Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 4313580, 11 pages
http://dx.doi.org/10.1155/2016/4313580

Research Article

A New Reversible Date-Hiding Algorithm for Encrypted Images

Laicheng Cao and Hao Zhou

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

Correspondence should be addressed to Laicheng Cao; caolaicheng@163.com

Received 26 February 2016; Accepted 3 August 2016

Academic Editor: Haipeng Peng

In order to effectively increase embedding capacity and completely extract the watermarking information in information hiding of encrypted images, a new reversible watermarking embedding algorithm based on rhombus prediction model and difference histogram shifting ideas is proposed. Firstly, the images are pretreated according to rhombus prediction model. Then, the watermarking information is embedded in encrypted images by effective combination of homomorphism encryption scheme and reversible watermarking techniques. Finally, the watermarking information is completely extracted and the images are recovered based on computed difference histogram from left to right and from top to bottom. So, the efficiency and reversibility are ensured when watermarking information is embedded in encrypted image. Experiment results show that the proposed algorithm is simple and easy to realize, the embedding capacity is effectively increased, watermarking information is completely reversible, and the image can be recovered with no distortion.

1. Introduction

With the advent of the digital age, media content gradually changes from analog to digital. Digital technology makes the multimedia data (image, video, and text) storage, replication, and communication become very convenient. Therefore, how to implement effective copyright protection and information security measures in the network environment has become an urgent realistic problem.

At present, there are three types of reversible watermarking methods: (1) compression based reversible watermarking [1–3]; (2) expansion based reversible watermarking [4–6]; (3) histogram modification based reversible watermarking [7–9].

Encryption technology [10] and digital watermarking [11] are all with different maintenance information security function. So, they are often used together. In traditional way, the watermarking information is embedded in the multimedia works, and then the multimedia works that contain the watermarking information are encrypted. However, in some occasions, we must first encrypt the image and then embed watermarking. So there is the digital watermarking in encryption images. Reference [12] has proposed the corresponding algorithm, but the algorithm is all irreversible. However, in the military, medicine, and so on, the fidelity of the occasion possesses high demand, and any distortion of the multimedia data is not allowed. So, the encryption technique of reversible watermarking is developed. Zhang et al. [13] have truly realized the combination of encryption and watermarking. They embed watermarking by flipping the cipher image pixels' three least significant bits (LSBs), and then the same operation is performed in the decrypted image. Using the correlation between the nature image spaces, it can extract watermarking information. But the amount of data and image watermarking extraction rate is not optimistic, especially when the block is smaller and the error rate is higher. Hong et al. [14] have proposed an algorithm of reversible data hiding in encrypted images by fully considering the images edge pixel and using edge match technology, so that the extraction accuracy is increased some. However, errors still exist in the restored image, and algorithm is not reversible. Zhang et al. have put forward a new idea [15, 16] that uses the method of matrix calculation compressing the encrypted images LSB (least significant bit) to make the room for hiding information. In the receiver, the data extraction and image restoration can be separated. However, watermarking extraction still uses the characteristics of the relationship of the nature image pixel and so does not guarantee the ability to completely extract the watermarking

information. Reference [17] has proposed reserving room before encryption. However, this method is not handled in the encrypted domain, because it is the original place to fill the data. Zheng et al. [18] have proposed using wonton scrambling to image encryption, in the cipher image; based on histogram shift algorithm, watermarking information is embedded in the cipher image. Reference [19] has proposed hiding algorithm based on cipher image lossless compression, but the watermarking extraction is still using the relationship of the nature image pixels characteristics, so it dose not guarantee the ability to completely extract the watermarking information. Others like [20–25] are also about data hiding in encrypted images.

According to the problem above, we propose a new reversible watermarking embedding algorithm in encrypted image. Figure 1 gives a sketch of reversible data hiding in encrypted image. The image is first pretreated and encryption key is embedded before the image is encrypted, then the sender encrypts image by this encryption key, the administrator embeds addition bits through this embedded encryption key, and finally the receiver can get the encrypted image with embedded bits. In the same ways, when the watermarking is extracted and the image is recovered, the encrypted image is first preoperated with embedded bits, and then embedded encryption key is gotten, and finally the receiver decrypts image through decryption key and gets the decrypted image with embedded bits; accordingly legitimate watermarking information application can extract encryption key and addition bits, and it can also recover image.

Our reversible watermarking embedding algorithm is based on rhombus prediction model and difference histogram shifting ideas; it can effectively increase embedding capacity and completely extract the watermarking information in information hiding of encrypted images while the efficiency and reversibility can be ensured. In order to achieve this innovation, we first pretreat the images according to rhombus prediction model, and then we embed the watermarking information in encrypted images by effective combination of homomorphism encryption scheme and reversible watermarking techniques; the watermarking information can be completely extracted and the images are recovered based on computed difference histogram from left to right and from top to bottom.

The remainder of this paper is organized as follows. In Section 2, the theory of homomorphic encryption and the information entropy of the encrypted image are analyzed. The proposed algorithm is presented in Section 3. Section 4 shows our experimental results and analysis. Finally, in Section 5, we summarize our results and present the conclusion.

2. Theory Analysis

2.1. Homomorphic Encryption. The characteristics of homomorphic encryption are for some operations in plaintext and then encryption effect is equivalent to some operation directly in the ciphertext [26–30]. The general definition of homomorphic encryption is

$$E\left(m_1 \otimes_M m_2\right) \longleftarrow E\left(m_1\right) \oplus_C E\left(m_2\right), \quad \forall m_1, m_2 \in M, \quad (1)$$

where $\otimes_M$ represents the operation in the plaintext M, $\oplus_C$ represents the operation in the ciphertext C, and "$\leftarrow$" represents "homomorphic encryption can be directly obtained by calculating from the ciphertext"; namely, it does not exist in the middle of the decryption process.

