Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 935741, 11 pages
http://dx.doi.org/10.1155/2015/935741

Research Article

Reversible Integer Wavelet Transform for the Joint of Image Encryption and Watermarking

Bin Wang

Key Laboratory of Advanced Design and Intelligent Computing, Dalian University, Ministry of Education, Dalian 116622, China

Correspondence should be addressed to Bin Wang; wangbinpaper@gmail.com

Received 30 March 2015; Revised 2 June 2015; Accepted 21 June 2015

Academic Editor: Uchechukwu E. Vincent

In recent years, signal processing in the encrypted domain has attracted considerable research interest, especially embedding watermarking in encrypted image. In this work, a novel joint of image encryption and watermarking based on reversible integer wavelet transform is proposed. Firstly, the plain-image is encrypted by chaotic maps and reversible integer wavelet transform. Then the lossless watermarking is embedded in the encrypted image by reversible integer wavelet transform and histogram modification. Finally an encrypted image containing watermarking is obtained by the inverse integer wavelet transform. What is more, the original image and watermarking can be completely recovered by inverse process. Numerical experimental results and comparing with previous works show that the proposed scheme possesses higher security and embedding capacity than previous works. It is suitable for protecting the image information.

1. Introduction

Along with significant improvements in information technology, information security is attracting a great deal of attention, especially image security. It can be mainly divided into two parts: one is image encryption [1–9]; the other is image watermarking [10–16].

In image encryption, chaotic maps are usually used to encrypt image, because they have the features of ergodicity, sensitivity to initial conditions and control parameters, and so forth. In [1], Chen et al. generalized two-dimensional chaotic cat map to 3D for designing a real-time secure symmetric encryption scheme, used 3D cat map to permute the position of image pixels in the permutation stage, and employed logistic chaotic system to diffuse the permuted image in the diffusion stage. In [2], Lian et al. firstly analyzed the parameter sensitivity of standard map and compared the secret key space of standard map with that of cat map and baker map. Then an improved standard map was used to realize position permutation, while the diffusion function consisted of logistic map that was used to realize the diffusion of image. In [3], Wong et al. introduced a certain diffusion effect in the permutation stage by simple sequential add-and-shift operations. Although that led to a longer processing time in a single round, the overall encryption time was reduced as fewer rounds were required. In [5], Wang et al. proposed a system of image encryption, which was based on Lorenz chaotic system and perceptron model of neural network. A fast image encryption algorithm combined with permutation and diffusion was proposed in [6]; the image was partitioned into blocks of pixels. Then, spatiotemporal chaos was employed to shuffle the blocks and, at the same time, to change the pixel values. Meanwhile, an efficient method for generating pseudorandom numbers from spatiotemporal chaos was suggested, which further increases the encryption speed. In [7], the paper presents a novel and robust chaos-based cryptosystem for secure transmitted images and four other versions. Many tests are performed and experimental results show that the suggested cryptosystem has a high security level. In [8], a secure solution during communication and transmission of fingerprint data was proposed in the form of a novel encryption technique based on reversible hidden transform and fractional wavelet packet transform. In [9], Wang et al. proposed an encryption system for colour image, in which the R, G, and B components of a colour image affect each other. In image watermarking, the main methods can be divided into two parts: difference expansion and histogram

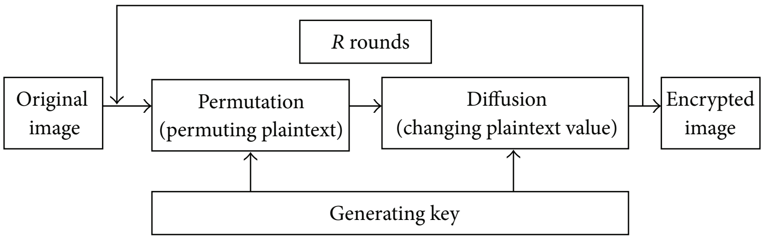

R rounds
Original image
Permutation (permuting plaintext)
Diffusion (changing plaintext value)
Encrypted image
Generating key

Figure 1: Flowchart of permutation-diffusion type of chaos-based image cryptosystems.

modification. In [10], Tian firstly proposed the difference expansion to use in reversible data embedding. The difference expansion of a generalized integer transform was proposed in [17]. In [12], Hu et al. designed a new embedding scheme to construct an efficient payload-dependent overflow location map which had good compressibility. In [13], Thodi and Rodríguez proposed a reversible data-embedding technique called prediction-error expansion. This new technique better exploited the correlation inherent in the neighborhood of a pixel than the difference expansion scheme. Recently Coltuc proposed a low-distortion transform for prediction-error expansion reversible watermarking. On the other hand, Ni et al. firstly proposed a novel reversible data hiding algorithm based on histogram modification [14]. It could recover the original image without any distortion from the marked image after the hidden data had been extracted. In [15], Tai et al. presented a reversible data hiding scheme based on histogram modification/shifting. They exploited a binary tree structure to solve the problem of communicating pairs of peak points. Recently, Wang and Yu proposed a Markov-based reversible data hiding scheme to yield high data capacity and image quality and gave two feasible approaches to reduce the overhead yielded during the data embedding [16].

As a natural idea, the technology which embeds watermarking in encrypted image is proposed. In [18], the intraprediction mode, motion vector difference, and signs of DCT coefficients were encrypted, while a watermarking was embedded into the amplitudes of DCT coefficients. Zhang proposed a novel reversible data hiding scheme for encrypted image [19]. The additional message was embedded by modifying a part of encrypted data. Hong et al. improved the work of Zhang [20]. It was significantly lower than the error rate of the cover of Zhang's work by side match. Zhang recently proposed a novel scheme for separable reversible data hiding in encrypted images [21]. In his work, the original image was encrypted using an encryption key and the additional data were embedded into the encrypted image using a data hiding key. Different receiver had different access authorization for encryption and data hiding.

However, these works have two main loopholes: without considering the permutation in encrypted stage and low embedding capacity in hiding data stage. Shannon suggested permutation and diffusion operations should be used in any cryptosystems [22]. If a cryptosystem does not have the permutation, it will be easily broken. What is more, if the permutation is used in [19, 20], they will increase the error rate of recovered image. So the first problem should be considered in the stage of encryption. For the embedding capacity, when the encrypted image and watermarking can be recovered in one piece, up to 256 bits can be embedded in encrypted test image Lena with size 512 × 512 in [19, 20]. And the embedding rate of [21] is 0.0033 bpp in test image Lena with size 512 × 512. From the above description, these works have low embedding capacity in hiding data stage.

