Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 861782, 12 pages
http://dx.doi.org/10.1155/2014/861782

Research Article

A Joint Encryption and Reversible Data Hiding Scheme Based on Integer-DWT and Arnold Map Permutation

Shun Zhang, Tiegang Gao, and Guorui Sheng

College of Software, Nankai University, Wei Jin Road No. 94, Nankai District, Tianjin 300071, China

Correspondence should be addressed to Shun Zhang; shentengvip@gmail.com

Received 6 January 2014; Revised 26 March 2014; Accepted 26 March 2014; Published 9 April 2014

Academic Editor: Feng Gao

A joint encryption and reversible data hiding (joint encryption-RDH) scheme is proposed in this paper. The cover image is transformed to the frequency domain with integer discrete wavelet transform (integer DWT) for the encryption and data hiding. Additional data is hidden into the permuted middle (LH, HL) and high (HH) frequency subbands of integer DWT coefficients with a histogram modification based method. A combination of permutations both in the frequency domain and in the spatial domain is imposed for the encryption. In the receiving end, the encrypted image with hidden data can be decrypted to the image with hidden data, which is similar to the original image without hidden data, by only using the encryption key; if someone has both the data hiding key and the encryption key, he can both extract the hidden data and reversibly recover the original image. Experimental results demonstrate that, compared with existing joint encryption-RDH schemes, the proposed scheme has gained larger embedding capacity, and the distribution of the encrypted image with data hidden has a random like behavior. It can also achieve the lossless restoration of the cover image.

1. Introduction

Compared with traditional watermarking and data hiding schemes, reversible data hiding schemes can be applied in a larger field of secure communication and watermarking due to its reversibility. Many reversible data hiding schemes have been proposed in recent years, which can be classified into three main catalogues: the first one is compression based scheme [1], the second one is difference expansion based scheme [2–5], and the third one is histogram modification based scheme [6–10]. Reversible data hiding based on compression makes use of the redundancy of cover images, so the characters of the cover images limit the capacity and quality of data hiding. Difference expansion based scheme was firstly proposed by Tian [5], which hid one-bit data by extending the difference between two neighbor pixels. Alattar [2–4] improved the hiding capacity by extending $n-1$ pairs of neighbor pixels' differences to hide $n-1$ bitsdata. However, the quality of the cover image drops quickly, while the hiding capacity increases. Schemes based on histogram modification cause less distortion to the cover image. However, the peak points of the histogram limit the hiding capacity [11]. There are two measures to increase the hiding capacity in histogram modification based data hiding schemes: raising the peak points' height or increasing the number of peak points. Many schemes based on the two ways have been proposed. Lin et al. [6] proposed a multilevel embedding strategy to increase the number of peak points. Some schemes increased the height of the peak points through generating the histogram of the difference image. For example, Tsai et al. [8] constructed the difference image by a prediction model that makes full use of the similarity between neighbor pixels. Kim et al. [12] sampled the original image to construct the difference images. A predicted image based on the sampled images was constructed. Then the histograms of difference images between the predicted image and these sampled images were generated for data hiding.

As is well known, encryption is an old and efficient way in secure communication. If combined with encryption, reversible data hiding will achieve greater security. Besides, there are also scenarios that data hiding needs to be done in the encrypted domain or combined with the encryption, especially in the age of big data and cloud computing. A content owner does not trust the processing service provider, and

the ability to manipulate the encrypted data while keeping the plain content unrevealed is desired [13]. Suppose that there are sensitive images uploaded to the cloud storage in the encrypted form and some additional data needs to be hidden into these images to mark their ownership. However, the data hiding process has to be done in the encrypted domain because the data administrator does not have the right and the key to decrypt the image.

In the past few years, some schemes that combine encryption and data hiding have been proposed [13–18]. From the data hider's point of view, data can be hidden into the spatial domain, the encrypted domain [13, 14, 16–18], or both of the two domains [15]. Although high image quality after data hiding has been achieved in [15], the scheme is not reversible. Reversible data hiding schemes in encrypted images are proposed in [13, 14]. In [14], an improved measurement of smoothness is proposed to make full use of all the pixels in the image, and a side match scheme is proposed to further decrease the error rate of extracted bits, both of which have improved the embedding capacity of the basic data hiding scheme in the encrypted image proposed in [13]. In [16], a reversible data hiding scheme in encrypted images by reserving room before encryption is proposed. The self-embedding of LSB planes guarantees the reversibility of LSB substitution embedding. However, the embedding capacity is limited by the embedding capacity of the reversible data hiding scheme in the selected area. In [17], some pixels are selected and estimated before encryption, and additional data is embedded into the estimated errors with a histogram modification method. In the receiving end, one can either decrypt the image with hidden data first or extract the hidden data first. Scheme proposed in [18] separates the data extraction and the recovery of original image. The image is encrypted with the encryption key. Then the encrypted image is passed to the data hider, and additional data is embedded into the encrypted image with the data hiding key. In the receiving end, the hidden data can be extracted with only the data hiding key; and only similar (not reversible) image can be recovered with the encryption key; both the hidden data can be extracted and original image can be reversibly recovered with both keys.

Different from all the joint encryption and data hiding schemes mentioned above, a joint encryption and reversible data hiding scheme based on integer DWT and Arnold map permutation is proposed. Not in the spatial domain, the data hiding is imposed in the integer DWT domain, which is more secure compared with those schemes in the spatial domain. The cover image is firstly transformed to the frequency domain with discrete integer wavelet transform (integer DWT). Then coefficients of the four subbands are permutated with Arnold map transform, respectively, for the first time. After that, additional data is embedded into the permuted middle (LH, HL) and high (HH) frequency subbands through a histogram modification based method. Finally, inverse integer DWT is imposed to get the primary encrypted image with hidden data. Another Arnold permutation based on sampling, which is related to the permutation in the frequency domain, is imposed on the primary encrypted image with hidden data in the spatial domain. In the receiving end, one can decrypt the image to get the image with hidden data, which is similar to the original image without hidden data, by only using the encryption key that includes permutation times of the twice permutations. If someone has both the encryption key and the data hiding key, he can both extract the hidden data and reversibly recover the original image. Note that the processing procedures in the sending end and in the receiving end described here are asymmetric, which can achieve many applications, such as scenarios mentioned above.

2. Preliminaries

2.1. Integer DWT. To achieve the reversible data hiding, reversible lifting integer DWT is applied. Integer DWT is implemented with the addition and subtraction of integers. Suppose that $I(x, y)$, $1 \le x \le M$, $1 \le y \le N$, is the pixel of the image size of $M \times N$; then 2D integer DWT is conducted as follows.

(A) Row Transformation

(1) Let $f1 = I(2 * i - 1, :)$ and $f2 = I(2 * i, :)$, $i = 1, 2, \ldots, M/2$, which are odd rows and even rows of I, respectively.

(2) Acquire the high frequency coefficients by calculating the difference of the two: $h\_r(i, :) = f1(i, :) - f2(i, :)$.

(3) Acquire the low frequency coefficients by calculating the average of the two: $l\_r(i, :) = f2(i, :) + \text{floor}(h(i, :))$.