But the encryption methods and the corresponding homomorphism (addition, subtraction, multiplication, and division) must satisfy the image encryption. In order to make the image encryption have a low complexity, we choose the RC4 encryption [30]; thus the corresponding homomorphism satisfies the additive homomorphic, and the encryption mechanism is specified as follows.

Record M is a gray image pixel in plaintext, a key seed S is selected to generate RC4 random key stream K that encrypts per pixel in M, and then the encrypted image C is gotten. In the following formula, $E(\,)$ represents encryption operation and $D(\,)$ represents decryption operation.

The encryption method is shown as follows:

$$\begin{aligned} C &= E\left(M, K\right) = \left(M + K\right) \bmod 256 \\ &= \left(m_i + k_i\right) \bmod 256 = c_i, \quad \forall i = 1, 2, \ldots, L, \end{aligned} \quad (2)$$

where L is the number of pixels in the image and m_i, k_i, c_i, respectively, show the ith plaintext pixel, the key stream of random number, and the ciphertext pixel.

The decryption method is shown as follows:

$$D\left(C, K\right) = \left(c_i - k_i\right) \bmod 256 = m_i \left(\bmod 256\right), \quad \forall i = 1, 2, \ldots, L. \quad (3)$$

Suppose that m_1, m_2 are, respectively, for two different pixels in gray image and k_1, k_2 are the random numbers from the key stream used to encrypt m_1, m_2. According to definition (1), if $\otimes_M$ is the arithmetic plus, $\oplus_C$ is the modular addition; then

$$\begin{aligned} E\left(m_1 \otimes_M m_2, k_1 + k_2\right) &= E\left(m_1 + m_2, k_1 + k_2\right) \\ &= \left(m_1 + m_2 + k_1 + k_2\right) \bmod 256 \\ &= \left(\left(m_1 + k_1\right) \bmod 256 + \left(m_2 + k_2\right) \bmod 256\right) \\ &\cdot \bmod 256 = \left(E\left(m_1, k_1\right) + E\left(m_2, k_2\right)\right) \bmod 256 \\ &= E\left(m_1, k_1\right) \oplus_C \left(m_2, k_2\right). \end{aligned} \quad (4)$$

So, this encryption method satisfies the additive homomorphism. Correspondingly, use $k_1 + k_2$ to description $E(m_1, k_1) \oplus_C E(m_2 + k_2)$, which is

$$\begin{aligned} &D\left(E\left(m_1, k_1\right) \oplus_C E\left(m_2, k_2\right), k_1 + k_2\right) \\ &= D\left(E\left(m_1 \otimes_M m_2, k_1 + k_2\right), k_1 + k_2\right) \\ &= \left(m_1 \otimes_M m_2\right) \left(\bmod 256\right) = \left(m_1 + m_2\right) \left(\bmod 256\right). \end{aligned} \quad (5)$$

The security of this encryption mechanism depends on the use of the stream cipher, and RC4 is a mature encryption mechanism [30].

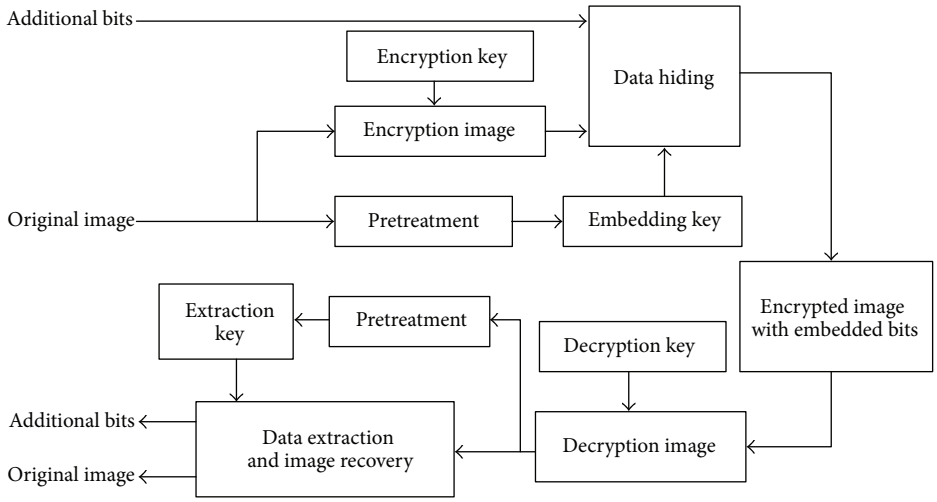


Figure 1: Sketch of reversible data hiding in encrypted image.

2.2. Analysis Information Entropy of the Encrypted Image. The information entropy is used to measure the amount of information lost in the signal transmission process before it is received. After the image is encrypted, if the chaos degree of its pixels reaches maximum, then the image information entropy tends to the maximum value. Reference [31] has proposed a general parser, and the encrypted signal bit stream is divided into a series of nonoverlapping fixed length fields, called the fictional codeword. Supposing that the length of each fictional codeword is L bits and the total length of the encryption signals is N bits, so the total number of fictional codewords C_L is calculated, which is

$$C_L = \left\lceil \frac{N}{L} \right\rceil. \tag{6}$$

The generic parser is a partition function $D(X, L)$, and it will divide X into nonoverlapping ordered tuples and each tuple contains L symbols in X, which is

$$D(X, L) = X_L = \left\{ T_1^L, T_2^L, \ldots, T_{C_L}^L \right\}. \tag{7}$$

Below, we give the definition of the information entropy of the X_L dependent on L.