In this paper, I propose a novel joint of image encryption and watermarking based on a reversible integer wavelet transform. In order to encrypt the image, a part of coefficient is diffused by the obits of chaotic maps. Then the inverse integer wavelet transform is used to obtain the encrypted image based on the diffused coefficient. In the stage of image watermarking, the encrypted image is processed by integer wavelet transform. Watermarking is embedded in another part of coefficient of integer wavelet transform by histogram modification. Finally an encrypted image containing watermarking is obtained by the inverse integer wavelet transform. Numerical experimental results and comparing with previous works show that the proposed scheme possesses higher security and embedding capacity than previous works. It is suitable for protecting the image information. In the part of the Discussions, another version of joint of image encryption and watermarking is presented. It can overcome the first loophole and improve visualized security of encrypted image.

The paper is organized as follows. In the next section, the related works are described in detail. In Section 3, the process of image encryption and watermarking is described in detail. Performance analyses and simulation results are reported in Section 4. In Section 5, a discussion is presented. Finally, conclusions are drawn in Section 6.

2. Related Works

In [1], the authors proposed a general cryptographic chaos-based architecture for image encryption, namely, permutation-diffusion architecture, which is shown in Figure 1. This architecture includes two iterative stages, namely, permutation stage and diffusion stage. The former permutes the plain-image but does not change the value of pixel. The latter changes the value of pixel but does not change the position of pixel. In order to improve encryption effect of algorithms, the whole permutation-diffusion round will be repeated R times.

2.1. Chaotic Maps Used in Image Encryption. In this paper, logistic chaotic map is used to encrypt image. It can be denoted as follows:

$$x_{i+1} = \mu x_i \left(1 - x_i\right). \tag{1}$$

Here, μ is control parameter for chaotic map and x_i and x_{i+1} are the ith and the $i+1$th state of chaotic map, respectively. I use two logistic maps with different parameters and initial value in this paper.

In [23], Zhang and Wang used a chosen-plaintext attack and chosen-ciphertext attack to break the logistic map based scheme. In order to improve the security of algorithm, I employ two logistic chaotic maps with the different parameters and initial values. At the same time, because of the degradation of dynamics under finite precision computations for logistic map, I use the conventional method, which is similar to previous works [24–28], to deal with the orbits of chaotic map.

2.2. Reversible Integer Wavelet Transform. Recently, reversible integer wavelet transforms are widely used in image compression and image watermarking. A reversible integer wavelet transform is used in this paper. It can be denoted as follows:

$$\begin{aligned} l &= \left\lfloor \frac{x+y}{2} \right\rfloor, \\ h &= x - y. \end{aligned} \tag{2}$$

Its inverse transform of (2) is

$$\begin{aligned} x &= l + \left\lfloor \frac{h+1}{2} \right\rfloor, \\ y &= l - \left\lfloor \frac{h}{2} \right\rfloor. \end{aligned} \tag{3}$$

The reversible integer transforms (2) and (3) are also called integer Haar wavelet transform, or the S transform. The reversible integer transforms set up a one-to-one correspondence between (x, y) and (l, h) [10]. For example, I have a pair of pixels $x = 156$ and $y = 141$. Then $l = \lfloor (156+141)/2 \rfloor = 148$ and $h = 156 - 141 = 15$ by (2). Recovering the original pixels $x = l + \lfloor (h+1)/2 \rfloor = 148 + 8 = 156$, $y = l - \lfloor h/2 \rfloor = 148 - 7 = 141$ by (3).

2.3. Preventing the Overflow and Underflow Problems. For the gray image, to prevent the overflow and underflow problems, it must restrict x, y in the range of $[0, 255]$. From (3), it is equivalent to having $0 \le l + \lfloor (h+1)/2 \rfloor \le 255$, and $0 \le l - \lfloor h/2 \rfloor \le 255$. Then, $-\lfloor (h+1)/2 \rfloor \le l \le 255 - \lfloor (h+1)/2 \rfloor$, and $\lfloor h/2 \rfloor \le l \le 255 + \lfloor h/2 \rfloor$. For the different values of h, there is

$$\begin{aligned} \left\lfloor \frac{h}{2} \right\rfloor \le l \le 255 - \left\lfloor \frac{h+1}{2} \right\rfloor, &\quad h \ge 0, \\ -\left\lfloor \frac{h+1}{2} \right\rfloor \le l \le 255 + \left\lfloor \frac{h}{2} \right\rfloor, &\quad h < 0. \end{aligned} \tag{4}$$

One can derive that the above inequalities are equivalent to

$$\begin{aligned} 0 \le l - \left\lfloor \frac{h}{2} \right\rfloor \le 255 - \left\lfloor \frac{h+1}{2} \right\rfloor - \left\lfloor \frac{h}{2} \right\rfloor, &\quad h \ge 0, \\ 0 \le l + \left\lfloor \frac{h+1}{2} \right\rfloor \le 255 + \left\lfloor \frac{h}{2} \right\rfloor + \left\lfloor \frac{h+1}{2} \right\rfloor, &\quad h < 0. \end{aligned} \tag{5}$$

Equations (4) and (5) are used to prevent the overflow and underflow problems in image encryption. As I embed a bit into the difference value h, the new difference value h' should satisfy (6) to prevent overflow and underflow [10]:

$$\left|h'\right| \le \min\left(2\left(255 - l\right), 2l + 1\right). \tag{6}$$

2.4. Histogram Modification. In this paper, the technology of histogram modification, which is proposed in Ni [14], is used to embed watermarking in encrypted image. In the histogram, I firstly find a zero point (zp for short) and a peak point (pp for short). A zp corresponds to the grayscale value which no pixel in the given image assumes. A pp corresponds to the grayscale value which the maximum number of pixels in the given image assumes. The grayscale value of pixels between zp and pp is incremented by "1." This step is equivalent to shifting the range of the histogram (zp, pp] to the right-hand side by 1 unit, leaving one the grayscale value empty. Finally, the watermarking is embedded in the pixels which are equivalent to the value of peak point. For more details please take a look on the original paper. It will not be described here.

Firstly, the watermarked image is scanned in the same sequential order as that used in the embedding procedure. If a pixel with its grayscale value, which is equivalent to pp + 1, is encountered, a bit "1" is extracted. If a pixel with its value, which is equivalent to pp, is encountered, a bit "0" is extracted. Then scanning the image again, for any pixel whose grayscale value is equivalent to the range of the histogram (zp, pp] the pixel value is subtracted by 1. If there is overhead bookkeeping information found in the extracted data, the original image can be recovered without any distortion.

3. The Process of Image Encryption and Watermarking

3.1. Encrypting Image. According to the characteristic of logistic map, logistic map is chosen as the chaotic map. The different parameters and initial values for (1) are denoted as μ_1, μ_2, $x_1(0)$, and $x_2(0)$, respectively, where $\mu_1, \mu_2 \in [3.9, 4]$ and $x_1(0), x_2(0) \in (0, 1)$. Ikey is denoted as the initial key. The detail of encryption is described as follows.