(4) Then coefficients after 1D transformation are $C\_\text{row} = [l\_r; h\_r]$.

(B) Column Transformation

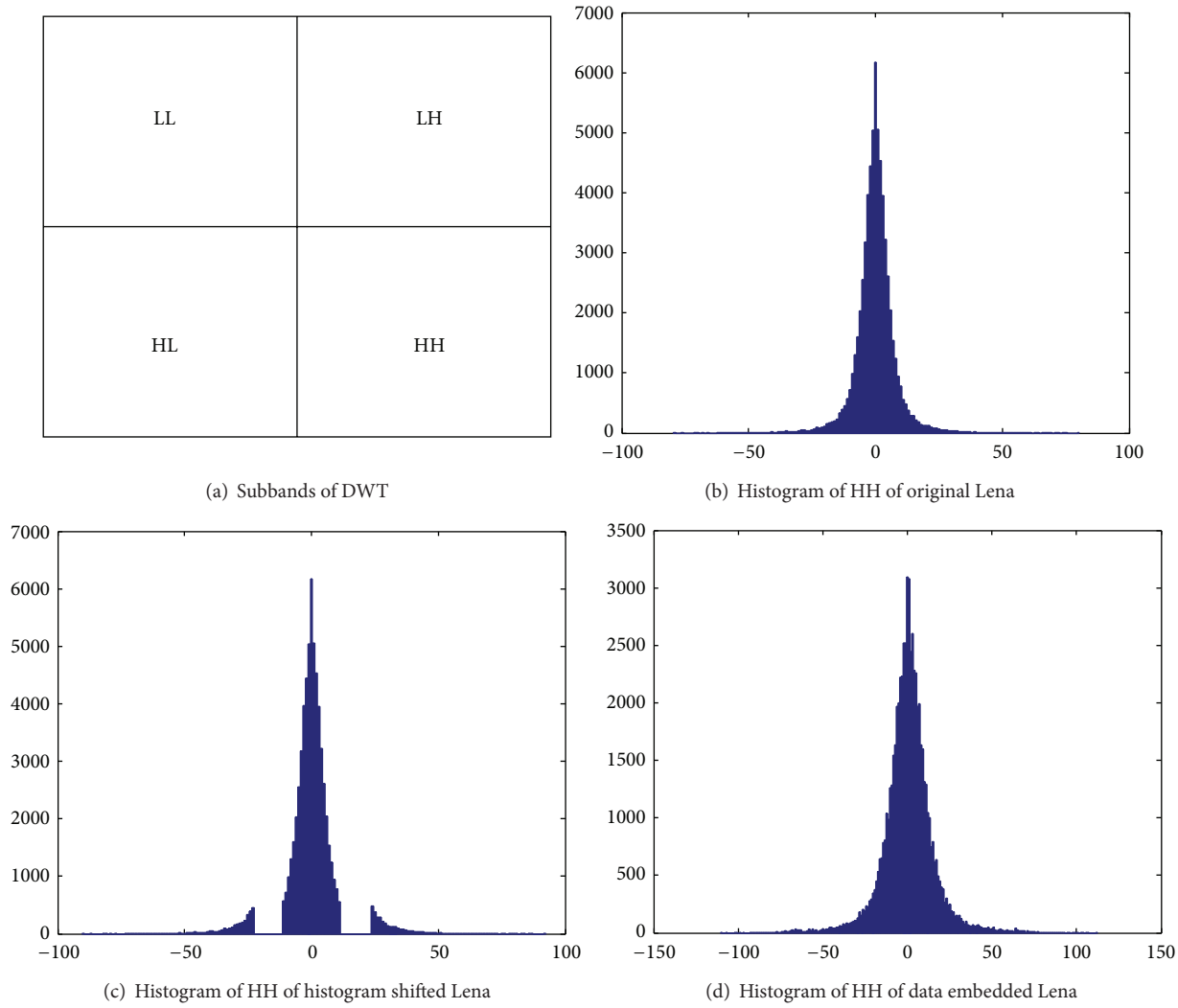
(1) Let $f1 = I(:, 2 * i - 1)$ and $f2 = C\_\text{row}(:, 2 * i)$, $i = 1, 2, \ldots, N/2$, which are odd columns and even columns of C_row, respectively.

(2) Acquire the high frequency coefficients by calculating the difference of the two: $h\_c(:, i) = f1(:, i) - f2(:, i)$.

(3) Acquire the low frequency coefficients by calculating the average of the two: $l\_c(:, i) = f2(:, i) + \text{floor}(h(:, i))$.

(4) Finally, the coefficients of 2D integer DWT are $C = [l\_c \;\; h\_c]$.

2.2. Reversible Data Hiding and Data Extraction Based on Histogram Expansion. Ni et al. [19] firstly proposed reversible data hiding based on histogram modification. It generates the histogram of an image; then a pair of peak point and zero point is found out in the histogram, and the histogram between peak point and zero point is shifted to the zero point side to produce the gap for data hiding. Very little distortion will be caused by such schemes, and Ni et al. [19] have pointed out that the peak signal-to-noise ratio (PSNR) between the original image and the image with hidden data is above 48. As mentioned in the beginning part of the paper, the drawback is the rare capacity of data hiding. A novel histogram modification based reversible data hiding scheme in integer DWT domain, which increases the capacity of data hiding greatly,



(a) Subbands of DWT

(b) Histogram of HH of original Lena

(c) Histogram of HH of histogram shifted Lena

(d) Histogram of HH of data embedded Lena

Figure 1: An example of histogram modification (HH subband of integer DWT of Lena).

is described here. Histograms of middle and high frequency subbands of integer DWT of images are Laplacian like distribution. Thus, they are suitable for histogram modification based data hiding method. Histograms are shifted to generate the gap for data hiding. A demo of histogram modification based data hiding method, which embeds data into the HH subbands of Lena image, is presented in Figure 1.

2.2.1. Reversible Data Embedding. The generated histogram of subband HH is depicted in Figure 1(b). Then the histogram is shifted to both sides by an embedding strength T (Figure 1(c)). At last, data is embedded by expanding histogram between T and $-T$, and the histogram after embedding is as Figure 1(d).

The histograms of LH, HL, and HH subbands are generated and data is embedded into the coefficients by histogram modification. For every coefficient C of LH, HL, and HH subbands, given an embedding strength parameter q,

(1) if $C \geq q$, then C is shifted to $C + q$;

(2) else if $C \leq -q$, then C is shifted to $C - q + 1$;

(3) else $C \leftarrow 2 \times C + B$, and B is the data to be embedded.

2.2.2. Data Extraction and Reversible Recovery of Matrix before Embedding. Generate the histograms of middle and high frequency subbands and shift these histograms to extract the hidden data, and the original coefficient matrices are reversibly recovered through the following steps. For every coefficient C of LH, HL, and HH subbands, given an embedding strength parameter q,

(1) if $C \geq 2 \times q$, then C is shifted to $C - q$;

(2) else if $C \leq -2 \times q + 1$, then C is shifted to $C + q - 1$;

(3) else $C \leftarrow \text{floor}(C/2)$, and data is extracted: $B = \text{mod}(C, 2)$.