Suppose that $X = \{X_1, X_2, \ldots, X_N\}$ is a collection of discrete uniform distribution of elements, the probability of each element satisfies $P(X_1) = P\{X_2\} = \cdots = P\{X_N\} = 1/N$, then a genera parser segmentation X is shown as follows:

$$\begin{aligned} X_L &= D(X, L), \\ X_{L+1} &= D(X, L+1). \end{aligned} \tag{8}$$

So, for $\forall T_i^L \in X_L$ and $\forall T_i^{L+1} \in T_{L+1}$, the information entropy of tuple T_i^L and T_i^{L+1} are $I(T_i^L)$ and $I(T_i^{L+1})$ satisfying the follow formula, which is

$$I\left(T_i^L\right) > I\left(T_i^{L+1}\right). \tag{9}$$

Equation (9) shows that the sum of self information of L should greater than the length of the self information of $L+1$.

That is to say, the information entropy of X_L is greater than X_{L+1}, which is shown as follows.

Suppose that a general parser divides X into X_L and X_{L+1}, such as (8), so,

$$H(X_L) > H(X_{L+1}), \tag{10}$$

where X is a discrete uniform distribution of symbol set and $H(X_L)$ and $H(X_{L+1})$ are, respectively, the information entropy for a set of tuples of X_L and X_{L+1}, $1 \le L \le N$.

Proof. According to the definition in [32], the initial $H(X_L)$ is

$$H(X_L) = \frac{1}{L} \sum_{i=1}^{C_L} p\left(T_i^L\right) \times \theta\left(T_i^L\right), \tag{11}$$

where $p(T_i^L)$ is the probability of tuple T_i^L in X_L, $\theta(T_i^L)$ is the codeword length that the encoded tuple T_i^L needs, because X is a discrete uniform distribution, $p(T_i^L)$ and $\theta(T_i^L)$ for $\forall T_i^L$ are constant, which are

$$\begin{aligned} p\left(T_i^L\right) &= \frac{1}{C_L}, \\ \theta\left(T_i^L\right) &= \log_2(C_L). \end{aligned} \tag{12}$$

So, (11) can be written as follows:

$$H(X_L) = \frac{C_L \times \log_2(C_L)}{L \times C_L} = \frac{\log_2(C_L)}{L}, \tag{13}$$

because

$$\begin{aligned} &\log_2(C_L) > \log_2(C_{L+1}) \Longrightarrow \\ &\frac{\log_2(C_L)}{L} > \frac{\log_2(C_{L+1})}{L} \Longrightarrow \\ &\frac{\log_2(C_{L+1})}{L} > \frac{\log_2(C_{L+1})}{L+1}. \end{aligned} \tag{14}$$

So

$$\frac{\log_2(C_L)}{L} > \frac{\log_2(C_{L+1})}{L+1}. \quad (15)$$

According to (13), (15) can be rewritten as follows:

$$H(X_L) > H(X_{L+1}). \quad (16)$$

Because X is fixed, at least it can eliminate redundancy in the parsed signal through the information entropy coding method. Namely, the encrypted image information still exists in redundancy, which means the encrypted image still exists in the information entropy that can be used, and we call it the information entropy difference. In order to make better use of this part of the information entropy difference, we propose a rhombus prediction scheme and use the difference histogram shifting scheme. □

3. The Proposed Algorithm

In the proposed algorithm, the sender will encrypt image and embed data, but the encryption section should be divided into two parts, which are pretreatment phase and encryption phase. When the receiver gets an encrypted image containing embedded data, he can decrypt image, extract the additional data, and recover the original image.

3.1. The Sender. This section is divided into three parts: part (1) is image pretreatment; part (2) is image encryption; part (3) is embedding watermarking information.

(1) Image Pretreatment. The image is pretreated before it is encrypted; Figure 2 shows the pixel division scheme in the rhombus prediction. Firstly, the pixels of the image are divided into two categories according to Figures 2(a) or 2(b), and white dots and black dots represent the two types of divided pixels. In Figure 2(a), the four black pixels adjacent to the white pixels form a rhombus pattern, the four black pixels are located in the four corners of the rhombus, and the white pixels are located in the center of the rhombus, and the middle white pixel value can be predicted by using the four black pixel values. In Figure 2(b), the four white pixels are located in the four corners of the rhombus, and the black pixels are located in the center of the rhombus; the middle black pixel value can be predicted by using the four white pixel values. In this paper, we use Figure 2(a) to divide the pixels. For the all white pixels $I_{i,j}$ in the intermediate rhombus, through its four adjacent black points, we can get corresponding prediction value $P_{i,j}$, which is

$$P_{i,j} = \frac{I_{i-1,j} + I_{i+1,j} + I_{i,j-1} + I_{i,j+1}}{4}, \quad (17)$$

where $I_{i-1,j}$, $I_{i+1,j}$, $I_{i,j-1}$, and $I_{i,j+1}$ are four adjacent pixels (up, down, left, and right) of the $I_{i,j}$. watermarking embedding is only to operate on the $I_{i,j}$, so the values of $I_{i-1,j}$, $I_{i+1,j}$, $I_{i,j-1}$, and $I_{i,j+1}$ are not changed; namely, the value of $P_{i,j}$ is not changed. Therefore, we can restore image according to the lowest effective bit of $P_{i,j}$. Then, we can get the corresponding prediction error, which is

$$e_{i,j} = P_{i,j} - I_{i,j}. \quad (18)$$

In (17), we compute all white pixel corresponding prediction error values $e_{i,j}$ that satisfy the condition, generate histogram $H(e_{i,j})$, and find the difference error, which are the top of the difference errors e_l and e_r ($e_r = e_l + 1$).