Step 1. It is randomly generating the initial key and obtaining cipher key by relating to the plain-image.

Step 2. It is evenly dividing cipher key into four parts as the parameters and initial values of logistic maps, iterating logistic maps for Q times to get rid of the transient effect, where $Q = 100$.

Step 3. It is processing the image by integer wavelet transform, namely, (2), and obtaining the transform coefficient l, h.

Step 4. It is obtaining new coefficient l' based on

$$\begin{aligned} l' &= l - \left\lfloor \frac{h}{2} \right\rfloor, \quad h \geq 0, \\ l' &= l + \left\lfloor \frac{h+1}{2} \right\rfloor, \quad h < 0. \end{aligned} \tag{7}$$

Step 5. It is diffusing the coefficient l once by the logistic chaotic orbits:

$$Cl(i) = l'(i) \oplus \text{Orbit}(i), \quad i = 1, 2, \ldots, M * \frac{N}{2}. \tag{8}$$

Step 6. It is recovering image by inverse integer wavelet transform with the diffused coefficient l and obtaining the encrypted image.

Step 7. Output the encrypted image.

Where $P(i)$ is the original image pixel value, Orbit(i) is the logistic chaotic orbit from Step 2, and $Cl(i)$ is the diffused coefficient l'. M and N are the width and height of the plain-image. $\oplus$ denotes the xor operator.

Note that implementing Step 5 is convenient to deal with the overflow and underflow problems while calculating (7). If $Cl(i)$ outs off the constraint from (5), $Cl(i)$ is processed by

$$\begin{aligned} &\bmod\left(Cl(i), 255 - \left\lfloor \frac{h+1}{2} \right\rfloor - \left\lfloor \frac{h}{2} \right\rfloor\right), \quad h \geq 0, \\ &\bmod\left(Cl(i), 255 + \left\lfloor \frac{h}{2} \right\rfloor + \left\lfloor \frac{h+1}{2} \right\rfloor\right), \quad h < 0. \end{aligned} \tag{9}$$

And the scheme saves the location of i for decrypting the cipher-image in one piece.

The detailed encryption is illustrated in Figure 2.

3.2. Embedding Watermarking. In order to encrypt image, the coefficient l of integer wavelet transform is diffused. However, the coefficient h of integer wavelet transform is not changed. So the coefficient h is used to embed watermarking by histogram modification. The details of embedding watermarking are described as follows.

Step 1. It is processing the encrypted image by integer wavelet transform, namely, (2), and obtaining the transform coefficient , h.

Step 2. It is calculating the histogram of h.

Step 3. It is embedding watermarking in h by the technology of histogram modification [14] and obtaining index code.

Step 4. It is recovering encrypted image by inverse integer wavelet transform with the changed coefficient h.

Step 5. It is obtaining the encrypted image which contains watermarking.

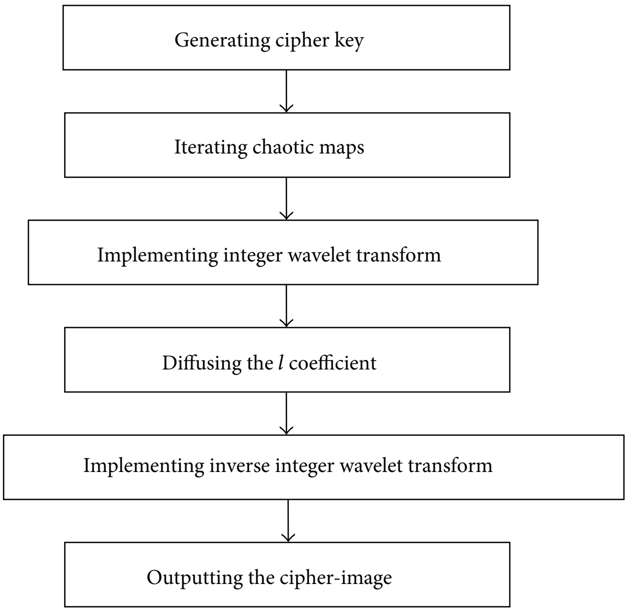

Generating cipher key
Iterating chaotic maps
Implementing integer wavelet transform
Diffusing the *l* coefficient
Implementing inverse integer wavelet transform
Outputting the cipher-image

Figure 2: The flowchart of encryption.

Note that, in order to prevent the overflow and underflow problems in embedding watermarking, index code is used in the stage. When the histogram is shifted, if the new values of h^* meet (10), it shows that the new values have overflow or underflow:

$$\left|h^*\right| > \min\left(2\left(255 - l\right), 2l + 1\right). \tag{10}$$

So these values are not shifted and mark "1" in index code. If these values are equivalent to the pp, these values are also not embedded watermarking. For the values which have been embedded in watermarking, if these values are incremented by "1" and satisfy (10), these values are marked "0" in index code.

3.3. Extracting Watermarking. The extracting process is similar to that of embedding procedure in the reversed order. It can be briefly stated as follows.

Step 1. It is implementing integer wavelet transform for the encrypted image which contains watermarking and obtaining the coefficient l, h.

Step 2. It is calculating the histogram of h.

Step 3. It is, according to the index code, zero point, and peak point, extracting the watermarking from peak point such as Ni's method [14].

Step 4. It is recovering the coefficient h without watermarking and lossless encrypted image by inverse integer wavelet transform.

Step 5. It is obtaining the encrypted image without watermarking.

In Step 3, the whole image is scanned in a sequential order, say, column-by-column, from left to right. The whole pixels which are between zero and peak point are incremented by "1." The values, which meet (10), are found out. Comparing with index code, if the values are not shifted, the corresponding index code is "1"; if the values are shifted, the corresponding index code is "0." According to the pp and above result, if the values are equivalent to pp, the extracting watermarking is "0"; if the values are equivalent to pp + 1, the extracting watermarking is "1" such as Ni's work [14].

3.4. Decrypting Image. The decryption process is similar to that of encryption procedure in the reversed order. It can be briefly stated as follows.

Step 1. It is iterating the logistic maps for Q times to get rid of the transient effect.

Step 2. It is concurrently generating the chaotic orbits as encryption process.

Step 3. It is obtaining Orbit(i).

Step 4. It is implementing integer wavelet transform and obtaining the coefficient Cl, h.

Step 5. It is recovering $l(i)$ by

$$l(i) = Cl(i) \oplus \text{Orbit}(i), \quad i = 1, 2, \ldots, M * \frac{N}{2}. \tag{11}$$

Step 6. It is implementing inverse wavelet transform by the coefficient l and h.

Step 7. It is outputting the plain-image.

4. Performance Analyses and Simulation

In this chapter, performance analyses and simulation of proposed algorithm are described in detail. Firstly, the experimental results of image encryption are presented, such as the space of key and key sensitivity. Then the effect of image watermarking is provided, such as embedding capacity and peak signal-to-noise ratio (PSNR for short).