Now, every coefficient C of subbands LH, HL, and HH is reversibly recovered and the extracted B is the data embedded before.

2.3. Arnold Permutation [20]. Russian mathematician Vladimir I. Arnold discovered Arnold's cat map using an image of cat. An image, not necessarily a cat, of course, can be transformed to a random noise like image by rearranging the position of original pixels. However, if iterated for moderate times (denoted by permutation periods as presented in Table 1), the original image will reappear. The permutation periods differ as the sizes of images differ. The permutation periods of images with different sizes of the traditional Arnold permutation are presented in Table 1.

Let $I(x, y)$ be the pixel of an image matrix with size $N \times N$; then $\begin{bmatrix} x \\ y \end{bmatrix}$ represents the position of the pixel. The Arnold transform Γ can be explained as $\Gamma \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x+y \\ x+2y \end{bmatrix} \bmod n$, where mod is the modulo-operation.

To better explain the theory, the transform can be decomposed into three elemental steps: in the x-direction: $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x+y \\ y \end{bmatrix}$, in the y-direction: $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} y \\ x+y \end{bmatrix}$, and in the modulo-operation: $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} \bmod n$.

2.4. Permutation in the Frequency Domain and Sample Permutation in the Spatial Domain. Through permutation in the frequency domain, nice encryption results will be achieved. A novel encryption scheme based on the cooperation of permutation in the frequency domain and sample permutation in the spatial domain is proposed to accommodate the joint encryption-RDH scheme in this paper. It is found out through experiments that the proposed permutation in the integer DWT domain can achieve the same results as the proposed sample permutation scheme in the spatial domain. Such features are applied in the design of joint encryption-RDH scheme. Suppose that there is an image matrix M with size $N \times N$, and the permutation in the integer DWT domain and sample permutation in the spatial domain are described, respectively, in the followings.

(A) Permutation in the Integer DWT Domain

(1) Decompose the original image matrix M with integer DWT to obtain the four subbands (1 low frequency subband LL, 2 middle frequency subbands LH and HL, and 1 high frequency subband HH) as depicted in Figure 1(a).

(2) Permute the four subbands after integer DWT with Arnold map permutation that is presented in Section 2.1, and the different permutation times are $P1\_T$, $P2\_T$, $P3\_T$, and $P4\_T$, respectively:

$$\begin{aligned} \mathrm{LL}' &= \mathrm{Arnold}\,(\mathrm{LL}, P1\_T)\,, \\ \mathrm{LH}' &= \mathrm{Arnold}\,(\mathrm{LH}, P2\_T)\,, \\ \mathrm{HL}' &= \mathrm{Arnold}\,(\mathrm{HL}, P3\_T)\,, \\ \mathrm{HH}' &= \mathrm{Arnold}\,(\mathrm{HH}, P4\_T)\,. \end{aligned} \tag{1}$$

(3) Impose invers integer DWT on the coefficients after permutation to get the encrypted matrix M'.

(B) Sample Permutation in the Spatial Domain

(1) Sample matrix M into four submatrices:

$$\begin{aligned} \mathrm{Sam1} &= M\,(1:2:N-1, 1:2:N-1)\,, \\ \mathrm{Sam2} &= M\,(2:2:N, 1:2:N-1)\,, \\ \mathrm{Sam3} &= M\,(1:2:N-1, 2:2:N)\,, \\ \mathrm{Sam4} &= M\,(2:2:N-1, 2:2:N)\,. \end{aligned} \tag{2}$$

(2) Permute the four sampled submatrices with Arnold map with different permutation times $S1\_T$, $S2\_T$, $S3\_T$, and $S4\_T$:

$$\begin{aligned} \mathrm{Sam1}' &= \mathrm{Arnold}\,(\mathrm{Sam1}, S1\_T)\,, \\ \mathrm{Sam2}' &= \mathrm{Arnold}\,(\mathrm{Sam2}, S2\_T)\,, \\ \mathrm{Sam3}' &= \mathrm{Arnold}\,(\mathrm{Sam3}, S3\_T)\,, \\ \mathrm{Sam4}' &= \mathrm{Arnold}\,(\mathrm{Sam4}, S4\_T)\,. \end{aligned} \tag{3}$$

(3) Compose the permuted sampled submatrix to get the encrypted matrix M'

$$\begin{aligned} M'\,(1:2:N-1, 1:2:N-1) &= \mathrm{Sam1}'\,, \\ M'\,(2:2:N, 1:2:N-1) &= \mathrm{Sam2}'\,, \\ M'\,(1:2:N-1, 2:2:N) &= \mathrm{Sam3}'\,, \\ M'\,(2:2:N-1, 2:2:N) &= \mathrm{Sam4}'\,. \end{aligned} \tag{4}$$

If the permutation times are equal to the corresponding permutation times in the two permutation schemes, which mean that $P1\_T = S1\_T$, $P2\_T = S2\_T$, $P3\_T = S3\_T$, and $P4\_T = S4\_T$, the encryption results are equivalent.

3. Proposed Scheme

Different from existing joint encryption-RDH schemes [13–18], which are based on the spatial domain, the proposed scheme is based on the integer DWT domain. The detailed joint encryption-RDH scheme is presented in this section. The scheme is composed of two parts. One part is data hiding and image encryption, as presented in Figure 2; the other part is data extraction and original image recovery, which is presented in Figure 3. Note that the two parts are not symmetric. The encryption is achieved with permutation before data hiding and after data hiding in Figure 2, while the data extraction is after decryption in Figure 3. The asymmetric design can be applied in such a scenario. When someone only has the encryption key, he can decrypt the image and get the decrypted image with hidden information, which is very similar to the original image. The decrypted image can be utilized

Table 1: Arnold permutation periods of images with different sizes.

Image size	512 × 512	256 × 256	128 × 128	64 × 64	32 × 32	16 × 16	8 × 8	4 × 4
Period	384	192	96	48	24	12	6	3

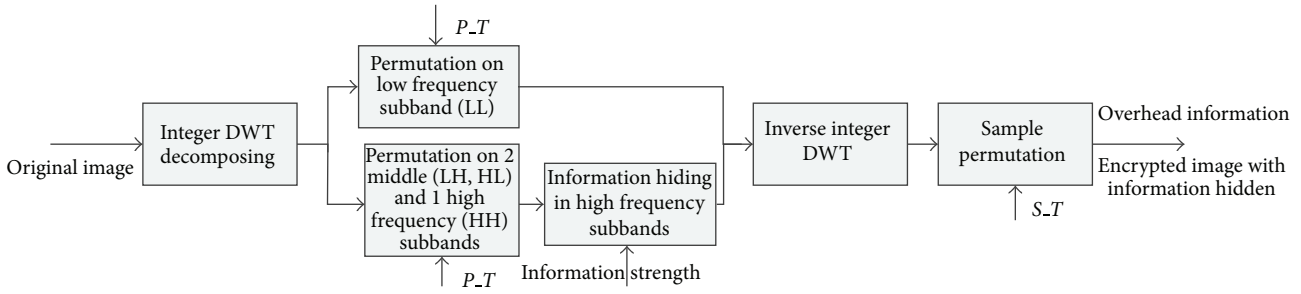


Figure 2: Data hiding and image encryption.