Then, the predicted value $e_{i,j}$ of all pixels $I_{i,j}$ is divided into two categories according to Figure 3. The error value is in the middle of the diamond and the error value is in the diamond-four angle. Each type of histogram is generated according to the following method. Firstly, the histogram is pretreated, secondly, the watermarking information is embedded in it, and finally, the histogram is encrypted by the RC4 encryption.

We compute all white pixel corresponding error $e_{i,j}$ that satisfy the condition and generate two parts of histograms $H(e_{i,j})$ of the two categories difference; then, the same operation is performed in each histogram. we find the maximum vertex in the the histogram of the difference between the corresponding e_r and second values e_l. In order to facilitate the narrative, remember $e_l < e_r$, and $\mathrm{LSB}_{P_{i,j}}$ represents the least significant bit of the $P_{i,j}$. We introduce location map Lm1 and location map Lm2; they are a series of binary sequences. Among them, Lm1 is used to mark the location information which can be embedded (0 represents no embedded location and 1 represents embedded location), and Lm2 is used to record the embedding watermarking information (binary 0 and 1). For the white point of satisfying the rhombus conditions, they are adjusted from left to right and from top to bottom, and the operation is shown as follows.

Case 1. If $e_{i,j} < e_l - 1$ then $I_{i,j} = I_{i,j} + 1$ ($e_{i,j} = e_{i,j} - 1$), Lm1 = 0, Lm2 not marked.

Case 2. If $e_l - 1 < e_{i,j} \le e_l$ and $\mathrm{LSB}_{P_{i,j}} = 0$ then Lm1 = 1, Lm2 = 0.

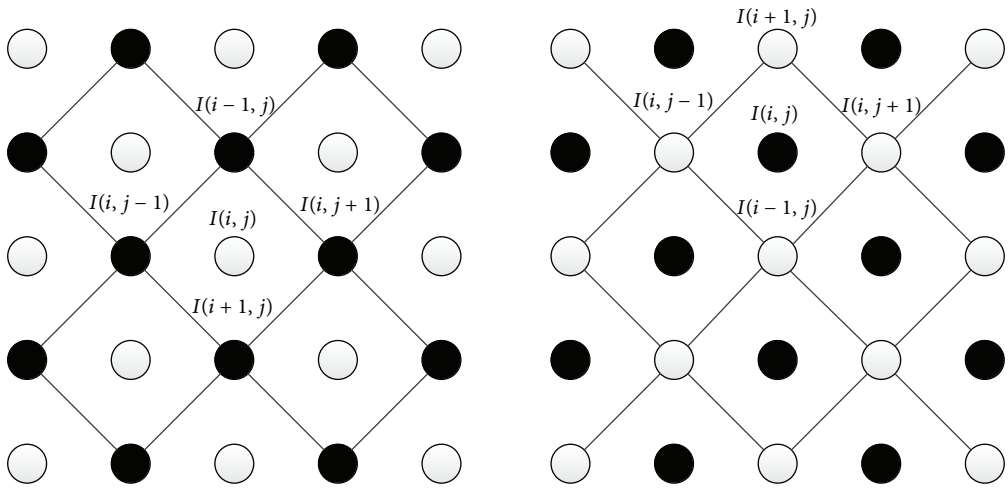
Case 3. If $e_l - 1 < e_{i,j} \le e_l$ and $\mathrm{LSB}_{P_{i,j}} = 1$ then $I_{i,j} = I_{i,j} + 1$ ($e_{i,j} = e_{i,j} - 1$), Lm1 = 0, Lm2 not marked.

Case 4. If $e_l < e_{i,j} \le e_r$ and $\mathrm{LSB}_{P_{i,j}} = 1$ then Lm1 = 1, Lm2 = 1.

Case 5. If $e_l < e_{i,j} \le e_r$ and $\mathrm{LSB}_{P_{i,j}} = 0$ then $I_{i,j} = I_{i,j} - 1$ ($e_{i,j} = e_{i,j} + 1$), Lm1 = 0, Lm2 not marked.

Case 6. If $e_{i,j} > e_r$ then $I_{i,j} = I_{i,j} - 1$ ($e_{i,j} = e_{i,j} + 1$), Lm1 = 0, Lm2 not marked.

In this way, we can get the two parts of the different histogram corresponding to the two location maps. The temporary maps Lm1 and Lm2 are encrypted and compressed; then they are transferred to the user of legitimate embedding watermarking. Once finishing the embedding watermarking at the sender, two temporary maps are immediately destroyed.



(a) First phase (b) Second phase

Figure 2: Rhombus prediction scheme.

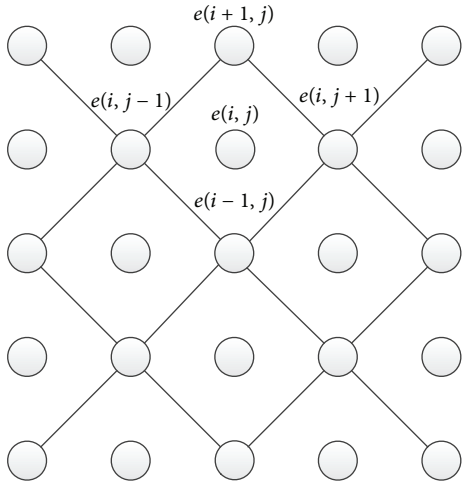


Figure 3: The dividing scheme of rhombus for difference error value.

(2) Image Encryption. We encrypt each pixel in M by the RC4's random key stream K and get encrypted image C (the specific method is in accordance with (2)).