4.1. Performance Analyses and Simulation in Image Encryption

4.1.1. The Space of Key. In a good image cryptosystem, the space of key should be large enough to make brute-force attack infeasible [29]. In this proposed architecture, the Ikey consists of 16 elements, namely, Ikey $= \{x_i\}$, $i = 1, 2, \ldots, 16$, $x_i \in [0, 255]$. So the key space of the proposed architecture is equivalent to $2^{128} \approx 3.4 \times 10^{38}$, which is sufficiently large to meet the need for practical application.

4.1.2. Key Sensitivity. In this part, the key sensitivity will be performed as follows.

Step 1. It is calculating the Imkey of the standard test 512 × 512 image Lena.

Step 2. It is encrypting the test image by Ikey 987654321012345.

Step 3. It is slightly changing the generated Ikey 987654321012346, and encrypt the same plain-image.

Step 4. It is comparing the cipher-image which is encrypted by different key.

The results are as follows: the image encrypted by the key 987654321012345 has 99.32% of difference from the image encrypted by the key 987654321012346 in terms of pixel values, although there is only one-bit difference in the two keys. Figure 3 shows the test result. Moreover, when a key is used to encrypt an image while another trivially changed key is used to decrypt the ciphered image, the decryption also completely fails.

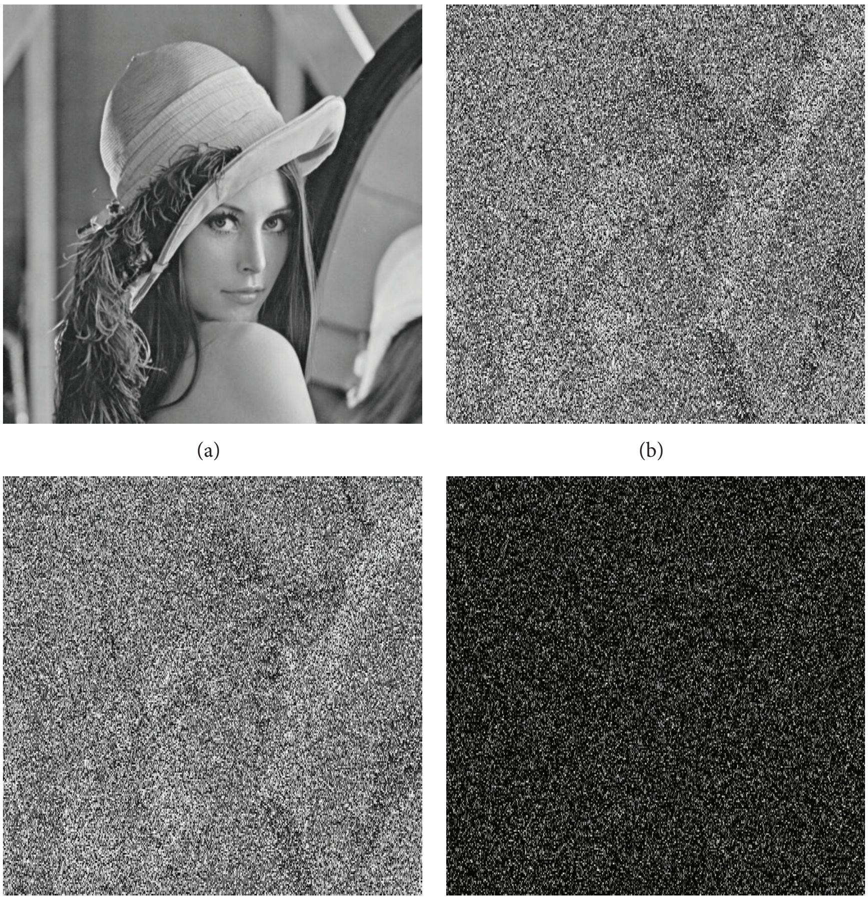
In image encryption, mutual information is usually used to make the quantitative analysis and evaluate the key sensitivity of two ciphered images [29]. Note that the ciphered images must be encrypted by different secret keys (slightly change) on the same plaintext image. More details can be referred to in [29]. In Figure 3, the mutual information of (b) and (c) is 0.2211. It shows that the proposed algorithm has the higher key sensitivity of secret keys.

4.1.3. Statistical Analysis. In this proposed architecture, the standard Lena test image of size 512 × 512 is selected to test the property of resisting statistical analysis. The histograms of encrypted image are shown in Figure 4. From the figure, one can see that the histogram of the ciphered image is fairly uniform and is significantly different from that of the original image. To test the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, and two diagonally adjacent pixels, respectively, in a ciphered image, the following procedure is carried out. First, randomly select 1000 pairs of two adjacent pixels from an image. Then, calculate the correlation coefficient of each pair by using the following two formulas [1]:

$$\begin{aligned} \operatorname{cov}(x, y) &= E\{(x - E(x))(y - E(y))\}, \\ r_{xy} &= \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \end{aligned} \tag{12}$$

where x and y are greyscale values of two adjacent pixels in the image. In numerical computation, the following discrete formulas are employed:

$$\begin{aligned} E(x) &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\ D(x) &= \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2, \\ \operatorname{cov}(x, y) &= \frac{1}{N}\sum_{i=1}^{N} \{(x_i - E(x))(y_i - E(y))\}. \end{aligned} \tag{13}$$



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

Figure 3: (a) Plain-image of Lena. (b) Encrypted image by key: 987654321012345. (c) Encrypted image by key: 987654321012346. (d) Difference image.

Figure 5 shows the correlation distribution of two vertically adjacent pixels in the plain-image and that in the cipher-image: the correlation coefficients are 0.9669 and 0.0206, respectively, which are far apart.

