Zhao, Y.; Ni, R.R. Reversible data hiding using invariant pixel-value-ordering and prediction-error expansion. *Signal Process. Image Commun.* **2014**, *29*, 198–205. [CrossRef]
17. Peng, F.; Li, X.L.; Yang, B. Improved pvo-based reversible data hiding. *Digit. Signal Process.* **2014**, *25*, 255–265. [CrossRef]
18. Ou, B.; Li, X.L.; Zhang, W.M.; Zhao, Y. Improving Pairwise PEE via Hybrid-Dimensional Histogram Generation and Adaptive Mapping Selection. *IEEE Trans. Image Process.* **2019**, *29*, 2176–2190. [CrossRef]
19. Ma, K.D.; Zhang, W.M.; Zhao, X.F.; Yu, N.H.; Li, F.H. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 553–562. [CrossRef]
20. Zhang, W.M.; Ma, K.D.; Yu, N.H. Reversibility improved data hiding in encrypted images. *Signal Process.* **2014**, *94*, 118–127. [CrossRef]
21. Cao, X.C.; Du, L.; Wei, X.X.; Meng, D.; Guo, X.J. High capacity reversible data hiding in encrypted images by patch-level sparse representation. *IEEE Trans. Cybern.* **2016**, *46*, 1132–1143. [CrossRef]
22. Shiu, C.W.; Chen, Y.C.; Hong, W. Encrypted image-based reversible data hiding with public key cryptography from difference expansion. *Signal Process. Image Commun.* **2015**, *39*(Part A), 226–233. [CrossRef]
23. Agrawal, S.; Kumar, M. Mean value based reversible data hiding in encrypted images. *Optik* **2017**, *130*, 922–934. [CrossRef]
24. Yu, C.Q.; Zhang, X.Q.; Tang, Z.J.; Chen, Y.; Huang, J.Y. Reversible Data Hiding with Pixel Prediction and Additive Homomorphism for Encrypted Image. *Secur. Commun. Netw.* **2018**, *2018*, 1–13. [CrossRef]
25. Nguyen, T.; Chang, C.C.; Chang, W. High capacity reversible data hiding scheme for encrypted images. *Signal Process. Image Commun.* **2016**, *44*, 84–91. [CrossRef]
26. Zhang, X.P. Reversible data hiding in encrypted image. *IEEE Signal Process. Lett.* **2011**, *18*, 255–258. [CrossRef]
27. Hong, W.; Chen, T.S.; Wu, H.Y. An improved reversible data hiding in encrypted images using side match. *IEEE Signal Process. Lett.* **2012**, *19*, 199–202. [CrossRef]
28. Liao, X.; Shu, C.W. 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]
29. Qin, C.; Zhang, X.P. Effective reversible data hiding in encrypted image with privacy protection for image content. *J. Vis. Commun. Image Represent.* **2015**, *31*, 154–164. [CrossRef]
30. Zhang, X.P. Separable reversible data hiding in encrypted image. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 826–832. [CrossRef]
31. Qian, Z.X.; Zhang, X.P. Reversible data hiding in encrypted images with distributed source encoding. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 636–646. [CrossRef]
32. Wu, X.T.; Sun, W. High-capacity reversible data hiding in encrypted images by prediction error. *Signal Process.* **2014**, *104*, 387–400. [CrossRef]
33. Yin, Z.Y.; Luo, B.; Hong, W. Separable and error-free reversible data hiding in encrypted image with high payload. *Sci. World J.* **2014**, *2014*, 1–8. [CrossRef] [PubMed]
34. Yi, S.; Zhou, Y.C.; Hua, Z.Y. Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion. *Signal Process. Image Commun.* **2018**, *64*, 78–88. [CrossRef]
35. Di, F.Q.; Huang, F.J.; Zhang, M.Q.; Liu, J.; Yang, X.Y. Reversible data hiding in encrypted images with high capacity by bitplane operations and adaptive embedding. *Multimed. Tools Appl.* **2018**, *77*, 20917–20935. [CrossRef]
36. Xiao, D.; Xiang, Y.P.; Zheng, H.Y.; 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]
37. Yu, C.Q.; Zhang, X.Q.; Tang, Z.J.; Xie, X.J. Separable and Error-Free Reversible Data Hiding in Encrypted Image Based on Two-Layer Pixel Errors. *IEEE Access* **2018**, *6*, 76956–76969. [CrossRef]
38. Tang, Z.J.; Xu, S.J.; Yao, H.; Qin, C.; Zhang, X.Q. Reversible data hiding with differential compression in encrypted image. *Multimed. Tools Appl.* **2019**, *78*, 9691–9715. [CrossRef]
39. Liu, Z.L.; Pun, C.M. Reversible data-hiding in encrypted images by redundant space transfer. *Inf. Sci.* **2018**, *433*, 188–203. [CrossRef]
40. Qin, C.; Qian, X.K.; Hong, W.; Zhang, X.P. An efficient coding scheme for reversible data hiding in encrypted image with redundancy transfer. *Inf. Sci.* **2019**, *487*, 176–192. [CrossRef]

41. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In *Theory and Application of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.
42. Chen, C.L.P.; Liu, L.C.; Chen, L.; Tang, Y.Y.; Zhou, Y.C. Weighted Couple Sparse Representation with Classified Regularization for Impulse Noise Removal. *IEEE Trans. Image Process.* **2015**, *24*, 4014–4026. [CrossRef]
43. Vilardy, J.M.; Torres, C.O.; Jimenez, C.J. Double image encryption method using the Arnold transform in the fractional Hartley domain. In Proceedings of the 8th Iberoamerican Optics Meeting and 11th Latin American Meeting on Optics, Lasers, and Applications, Porto, Portugal, 18 November 2013.
44. Lin, C.H.; Tsai, J.S.; Chiu, C.T. Switching Bilateral Filter with a Texture/Noise Detector for Universal Noise Removal. *IEEE Trans. Image Process.* **2010**, *19*, 2307–2320. [PubMed]