(3) Embedding Watermarking Information

Case 1. Legitimate user of the embedding watermarking information gets location maps Lm1 and Lmp2; he can find the embedding watermarking location according to Lm1; then, Lm1 is read by scanning from left to right and from top to bottom of the sequence of operation; if Lm1 = 1, directly operate Case 2; otherwise, skip position and go next judgment.

Case 2. Embedding watermarking information $b = 0$, read Lm2, If corresponding Lm2 = 0, then $WC_{i,j} = C_{i,j}$, (where $WC_{i,j}$ represents the encryption image with watermarking and $C_{i,j}$ represents ciphertext); if corresponding Lm2 = 1, then $WC_{i,j} = (C_{i,j} + 1) \bmod 256$; embedding watermarking information $b = 1$, read Lm2; if corresponding Lm2 = 0, then $WC_{i,j} = (C_{i,j} - 1) \bmod 256$; if corresponding Lm2 = 1, then $WC_{i,j} = C_{i,j}$.

Case 3. Until all the watermarking information is embedded, Lm1 and Lm2 are destructed; end and the encrypted image with watermarking is formed.

3.2. The Receiver. This section is divided into two parts: part (1) is image decryption; part (2) is extraction watermarking and restoration image.

(1) Image Decryption. The receiver gets the encrypted image with watermarking and the same random seed S and decrypts image by the RC4 decryption after the key sequence R_k is generated (the specific method according to (3)).

(2) Extraction Watermarking and Restoration Image

(i) Preoperation. Preoperation is done before the extraction watermarking and restoration image. We introduce location map Lm3; it is used to mark the location where watermarking information is embedded or the location where it is not embedded (Lm3 = 1 represents the location where location is embedded and Lm3 = 0 represents the location where it is not embedded). The steps are shown as follows.

Step 1. The sender divides the same pixel as one group and calculates each part of the different error value $ce_{i,j}$ ($ce_{i,j}$ represents the pixel of the decrypted image) which is calculated by late the corresponding location pixel $wm_{i,j}$ ($wm_{i,j}$ represents plaintext with watermarking information). Then, a difference histogram is generated after the image is decrypted; we can find the difference error value ce_l at the top of the difference histogram and the other difference error value ce_r ($ce_r = ce_l + 1$).

Step 2. To process the plaintext with watermarking, it is shown as follows.

Case 1. If $ce_{i,j} \le ce_l - 2$, then $wm_{i,j} = wm_{i,j} - 1$, Lm3 not marked.

Case 2. If $ce_l - 2 < ce_{i,j} \le ce_l - 1$, then $wm_{i,j} = wm_{i,j} - 1$, Lm3 = 0.

Case 3. If $ce_l - 1 < ce_{i,j} \le ce_r$, then Lm3 = 1.

Case 4. If $ce_r < ce_{i,j} \le ce_r + 1$, then $wm_{i,j} = wm_{i,j} + 1$, Lm3 = 0.

Case 5. If $ce_{i,j} > ce_r + 1$, then $wm_{i,j} = wm_{i,j} + 1$, Lm3 not marked.

So, we can get the decrypted image with watermarking information; therefore, we send Lm3 (Lm3 has been compressed and encrypted) to the legitimate user who can extract the watermarking information.

Step 3. In the algorithm, due to difference histogram change, corresponding pixels will change and cause overflow or underflow phenomenon. This phenomenon is not much; it can be processed according to the following.

Case 1. If $175 < ce_{i,j} \le 254$, then $wm_{i,j} = 255$.

Case 2. If $254 < ce_{i,j} \le 255$, then $wm_{i,j} = 255$.

Case 3. If $255 < ce_{i,j} \le 256$, then $wm_{i,j}$ is unchanged.

Case 4. If $-255 < ce_{i,j} \le -175$, then $wm_{i,j} = 0$.

Case 5. If $-255 < ce_{i,j} \le -254$, then $wm_{i,j} = 0$.

Case 6. If $-156 < ce_{i,j} \le -255$, then $wm_{i,j}$ unchanged.

(ii) Image Recovery. Image recovery steps are shown as follows.

Step 1. Applying to Lm3, the legitimate user extracts the watermarking information and divides image again at the same sender's pixel group. $ce_{i,j}$ is calculated by the corresponding white point pixel $wm_{i,j}$, according to the order from top to bottom and from left to right. Then, a difference histogram is generated, and we can find the difference error value ce_l at the top of the difference histogram and the other difference error value e_l and e_r ($e_r = e_l + 1$); at this time $P'_{i,j} = (wm_{i-1,j} + wm_{i+1,j} + wm_{i,j-1} + wm_{i,j+1})/4$, $\mathrm{LSB}_{P'_{i,j}}$ represents the lowest significant bit of $P'_{i,j}$, and $m_{i,j}$ represents the restoration pixel.

Step 2. If $ce_l - 1 < ce_{i,j} \le ce_l$, read the corresponding Lm3, and when Lm3 = 1, extract watermarking $b = 1$; however, when $\mathrm{LSB}_{P'_{i,j}} = 1$, then $m_{i,j} = wm_{i,j}$; when $\mathrm{LSB}_{P'_{i,j}} = 0$, then $m_{i,j} = wm_{i,j} + 1$.

Step 3. If $ce_r - 1 < ce_{i,j} \le ce_r$, read the corresponding Lm3, and when Lm3 = 1, extract watermarking $b = 0$; however, when $\mathrm{LSB}_{P'_{i,j}} = 0$, then $m_{i,j} = wm_{i,j}$; when $\mathrm{LSB}_{P'_{i,j}} = 1$, then $m_{i,j} = wm_{i,j} - 1$.

Table 1: The test of resisting rotation attack.