4.1.4. Differential Attack. To test the property of resisting differential attack of the proposed architecture, two common quantitative criteria are employed: number of pixels' change rate (NPCR) and unified average changing intensity (UACI). The NPCR and UACI are defined as follows [30, 31]:

$$\mathrm{NPCR} = \frac{\sum_{i,j} D\left(i, j\right)}{W \times H} \times 100\%,$$
$$\mathrm{UACI} = \frac{1}{W \times H}\left[\sum_{i,j} \frac{\left|C_1\left(i, j - C_2\left(i, j\right)\right)\right|}{255}\right] \times 100\%, \tag{14}$$

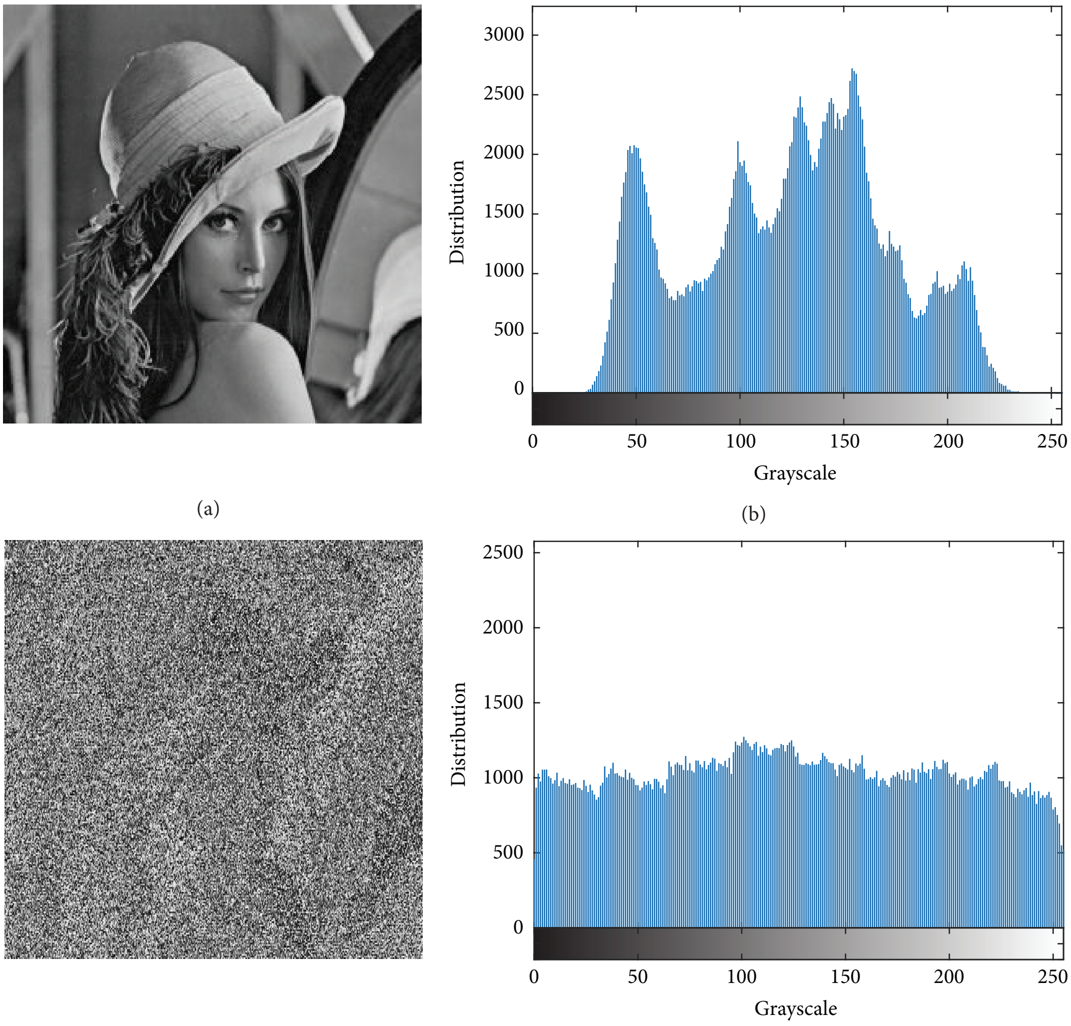
where C_1 and C_2 are the two cipher-images whose corresponding Ikeys have only difference and the greyscale values of the pixels at poison (i, j) of C_1 and C_2 are denoted as $C_1(i, j)$ and $C_2(i, j)$, respectively; W and H are the width and height of the cipher-image; $D(i, j)$ is determined by $C_1(i, j)$ and $C_2(i, j)$; namely, if $C_1(i, j) = C_2(i, j)$ then $D(i, j) = 0$; otherwise, $D(i, j) = 1$.

Table 1: Comparing results.

	NPCR	UACI
Proposed scheme	99.96%	29.74%
Wang's work	44.27%	14.874%
Gupta's work	99.62%	17.30%
Teng's work	93.68%	33.34%

In Table 1, I compare our results of NPCR with Wang et al. [4], Gupta and Silakari [32], and Teng and Wang [33] works. The whole round is repeated only once. Our results are the average of ten trials. Comparing with previous works, it suggests that our scheme has higher security.

4.1.5. Resistance to Known-Plaintext and Chosen-Plaintext Attacks. In the proposed scheme of image encryption, this scheme uses different parameters and values of chaotic maps which are used in the diffusion stage. At the same time, it makes the cipher key related to the plain-image. So the



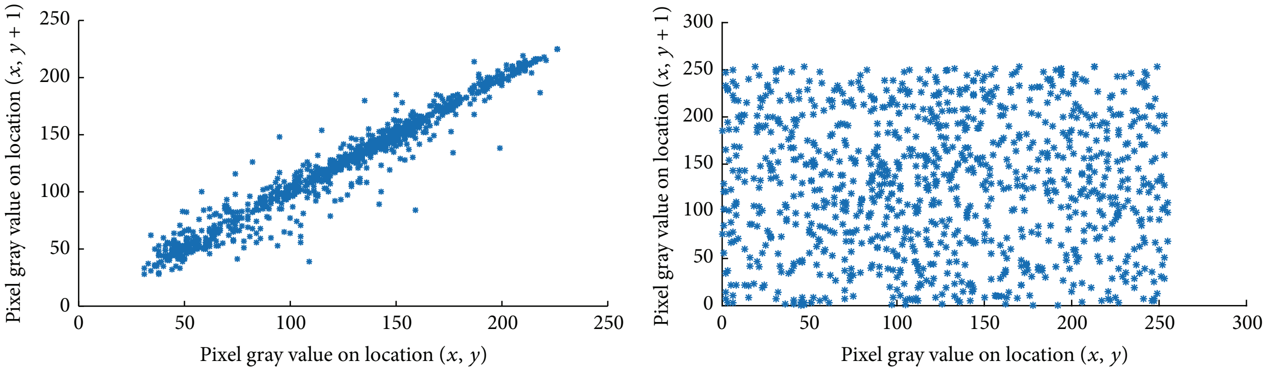
(a)

(b)

(c)

(d)