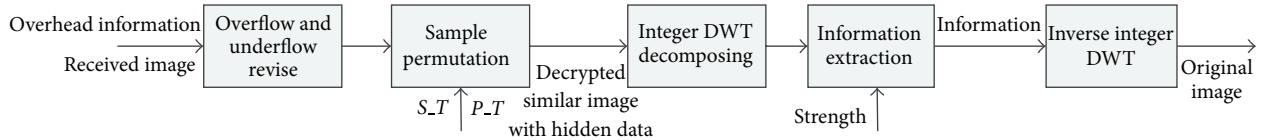


Figure 3: Data extraction and image recovery.

in a variety of applications. However, he cannot get rid of the hidden data, which may work as the watermark for the copyright or authentication.

In the sending end, the data hiding and image encryption process are achieved alternately. Original image I is firstly decomposed with the integer DWT proposed in Section 2.1. Then, the Arnold permutation is imposed on the four subbands for the first time encryption. Data is embedded into the permuted middle and high frequency subbands with a reversible data hiding scheme based on histogram modification. After that inverse integer DWT is imposed to acquire the primary permuted image with hidden data I'. Finally, a sample-permutation scheme is imposed on I' to get the final encrypted image with hidden data I''. Because the reversible data hiding is based on histogram modification, overflow/underflow is hard to avoid. Therefore, a location map for recording the positions and values of the underflow and overflow pixels in the spatial domain is constructed. The location map is compressed and encrypted as the key for data extraction. The flow chart is presented in Figure 2.

In the receiving end, there are two cases. One is simple decryption, and the other one is data extraction and original image recovery. In the former case, the image with hidden data that is similar to the original image I is decrypted. In the latter case, the hidden data is extracted and the original image I is reversibly recovered. The received encrypted image with hidden data I'' is firstly revised according to the location map. Then it is decrypted into the image with hidden data. Note that, the encryption based on the two permutations in the sending end can be decrypted by one permutation based on the sample-permutation method. The permutation based on sample permutation in the spatial domain can achieve the same results as the permutation in the integer DWT domain as presented in Section 2.4. Besides, the data hiding scheme based on histogram modification in the integer DWT domain can be implemented either before or after the permutation in the integer DWT domain. Both of the features guarantee the asymmetric decryption and data extraction. The total permutation times are calculated according to the Arnold permutation periods of image with different sizes (Table 1), the permutation time P_T in the integer DWT domain permutation, and the permutation time S_T of the sample permutation in the spatial domain. Through the delicate design of the two permutations, decryption can be done without integer DWT and inverse integer DWT. Although additional data is hidden in the permutated integer DWT domain, the proposed histogram modification based data hiding scheme in the integer DWT domain guarantees the integrity of the hidden data and the reversibility of the original cover image. The flow chart is presented in Figure 3.

3.1. Data Hiding and Image Encryption

(1) Decompose the original image I (with size $N \times N$) with integer DWT (proposed in Section 2.1) to obtain the four subbands (one low frequency subband LL, two middle frequency subbands LH and HL, and one high frequency subband HH) as depicted in Figure 1(a).

(2) Permute the four subbands (LL, LH, HL, and HH) with Arnold map permutation (proposed in

Section 2.3) synchronously. Note that, the permutation times are the same, denoted by P_T:

$$\begin{aligned} \mathrm{LL} &= \mathrm{Arnold}\,(\mathrm{LL}, P\_T)\,, \qquad \mathrm{LH} = \mathrm{Arnold}\,(\mathrm{LH}, P\_T)\,, \\ \mathrm{HL} &= \mathrm{Arnold}\,(\mathrm{HL}, P\_T)\,, \qquad \mathrm{HH} = \mathrm{Arnold}\,(\mathrm{HH}, P\_T)\,. \end{aligned} \tag{5}$$

(3) Embed the preprocessed data into the middle and high frequency subbands (LH, HL, and HH) with histogram modification based method (proposed in Section 2.2.1).

(4) Impose invers integer DWT on the coefficients after permutation and data hiding to get the primary permuted image with hidden data, denoted by I'.

(5) Sample I' to four subimages:

$$\begin{aligned} \mathrm{Sam1} &= I'\,(1:2:N-1, 1:2:N-1)\,, \\ \mathrm{Sam2} &= I'\,(2:2:N, 1:2:N-1)\,, \\ \mathrm{Sam3} &= I'\,(1:2:N-1, 2:2:N)\,, \\ \mathrm{Sam4} &= I'\,(2:2:N-1, 2:2:N)\,. \end{aligned} \tag{6}$$

(6) Permute the four sampled submatrices with Arnold map with different permutation times $S1\_T$, $S2\_T$, $S3\_T$, and $S4\_T$:

$$\begin{aligned} \mathrm{Sam1}' &= \mathrm{Arnold}\,(\mathrm{Sam1}, S1\_T)\,, \\ \mathrm{Sam2}' &= \mathrm{Arnold}\,(\mathrm{Sam2}, S2\_T)\,, \\ \mathrm{Sam3}' &= \mathrm{Arnold}\,(\mathrm{Sam3}, S3\_T)\,, \\ \mathrm{Sam4}' &= \mathrm{Arnold}\,(\mathrm{Sam4}, S4\_T)\,. \end{aligned} \tag{7}$$

(7) Compose the permuted sampled submatrix to get the postprocessed matrix I'':

$$\begin{aligned} I''\,(1:2:N-1, 1:2:N-1) &= \mathrm{Sam1}', \\ I''\,(2:2:N, 1:2:N-1) &= \mathrm{Sam2}', \\ I''\,(1:2:N-1, 2:2:N) &= \mathrm{Sam3}', \\ I''\,(2:2:N-1, 2:2:N) &= \mathrm{Sam4}'. \end{aligned} \tag{8}$$

(8) Construct the location map L of overflow and under flow pixels according to I'', and for those few overflow/underflow pixels, change their values with random integer value in the range of (0,255).

These permutation times are encoded and encrypted as the encryption key. The location map and embedding strength parameters are compressed and encrypted as the data hiding key.

3.2. Data Extraction and Original Image Recovery

(1) Revise the received image matrix according to location map to get image I''.