Items	Datum							
RA	−120°	−90°	−60°	−30°	30°	60°	90°	120°
TT	12	12	12	12	12	12	12	12
SET	12	12	12	12	12	12	12	12

Step 4. In other cases, we directly judge the next embedding watermarking information and go to Step 1.

Step 5. When all the watermarking information is extracted, the two-part watermark information is extracted according to the first diamond inner, and then the diamond-four angle is connected. We can get all the watermark information, and the image gets recovery; end.

(iii) Overflow or Underflow Processing. All the watermarking information extraction is finished and the image is recovered; there will also be a small amount of pixels overflow or underflow, and it is processed as follows.

Case 1. If $255 < ce_{i,j} \le 256$ then $m_{i,j} = 255$, $b = 0$.

Case 2. If $-156 < ce_{i,j} \le -255$ then $m_{i,j} = 0$, $b = 1$.

4. Experimental Results and Analysis

4.1. The Data Confidentiality. The RC4 encryption is stream ciphering algorithm with alterable length of key stream K; it accepts a key K from 1 to 256 bytes. As shown in (1) to (5), the RC4 encryption satisfies the confidentiality requirements when the key stream K is long enough and complex enough.

Supposing the length of key stream K is 16 bytes (128 bits), the time of performing an encryption is 1 μs, so the number of the keys K is 2^{128}, and the time T of exhausting half key space is

$$T = \left(2^{128} \div 2\right) \times 1\,\mu\mathrm{s} \approx 5.4 \times 10^{24}\ \text{year}. \tag{19}$$

It is not feasible on computation for an attacker to analyze the key K by the sender and receiver's decrypted image. Also, the middle man cannot analyze the key K based on the decrypted image that is transferred over the channel.

4.2. Resisting Rotation Attack. We adopt the rotation angle (RA) of −120°, −90°, −60°, −30°, 30°, 60°, 90°, and 120° to test the rotation attack. Table 1 gives the successful extracting times (SET) of the watermarking information in encrypted images against the rotation attack; here the testing times (TT) are 12 times. It shows that this reversible date-hiding algorithm has good resisting rotation attack. The main reason is that the rotation cannot change the relative coordinate value of any pixel of the encrypted images, and there is no essential influence on the embedded watermarking information.

4.3. The Embedding Capacity and PSNR. In order to verify the effectiveness of our method, simulation experiments are conducted by using MATLAB7.0 platform. We use the peak

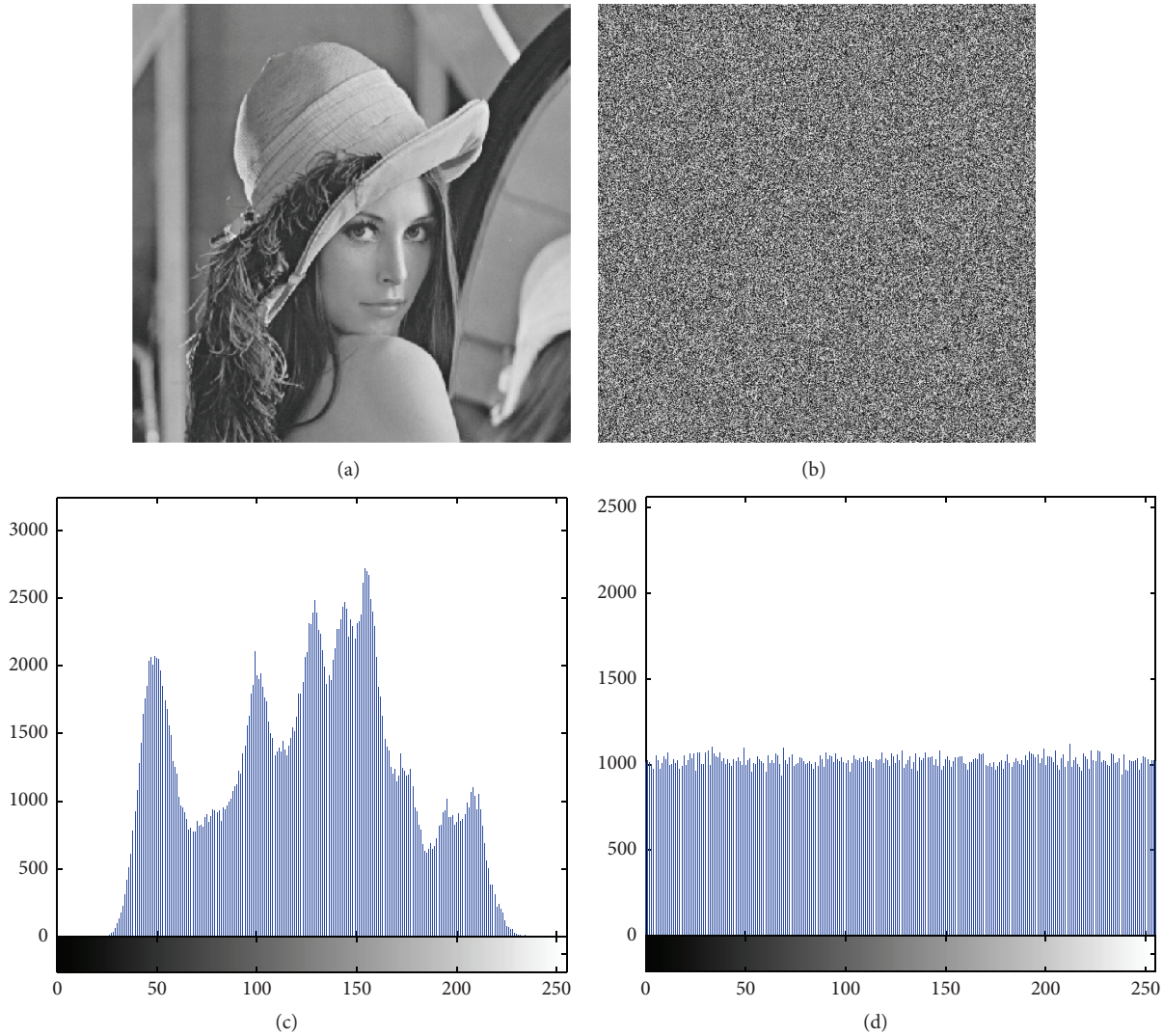


FIGURE 4: Comparing image Lena encryption before and after: (a) original image; (b) encrypted image; (c) the histogram of the original image; (d) the histogram of the encrypted image.