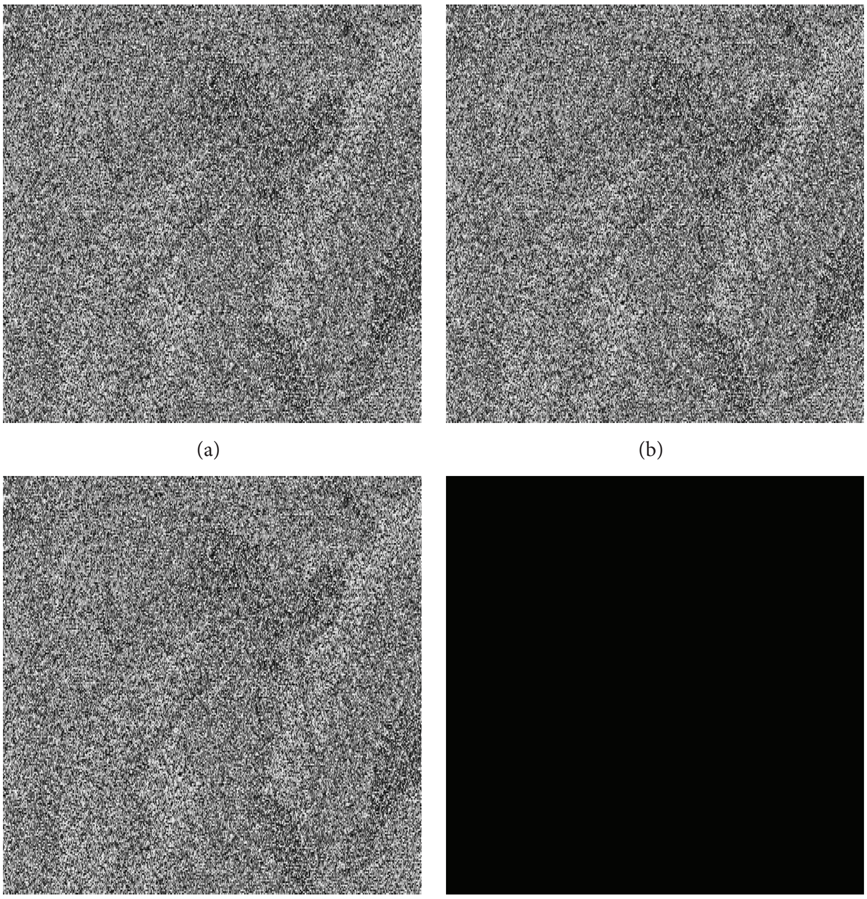
FIGURE 4: (a) Plain-image of Lena. (b) Histogram of the plain-image. (c) Cipher-image. (d) Histogram of the cipher-image.



(a)

(b)

FIGURE 5: (a) The correlation of vertical adjacent two pixels for original image. (b) The correlation of vertical adjacent two pixels for cipher-image.



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

Figure 6: (a) Encrypted image of Lena. (b) Encrypted image with watermarking. (c) Encrypted image after extracting watermarking. (d) Difference image of (a) and (c).

different control conditions, key streams, and nonidentical cipher-images will be generated by distinct plain-images. The attacker cannot obtain useful information by encrypting some special images since the resultant information is only related to those chosen-images. Therefore, the proposed algorithm can well resist the known-plaintext and the chosen-plaintext attacks.

4.2. Performance Analyses and Simulation in Image Watermarking

4.2.1. Embedding Watermarking. The standard test 512 × 512 image Lena is firstly encrypted by above scheme. Then the encrypted image is employed to embed watermarking. The effects of embedding watermarking, extracting watermarking, and difference are shown in Figure 6.

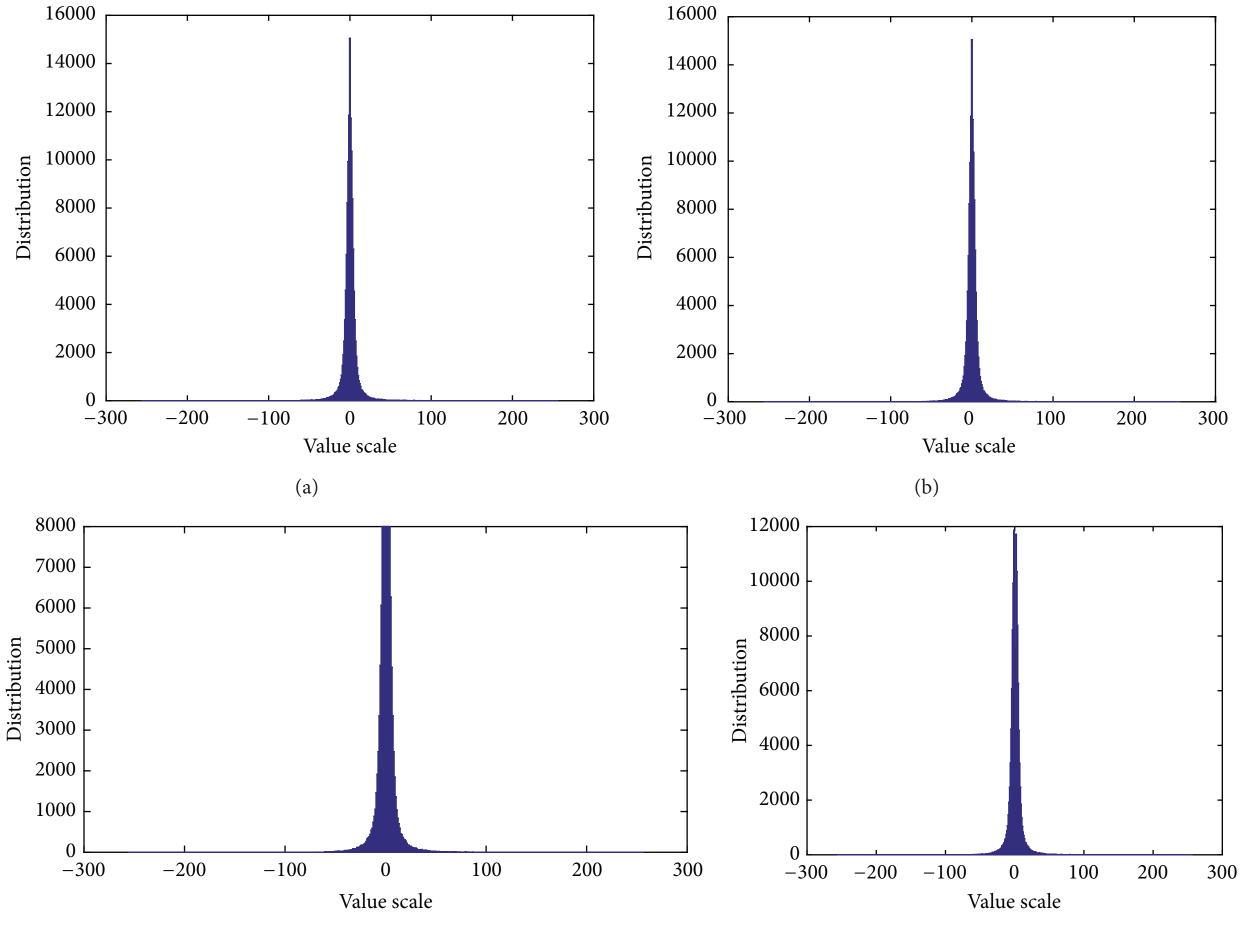
Because Ni's method is a reversible data hiding, the encrypted image and watermarking can be recovered in one piece. Watermarked image quality is evaluated by peak signal-to-noise ratio (PSNR), which is defined as

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{255^2}{\mathrm{MSE}}, \tag{15}$$

where $\mathrm{MSE} = (1/M \times N) \sum (X_i - X_i')^2$; M and N are image width and length, respectively; and X_i and X_i' are pixel value of the original image and watermarked image, respectively. The PSNR of Figure 6 is 54.16. The embedding rate is 0.057 bpp. However, if the encrypted image and watermarking can be recovered in one piece, the most embedding bits are only 256 bits, namely, 0.001 bpp, in [19, 20]. And the PSNR is 37.9 dB. These indexes are far less than that of proposed method. In Zhang's work [21], if the encrypted image and watermarking can be recovered in one piece, the most embedding rate is 0.033. What is more, PSNR is only 38.0. They are less than that of proposed method.