(2) Sample image I'' with size $N \times N$ into four submatrices:

$$\begin{aligned} \mathrm{Sam1\_}r &= I''\,(1:2:N-1, 1:2:N-1)\,, \\ \mathrm{Sam2\_}r &= I''\,(2:2:N, 1:2:N-1)\,, \\ \mathrm{Sam3\_}r &= I''\,(1:2:N-1, 2:2:N)\,, \\ \mathrm{Sam4\_}r &= I''\,(2:2:N-1, 2:2:N)\,. \end{aligned} \tag{9}$$

(3) Permute the four sampled submatrices with Arnold map with different permutation times $S1\_T'$, $S2\_T'$, $S3\_T'$, and $S4\_T'$, respectively:

$$\begin{aligned} \mathrm{Sam1\_}r' &= \mathrm{Arnold}\left(\mathrm{Sam1\_}r, S1\_T'\right), \\ \mathrm{Sam2\_}r' &= \mathrm{Arnold}\left(\mathrm{Sam2}, S2\_T'\right), \\ \mathrm{Sam3\_}r' &= \mathrm{Arnold}\left(\mathrm{Sam\_}r, S3\_T'\right), \\ \mathrm{Sam4\_}r' &= \mathrm{Arnold}\left(\mathrm{Sam4\_}r, S4\_T'\right), \end{aligned} \tag{10}$$

where $S1\_T' = T - P\_T - S1\_T$, $S2\_T' = T - P\_T - S2\_T$, $S3\_T' = T - P\_T - S3\_T$, and $S4\_T' = T - P\_T - S4\_T$. T is the permutation period of the sample images, and for the sample images with size 256×256, $T = 192$, just as presented in Table 1.

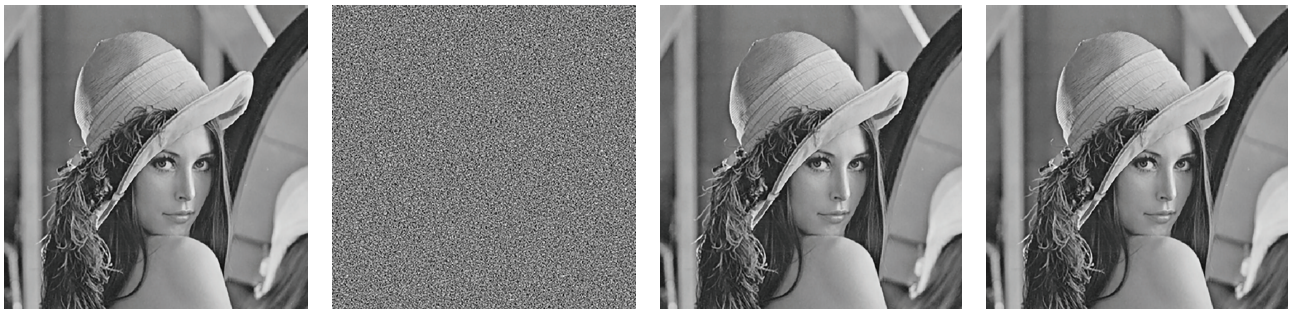
(4) Compose the permuted sampled submatrix to get the similar image with hidden data M_s:

$$\begin{aligned} M\_s\,(1:2:N-1, 1:2:N-1) &= \mathrm{Sam1\_}r', \\ M\_s\,(2:2:N, 1:2:N-1) &= \mathrm{Sam2\_}r', \\ M\_s\,(1:2:N-1, 2:2:N) &= \mathrm{Sam3\_}r', \\ M\_s\,(2:2:N-1, 2:2:N) &= \mathrm{Sam4\_}r'. \end{aligned} \tag{11}$$

Until now, the decryption process has been completed, and the similar image with hidden data is M_s. The hidden data can be extracted and the original image I can be reversibly recovered through the following steps.

(5) Impose integer DWT on image M_s to get the four subbands LL, LH, HL, and HH.

(6) Generate the histograms of the middle (LH, HL) and high (HH) frequency subbands and shift these histograms to extract the hidden data and reversibly recover the original subbands. The detailed steps are depicted in Section 2.2.2.



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

Figure 4: Images before and after disposing.

(7) Impose inverse integer DWT with the coefficients of the subbands after histogram shifting to recover the original image.

4. Experimental Results and Analysis

To testify the efficiency and validity of the proposed scheme, images (with size 512×512) from Miscellaneous gray level images [21] and USC-SIPI image database [22] are selected for the experiments. Random binary bits are embedded into these images as the hidden data. All of these experiments are performed on the MATLAB 2012a platform running on a personal computer with CPU of AMD Phenom (tm) IIX4 810 Processor at 2.6 GHz, memory of 4 GB, and operating system of Windows 7 x64 Ultimate Edition.

In Figure 4, standard image "Lena" is adopted to demonstrate the feasibility of the proposed scheme. The subfigure (a) is the original Lena, (b) is encrypted image with embedding rate 0.0827 bpp, (c) is decrypted image with data embedded (PSNR = 50.7279), and (d) is the reversibly recovered image.

The hiding capacity with different embedding strength parameters, the corresponding PSNRs after data hiding, and the overhead data needed to dispose for the reversible recovery of the original image are presented in Tables 2 and 3.

As is seen in the tables, the embedding strength parameter q is 1, 2, 4, 8, 16, and 32, respectively. The embedding rates (ER) increase as the embedding strength parameters increase. In Table 2, images from USC-SIPI image database are tested. In Table 3, images from Miscellaneous gray level images are tested. It is easily seen that the overhead data for reversible recovery of the original image is rare and even zero for most of the test images. However, it is necessary especially when multilevel embedding is utilized. If the location map is transferred as a part of the payload, the pure embedding rates (PER) that exclude the overhead are also given in the table.

As can be seen in Tables 2 and 3, the embedding rates increase as the embedding strength parameter q increases. However, more overhead information is generated in accompany with the increase of embedding rate and the embedding strength parameter q. More distortion will be caused by the greater amount of data hiding. Different images have different sensitivity to the embedding strength parameter q. Smooth images, such as "Airplane," "Lena," and "Boat," are less sensitive to the parameter q than those complex images, such as "Baboon" and "Peppers." That is because the histogram shifting based data hiding scheme imposed in the integer DWT domain depends largely on the similarity of adjacent pixels in the images.

Given the fix embedding rate, the plots between PSNR and embedding rate with different embedding strength parameters after decryption are demonstrated in Figure 5. The test images are selected from Miscellaneous gray level images database. The embedding strength of subfigure (a) is $q = 32$, (b) $q = 16$, (c) $q = 8$, (d) $q = 4$, (e) $q = 2$, and (f) $q = 1$, respectively.

The security of the proposed scheme is testified. As is known, there are similarities between adjacent pixels in natural images. One of the important things for the encryption of image is to destroy the correlation between two adjacent pixels. It can be calculated by the following formulas:

$$\begin{aligned} E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2, \\ \operatorname{cov}(x, y) &= \frac{1}{N}\sum_{i=1}^{N} E\left(x - E(x)\left(y - E(y)\right)\right), \\ r_{xy} &= \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}. \end{aligned} \tag{12}$$

We randomly select 4096 pairs of two adjacent horizontal pixels, two adjacent vertical pixels, and two adjacent diagonally pixels in "Lena" image, respectively, for the demonstration. Figure 6 presents the correlation of adjacent pixels of image "Lena" before encryption and after encryption. The detail coefficients r_{xy} of selected images from Miscellaneous gray level images are presented in Table 4.

Obviously, the similarities have been thoroughly destroyed after encryption. Through the delicate design of the

Table 2: Embedding rate and PSNR of different images (USC-SIPI).