signal-to-noise (PSNR) and the embedding rate (ER) to evaluate the quality of the encrypted image and embedding capacity. For the 512 × 512 gray image, it is

$$\begin{aligned} \text{PSNR} &= 10 \times \log_{10} \frac{255^2}{\text{MSE}} \ \text{(dB)}, \\ \text{MSE} &= \frac{1}{512 \times 512} \sum_{i=0}^{511} \sum_{j=0}^{511} \left(H_{i,j} - H'_{i,j} \right), \end{aligned} \tag{20}$$

where MSE represents mean squared errors.

Then, the ER is

$$\text{ER} = \frac{N_W}{N_P} \left(\frac{\text{bit}}{\text{pixel}}, \text{bpp} \right), \tag{21}$$

where N_W represents the number of binary bits of watermarking information and N_P represents the number of pixels of the image.

We select a standard test image Lean 512 × 512 to test, as shown in Figure 4(a), and the encrypted image as shown in Figure 4(b); obviously, the image is incomprehensible and completely covers the contents of the original image. In order to further illustrate the problem, we generate the histogram of the image before and after encryption; we can see that the encrypted images histogram is flat from Figure 4(d); it also shows great regularity and uncertainty compared to Figure 4(c); the information entropy tends to maximize.

After 15000 bits watermarking information is embedded in the encrypted image, we can get PSNR = 62.36 (dB) when it is decrypted, as shown in Figure 5(a); the effect is very satisfactory looking from the perspective; Figure 5(b) represents the complete recovering image after the watermarking information is extracted; this image and the original image are completely consistent by comparing the data; namely, the proposed algorithm is completely reversible. In order to further experiment, we select four images (512 × 512) as test images from the image database USC-SIPI: Lena, Plane,

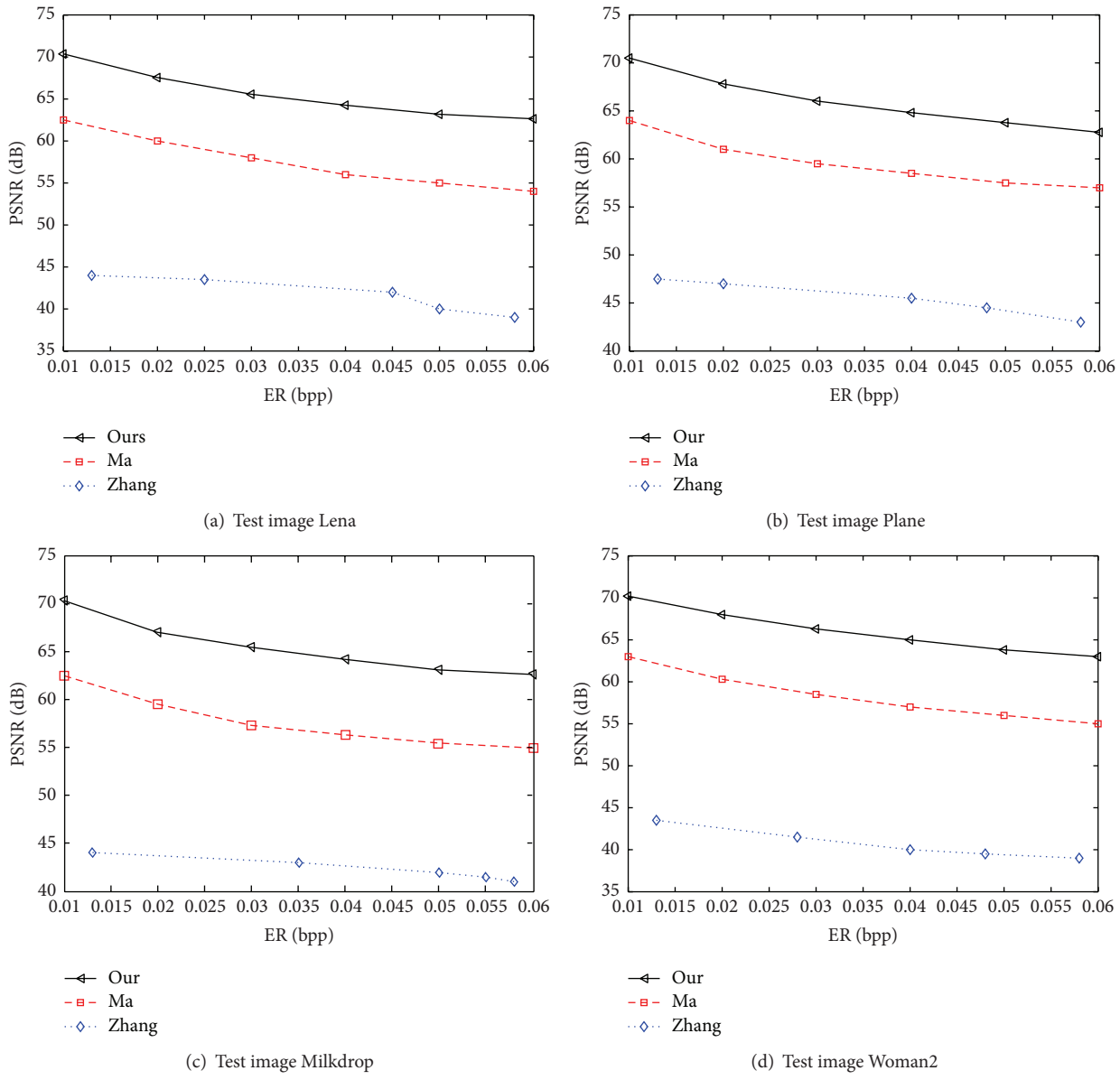
(a) (b)

Figure 5: The decrypted image Lena with watermarking and the ultimate recovery image Lena: (a) the decrypted image Lena with watermarking; (b) the ultimate recovery image Lena.