Figure 7 shows the values of coefficient h at different phases of proposed method. Comparing Figures 7(a) and 7(b), encrypting image cannot effect the value of coefficient h. In this figure, the values, which meet $h = 0$, are used as peak point; the values, which meet $h = 84$, are used as zero point. From Figure 7(c), the grayscale value of pixels between zp and pp is incremented by "1." Then a gap appears in $h = 1$. It is used to embed watermarking.

4.2.2. Index Code. In the third chapter, I briefly introduce index code in image watermarking. Index code will be



(a)

(b)

(c)

(d)

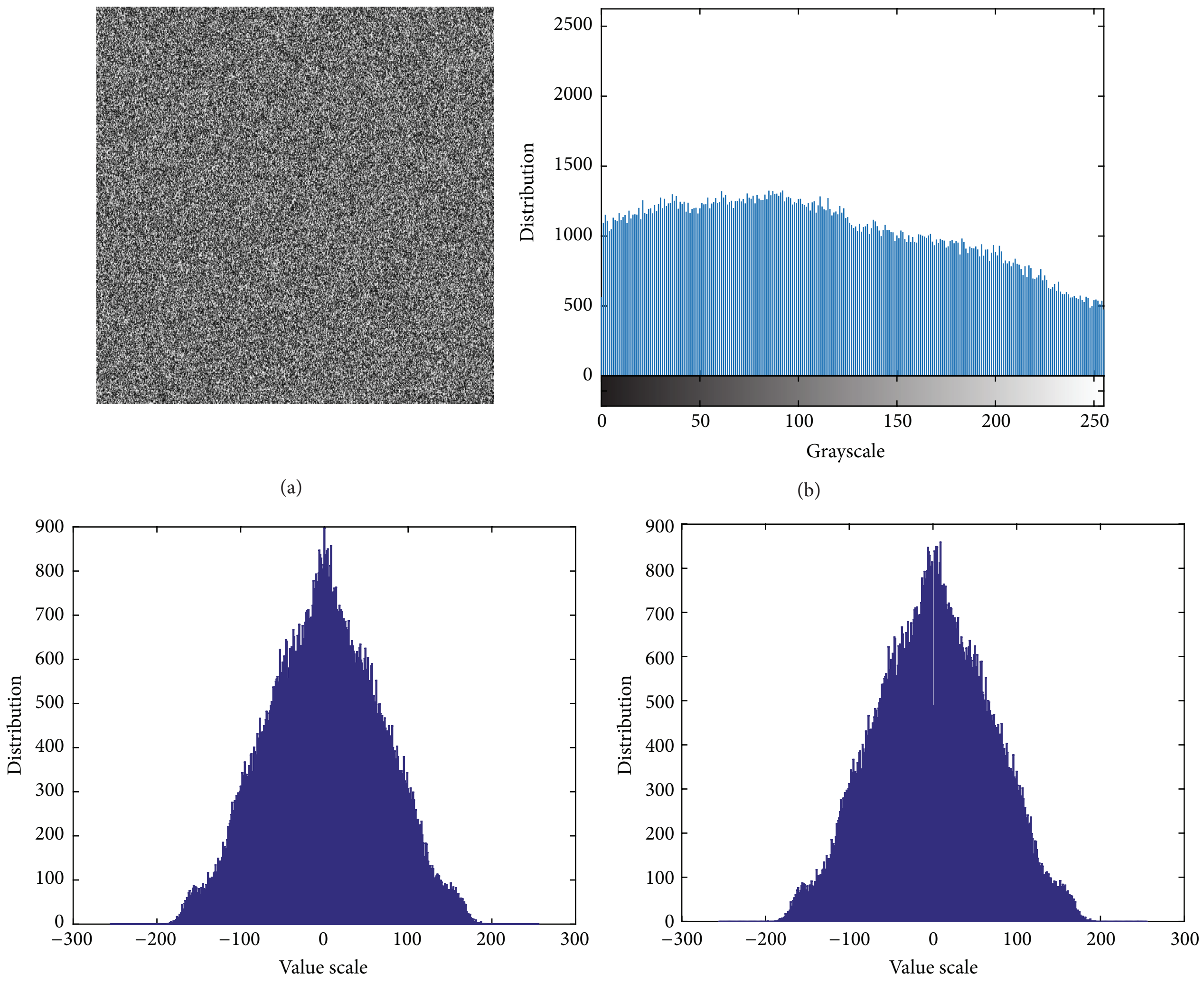
Figure 7: (a) Original image; (b) encrypted image; (c) after shifting operation; (d) after embedding watermarking.

described by example in this part. For Figure 6(a), there are two pairs of pixels, $(255, 251)$ and $(255, 250)$. Using (2), I obtain their coefficients, $(253, 4)$ and $(252, 5)$, respectively. Because of shifting operation, the value of coefficient h between $\mathrm{zp} = 84$ and $\mathrm{pp} = 0$ is incremented by "1." However, for the first pair, $h^* = 4 + 1 = 5$, $\min(2(255 - l), 2l + 1) = \min(4, 507) = 4$, it meets (10). So it cannot be incremented, and the index code is "1." For the second pair, $h^* = 5 + 1 = 6$, $\min(2(255 - l), 2l + 1) = \min(6, 505) = 6$, it can be incremented. After shifting operation, it is changed into $(252, 6)$. If it is incremented once again, $h^* = 6 + 1 = 7$, $\min(2(255-l), 2l+1) = 6$, it meets (10). So its index code is "0." Then after embedding watermarking, all the values, which meet (10), are corresponding to a unique index code. If the index code is "1," it shows that this value is not incremented; if the index code is "0," it shows that this value is incremented or embedded watermarking. The length of index code is only 445 bits.