Images		$q = 1$	$q = 2$	$q = 4$	$q = 8$	$q = 16$	$q = 32$
Airplane	ER	0.1206	0.3101	0.5044	0.6311	0.6971	0.7317
	PSNR	50.6962	46.8158	42.4294	38.5350	35.0636	32.1368
	Overhead	0	0	0	0	0	0
	PER	0.1206	0.3101	0.5044	0.6311	0.6971	0.7317
Baboon	ER	0.0251	0.0756	0.1707	0.3210	0.4960	0.6421
	PSNR	50.5884	46.0507	40.1912	34.7103	30.0639	26.4296
	Overhead	14	26	53	118	231	405
	PER	0.0250	0.0755	0.1705	0.3205	0.4951	0.6406
Barbara	ER	0.0656	0.1882	0.3602	0.5124	0.6113	0.6887
	PSNR	50.5979	46.3715	41.2511	36.5779	32.3365	28.7201
	Overhead	0	0	0	0	0	23
	PER	0.0656	0.1882	0.3602	0.5124	0.6113	0.6886
Boat	ER	0.0545	0.1577	0.3223	0.5162	0.6665	0.7287
	PSNR	50.5664	46.3085	41.0439	36.5185	33.1225	30.7374
	Overhead	3	10	19	38	109	296
	PER	0.0545	0.1576	0.3223	0.5161	0.6661	0.7276
Lena	ER	0.0827	0.2241	0.4310	0.6119	0.7032	0.7391
	PSNR	50.7257	46.5336	41.7441	37.8559	31.8851	32.7387
	Overhead	0	0	0	0	0	5
	PER	0.0827	0.2241	0.4310	0.6119	0.7032	0.7390
Peppers	ER	0.0632	0.1844	0.3743	0.5716	0.6961	0.7385
	PSNR	50.6754	46.4399	41.4909	37.5625	34.7859	32.8502
	Overhead	1	5	27	79	213	526
	PER	0.6632	0.1843	0.3472	0.5713	0.6953	0.7365

Table 3: Embedding rate and PSNR of different images (Miscellaneous gray level images).

Images		$q = 1$	$q = 2$	$q = 4$	$q = 8$	$q = 16$	$q = 32$
Airplane	ER	0.1227	0.3110	0.5071	0.6342	0.6988	0.7325
	PSNR	50.6757	46.8180	42.4507	38.5947	35.1570	32.2588
	Overhead	0	0	0	0	0	0
	PER	0.1227	0.3110	0.5071	0.6342	0.6988	0.7325
Baboon	ER	0.0252	0.0754	0.1702	0.3211	0.4961	0.6421
	PSNR	50.5954	46.0404	40.1840	34.7072	30.0631	26.4306
	Overhead	19	36	65	138	237	433
	PER	0.0251	0.0753	0.1700	0.3206	0.4953	0.6405
Barbara	ER	0.0604	0.1704	0.3350	0.4932	0.6047	0.6852
	PSNR	50.6255	46.3325	41.0880	36.3009	32.0515	28.5263
	Overhead	0	0	0	0	19	219
	PER	0.0604	0.1704	0.3350	0.4932	0.6046	0.6484
Boat	ER	0.0857	0.2367	0.4296	0.5819	0.6791	0.7295
	PSNR	50.6239	46.5614	41.8009	37.5187	33.8471	31.0650
	Overhead	1	1	1	1	2	38
	PER	0.0857	0.2367	0.4296	0.5819	0.6791	0.7294
Lena	ER	0.0837	0.2239	0.4304	0.6117	0.7031	0.7390
	PSNR	50.7225	46.5535	41.7346	37.8432	34.8781	32.7376
	Overhead	0	0	0	0	0	5
	PER	0.0837	0.2239	0.4304	0.6117	0.7031	0.7390
Peppers	ER	0.0673	0.1924	0.3838	0.5833	0.6995	0.7379
	PSNR	50.6647	46.4457	41.5086	37.5847	34.8350	32.8801
	Overhead	30	92	183	545	1119	1826
	PER	0.0672	0.1921	0.3810	0.5812	0.6952	0.7371

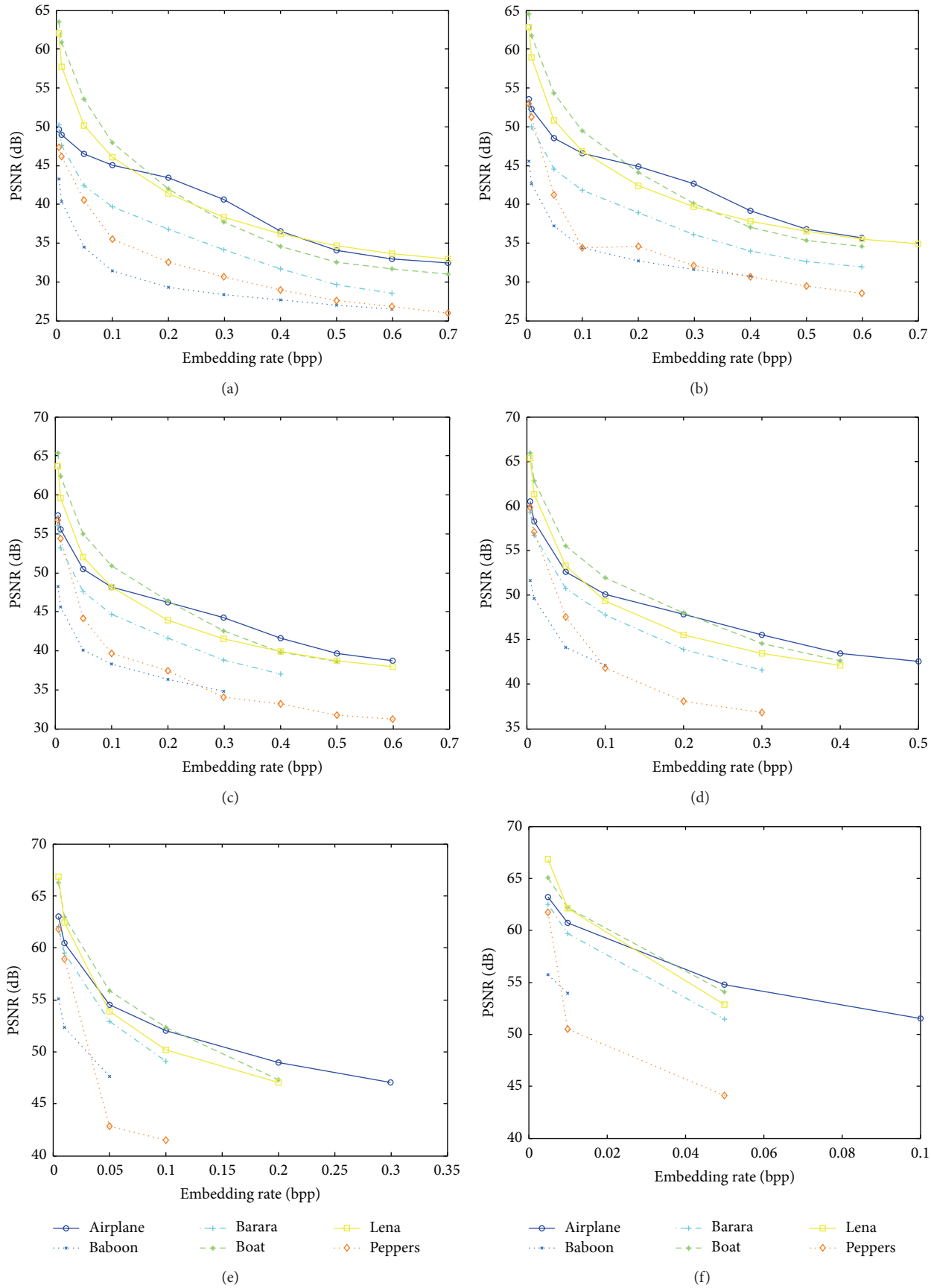


Figure 5: Embedding rate and PSNR of different images with different embedding strength parameter.

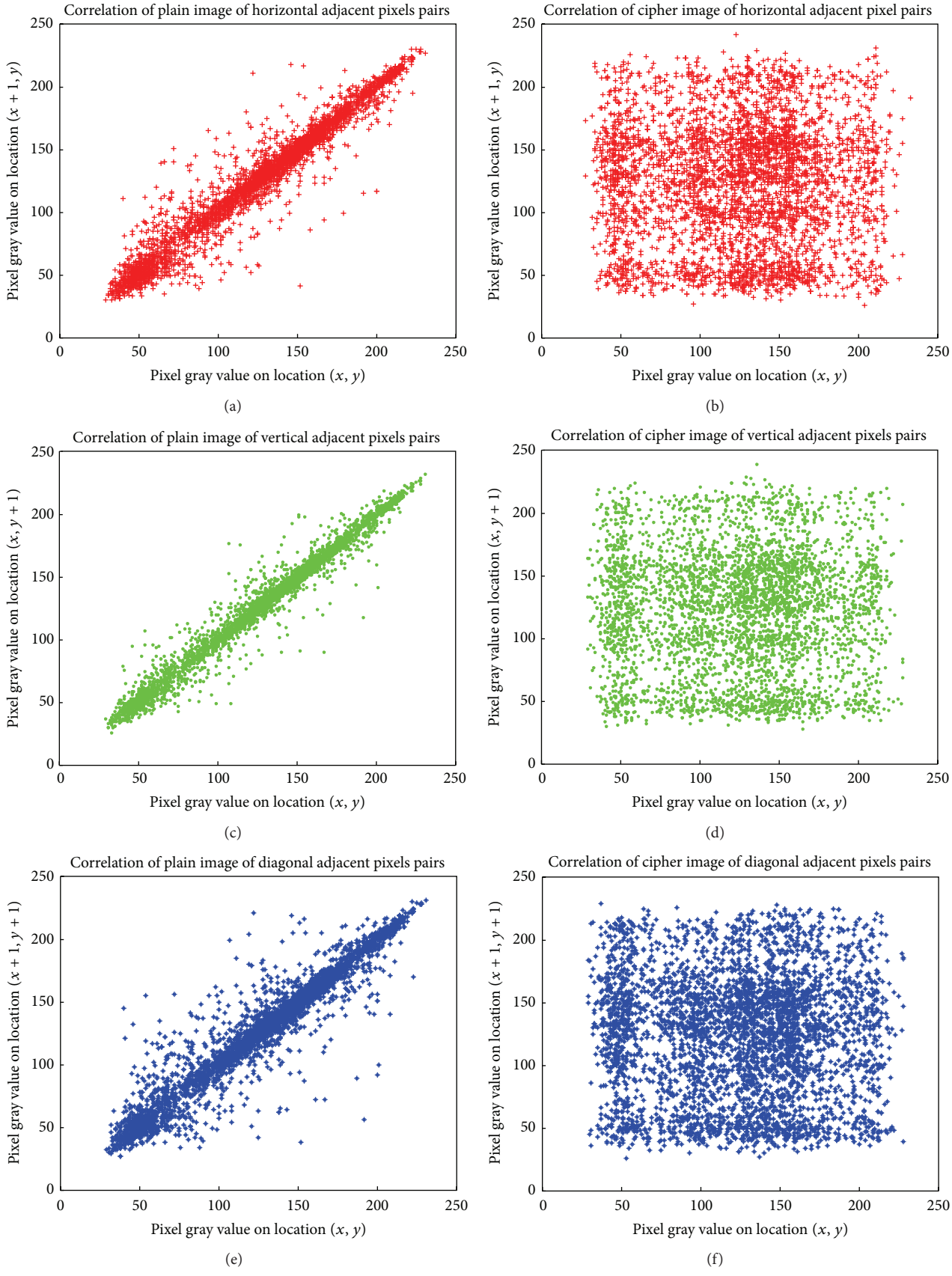
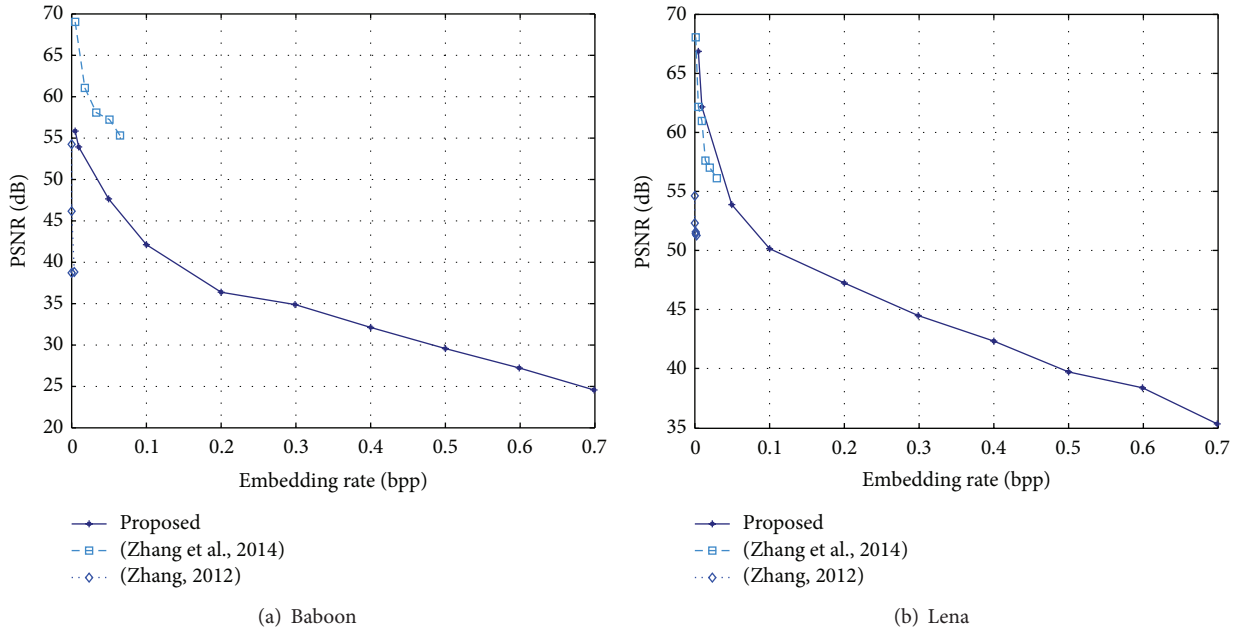


Figure 6: Correlations of two adjacent pixels in the plain image and in the cipher image of Lena.