5. Discussions

The above method only uses diffusion operation in the stage of image encryption. As I describe in Section 1, the previous works do not consider the permutation operation in the stage of image encryption. It will be easily broken and increase the error rate of recovered image. At the same time, there is a flaw in proposed method. A fuzzy outline of plain-image can be seen after encrypting image. To overcome these flaws, the permutation-diffusion architecture is used to encrypt plain-image in the stage of image encryption, namely, adding permutation operation. The plain-image is firstly permuted by the order from chaotic maps. Then the permuted image is processed by proposed method.

Figure 8(a) is the encrypted image by the permutation-diffusion architecture. Seeing Figure 8(a), the fuzzy outline of plain-image has disappeared, and the visualized security of encrypted image is improved. Figure 8(b) is its histogram. Figure 8(c) is the histogram of values of coefficient h of Figure 8(a). Figure 8(d) is the histogram of values of coefficient h after embedding watermarking. The values of NPCR and UACI did not change significantly. However, when the encrypted image and watermarking can be recovered in one piece, the embedding capacity is decreased significantly. In last section, the embedding capacity is 14985 bits, namely, 0.057 bpp. In this section, the embedding capacity is only 893 bits, namely, 0.0034 bpp. Although it is more than that of



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

Figure 8: (a) Encrypted image. (b) Histogram of the encrypted image. (c) Values of coefficient h of encrypted image. (d) Values of coefficient h of embedding watermarking.

[20, 34], it is less than that of last section. The PSNR is also 54.16 dB.

6. Conclusions

To overcome the flaws of previous works, I propose a novel joint of image encryption and watermarking based on a reversible integer wavelet transform in this paper. In pursuit of high embedding capacity at the expense of security, diffusion operation is only used to encrypt image in the stage of image encryption, namely, diffusing the coefficient l. To pursue high security of encrypted image at the expense of embedding capacity, permutation-diffusion architecture is employed to encrypt image in the stage of image encryption. Then, in the stage of image watermarking, the encrypted image is processed by integer wavelet transform. Watermarking is embedded in another part of coefficient of integer wavelet transform by histogram modification. Finally an encrypted image containing watermarking is obtained by the inverse integer wavelet transform. Numerical experimental results and comparing with previous works show that the proposed scheme possesses higher security and embedding capacity than previous works. It is suitable for protecting the image information.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 61370005, 61402066, and 61402067), and the project is sponsored by “Liaoning BaiQianWan Talents Program” (no. 2013921007) and the Basic Research

Program of the Key Lab in Liaoning Province Educational Department (nos. LZ2014049 and LZ2015004); the project is supported by the Natural Science Foundation of Liaoning Province (no. 2014020132), Scientific Research Fund of Liaoning Provincial Education Department (no. L2014499), and the Program for Liaoning Key Lab of Intelligent Information Processing and Network Technology in University.

References

[1] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[2] S. Lian, J. Sun, and Z. Wang, "A block cipher based on a suitable use of the chaotic standard map," *Chaos, Solitons & Fractals*, vol. 26, no. 1, pp. 117–129, 2005.

[3] K.-W. Wong, B. S.-H. Kwok, and W.-S. Law, "A fast image encryption scheme based on chaotic standard map," *Physics Letters, Section A: General, Atomic and Solid State Physics*, vol. 372, no. 15, pp. 2645–2652, 2008.

[4] Y. Wang, K.-W. Wong, X. Liao, T. Xiang, and G. Chen, "A chaos-based image encryption algorithm with variable control parameters," *Chaos, Solitons & Fractals*, vol. 41, no. 4, pp. 1773–1783, 2009.

[5] X.-Y. Wang, L. Yang, R. Liu, and A. Kadir, "A chaotic image encryption algorithm based on perceptron model," *Nonlinear Dynamics*, vol. 62, no. 3, pp. 615–621, 2010.

[6] Y. Wang, K.-W. Wong, X. Liao, and G. Chen, "A new chaos-based fast image encryption algorithm," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 514–522, 2011.

[7] A. Awad, "A new chaos-based cryptosystem for secure transmitted images," *IEEE Transactions on Computers*, 2011.

[8] G. Bhatnagar and Q. M. J. Wu, "Chaos-based security solution for fingerprint data during communication and transmission," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 4, pp. 876–887, 2012.

[9] X. Y. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Processing*, vol. 92, no. 4, pp. 1101–1108, 2012.

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

[11] 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.

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

[13] D. M. Thodi and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.

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

[15] 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.

[16] C.-T. Wang and H.-F. Yu, "A Markov-based reversible data hiding method based on histogram shifting," *Journal of Visual Communication and Image Representation*, vol. 23, no. 5, pp. 798–811, 2012.

[17] 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.

[18] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 6, pp. 774–778, 2007.

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

[20] 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.

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

[22] C. E. Shannon, "Communication theory of secrecy systems," *MD Computing*, vol. 15, no. 1, pp. 57–64, 1998.

[23] Y.-Q. Zhang and X.-Y. Wang, "Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation," *Nonlinear Dynamics*, vol. 77, no. 3, pp. 687–698, 2014.

[24] H. J. Liu and X. Y. Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Computers & Mathematics with Applications*, vol. 59, no. 10, pp. 3320–3327, 2010.

[25] H. J. Liu and X. Y. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Optics Communications*, vol. 284, no. 16-17, pp. 3895–3903, 2011.

[26] B. Wang, X. P. Wei, and Q. Zhang, "A novel and fast chaotic cryptosystem for image encryption," *Journal of Computational and Theoretical Nanoscience*, vol. 11, no. 3, pp. 731–738, 2014.

[27] B. Wang, X. Zheng, S. Zhou et al., "Encrypting the compressed image by chaotic map and arithmetic coding," *Optik*, vol. 125, no. 20, pp. 6117–6122, 2014.

[28] X. P. Wei, B. Wang, Q. Zhang, and C. Che, "Image encryption based on chaotic map and reversible integer wavelet transform," *Journal of Electrical Engineering*, vol. 65, no. 2, pp. 90–96, 2014.

[29] Y.-Q. Zhang and X.-Y. Wang, "A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice," *Information Sciences*, vol. 273, pp. 329–351, 2014.

[30] H. Hermassi, R. Rhouma, and S. Belghith, "Security analysis of image cryptosystems only or partially based on a chaotic permutation," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2133–2144, 2012.

[31] L. Zhao, A. Adhikari, D. Xiao, and K. Sakurai, "On the security analysis of an image scrambling encryption of pixel bit and its improved scheme based on self-correlation encryption," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 8, pp. 3303–3327, 2012.

[32] K. Gupta and S. Silakari, "Novel approach for fast compressed hybrid color image cryptosystem," *Advances in Engineering Software*, vol. 49, no. 1, pp. 29–42, 2012.

[33] L. Teng and X. Wang, "A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive," *Optics Communications*, vol. 285, no. 20, pp. 4048–4054, 2012.

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