Table 4: Coefficients of different images.

Coefficients	Plain image			Cypher image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Airplane	0.9728	0.9648	0.9416	−0.0092	0.0008	−0.0054
Baboon	0.8407	0.7500	0.7160	0.0437	0.0101	0.0162
Barbara	0.9076	0.9648	0.8898	−0.0142	−0.0078	−0.0222
Boat	0.9531	0.9827	0.9405	0.0320	−0.0088	−0.0090
Lena	0.9711	0.9851	0.9598	−0.0022	−0.0204	0.0215
Peppers	0.9779	0.9777	0.9665	−0.0400	0.00474	−0.0183



(a) Baboon (b) Lena

Figure 7: Comparisons of embedding rate and PSNR with existing schemes.

permutation in both transformed domain and spatial domain, the simple Arnold permutation can achieve nice encryption results. Besides, the encryption scheme proposed in this paper is efficient and timesaving due to the permutation only scheme. The stream cipher based encryption [13–15, 18] is more time-consuming because the encryptions are achieved by the bitwise exclusive OR operation or even the RC4 and AES encryption. Simulation results show that, for the images with size 512 × 512, the average time for image encryption and data hiding is 4.3012 s and the average time for decryption and data extraction is 4.0013 s using the proposed scheme. If the same amount of data is embedded in the images with size 512 × 512, the average encryption and data hiding time is more than 8.0332 s, and the average decryption and data extraction time is more than 7.8231 s for the encryption scheme with bitwise exclusive OR operation with hyperchaotic system.

The joint-RDH scheme proposed in [15] is applied in the medical images. It is not reversible. The joint-RDH scheme proposed in [14] increased the embedding capacity of the scheme proposed in [13]. However, their embedding capacity is rather low when the reversibility is achieved due to the design of the data hiding. At least a 8 × 8 block is needed for embedding one-bit information in their experiments. Therefore, the embedding rate is no more than 1/(8 × 8) according to their experiments. Detailed comparisons of the embedding rate are presented in Table 5. Comparisons of the plot between PSNR and embedding rate with scheme proposed in [17] and in [18] are presented in Figure 7.

Table 5: Embedding rate comparison with existing schemes.

ER (bpp)	Baboon	Lena	Lake	Man	Splash
Reference [14]	0.0013	0.0069	0.0025	0.0024	0.0156
Reference [13]	0.0010	0.0039	0.0025	0.0025	0.0039
Proposed	0.6405	0.7390	0.6381	0.6659	0.7123

Obviously, the proposed scheme has been achieved better performances compared with exiting schemes. The reason why the PSNRs are higher at the same embedding rate is that, in the proposed scheme, data is hidden in the transformed domain through the difference histogram modification method. Such reversible data hiding schemes can achieve lager embedding capacity while keeping low distortion to the cover image. The encryption is achieved through the corporation of permutation in the integer DWT domain and in the

spatial domain. Moreover, the permutation in the integer DWT domain will not affect the data hiding. Due to the design of the existing joint encryption-RDH schemes [13, 14, 17, 18], a group of pixels is operated only for one bit data hiding. Their embedding rates are rather low as can be seen in Table 5 and Figure 7. The proposed scheme can provide a much larger embedding capacity.

5. Conclusion

A joint encryption-RDH scheme based on integer DWT and Arnold permutation is proposed. Data is hidden in the integer DWT domain with histogram modification based method, which guarantees the high embedding capacity and safety of data hiding. Although data is embedded in the DWT domain, reversible recovery of original images has been achieved through the integer transform. Different from those traditional encryption schemes such as bitwise XOR with random streams, AES, RC4, and so forth, the encryption scheme designed in this paper is based on Arnold permutation and thus is less time consuming and more efficient. Besides, permutation will not change the value of matrix, and thus data embedded will not be lost during the decryption process. Sufficient experiments demonstrate the efficiency and validity of the proposed scheme. Adaptive embedding can be adopted for better results. Multilevel integer DWT can be adopted for an even higher embedding capacity.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

Acknowledgment

The work described in this paper was supported by the Key program of National Science Fund of Tianjin, China (Grant no. 11JCZDJC16000).

References

[1] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Security and Watermarking of Multimedia Contents III*, pp. 197–208, January 2001.

[2] A. M. Alattar, "Reversible watermark using difference expansion of triplets," in *Proceedings of International Conference on Image Processing (ICIP '03)*, vol. 1, pp. 501–504, September 2003.

[3] A. M. Alattar, "Reversible watermark using difference expansion of quads," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 3, pp. 377–380, May 2004.

[4] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.

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

[6] C.-C. Lin, W.-L. Tai, and C.-C. Chang, "Multilevel reversible data hiding based on histogram modification of difference images," *Pattern Recognition*, vol. 41, no. 12, pp. 3582–3591, 2008.

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

[8] P. Tsai, Y.-C. Hu, and H.-L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, no. 6, pp. 1129–1143, 2009.

[9] C.-H. Yang and M.-H. Tsai, "Improving histogram-based reversible data hiding by interleaving predictions," *IET Image Processing*, vol. 4, no. 4, pp. 223–234, 2010.

[10] Z. Zhao, H. Luo, Z.-M. Lu, and J.-S. Pan, "Reversible data hiding based on multilevel histogram modification and sequential recovery," *AEU-International Journal of Electronics and Communications*, vol. 65, no. 10, pp. 814–826, 2011.

[11] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 912–915, May 2003.

[12] K.-S. Kim, M.-J. Lee, H.-Y. Lee, and H.-K. Lee, "Reversible data hiding exploiting spatial correlation between sub-sampled images," *Pattern Recognition*, vol. 42, no. 11, pp. 3083–3096, 2009.

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

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

[15] D. Bouslimi, G. Coatrieux, and C. Roux, "A joint encryption/watermarking algorithm for verifying the reliability of medical images: application to echographic images," *Computer Methods and Programs in Biomedicine*, vol. 106, no. 1, pp. 47–54, 2012.

[16] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Data Forensics and Security*, vol. 8, pp. 553–562, 2013.

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

[18] X. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Data Forensics and Security*, vol. 7, pp. 826–832, 2012.

[19] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–361, 2006.

[20] V. I. Arnold and A. Avez, *Ergodic Problems of Classical Mechanics*, Mathematical Physics Monograph Series, Addison-Wesley, 1968.

[21] "Miscellaneous gray level images," http://decsai.ugr.es/cvg/dbimagenes/g512.php.

[22] USC-SIPI Image Database, http://sipi.usc.edu/database/database.php?volume=textures.