(a) (b)

(c) (d)

Figure 6: Four test images: (a) Lena; (b) Plane; (c) Milkdrop; (d) Woman2.



(a) Test image Lena

(b) Test image Plane

(c) Test image Milkdrop

(d) Test image Woman2

Figure 7: The chart of our algorithm compared with algorithm Zhang's and Ma's.

Table 2: The ER and PSNR of the four test images.

Images	PSNR/dB					
ER/bpp	0.01	0.02	0.03	0.04	0.05	0.06
Lena	70.36	67.08	65.56	64.26	63.18	62.64
Plane	70.46	67.12	65.61	64.21	63.22	62.68
Milkdrop	70.38	67.10	65.48	64.18	63.12	62.67
Woman2	70.62	67.32	65.52	64.32	63.28	62.72

Milkdrop, and Woman2, as shown in Figure 6. Table 2 lists the four test images' PSNR in different ER, and we can see from the table, with the ER increasing, the PSNR of the image decreases gradually. When the ER is 0.06, PSNR can be maintained at more than 62 dB; it not only shows that the images' PSNR can be maintained at a high level by using this algorithm in different ER to calculate the PSNR but also reflects that the image distortion is still not perceived when PSNR is 62 dB in theory; then it explains the feasibility of this algorithm.

4.4. The Experimental Contrast. We use PSNR value to measure direct decrypted image with hidden information. For the images Lena, Plane, Milkdrop, and Woman2, we compare our algorithm with Zhang et al. [13] and Ma's algorithm [17] from the ER and PSNR, as shown in Figure 7. We assume that Ma and Zhang's algorithm is perfectly correct in extraction. When the ER is greater than 0.015, we use Zhangs algorithm to make the experiment; the image will appear as serious distortion; that is to say, when the ER is greater than 0.015, the test will lose its meaning, so we only

take the ER at 0.005, 0.01, and 0.015. When the ER is between 0 and 0.02 bpp, we use a least significant bit, and when it is more than 0.02 bpp, we use the least two significant bits. We can see that the PSNR gradually decreases with the ER increasing, but our algorithm's curve is located above Ma and Zhang's algorithm; namely, our algorithms PSNR is significantly big in the same ER; it further not only shows that this algorithm is more feasible but also illustrates that this algorithm is better than other algorithms.

5. Conclusion

The cipher image embedding technique is combined with the advantage of encryption technology and information hiding technology. It can be satisfied with the need of important privacy protection and become a new field. This paper presents a novel reversible watermarking embedding algorithm in encrypted image; the algorithm adopts rhombus prediction thought and histogram modification scheme. It truly realizes reversible watermarking extraction and image lossless recovery. Experiments show that the proposed algorithm is better than other algorithms.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Nature Science Foundation of China (no. 61562059 and no. 61461027).

References

[1] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," in *Security and Watermarking of Multimedia Contents III*, vol. 4314 of *Proceedings of SPIE*, pp. 197–208, San Jose, Calif, USA, 2001.

[2] M. Goljan, J. J. Fridrich, and R. Du, "Distortion-free data embedding for images," in *Information Hiding*, I. S. Moskowitz, Ed., vol. 2137 of *Lecture Notes in Computer Science*, pp. 27–41, Springer, Berlin, Germany, 2001.

[3] M. Goljan and R. Du, "Lossless data embedding—new paradigm in digital watermarking," *EURASIP Journal on Advances in Signal Processing*, vol. 2002, Article ID 986842, 12 pages, 2002.

[4] 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.

[5] Q. Pei, X. Wang, Y. Li, and H. Li, "Adaptive reversible watermarking with improved embedding capacity," *Journal of Systems and Software*, vol. 86, no. 11, pp. 2841–2848, 2013.

[6] C.-C. Lin, S.-P. Yang, and N.-L. Hsueh, "Lossless data hiding based on difference expansion without a location map," in *Proceedings of the 1st International Congress on Image and Signal Processing (CISP '08)*, pp. 8–12, Sanya, China, May 2008.

[7] 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.

[8] Y.-S. Juang, L.-T. Ko, J.-E. Chen, Y.-S. Shieh, T.-Y. Sung, and H. C. Hsin, "Histogram modification and wavelet transform for high performance watermarking," *Mathematical Problems in Engineering*, vol. 2012, Article ID 164869, 14 pages, 2012.

[9] Kamran, A. Khan, and S. A. Malik, "A high capacity reversible watermarking approach for authenticating images: exploiting down-sampling, histogram processing, and block selection," *Information Sciences*, vol. 256, pp. 162–183, 2014.

[10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in Cryptology—EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30-June 3, 2010. Proceedings*, vol. 6110 of *Lecture Notes in Computer Science*, pp. 24–43, Springer, Berlin, Germany, 2010.

[11] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, 2010.

[12] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. De Natale, and A. Neri, "A commutative digital image watermarking and encryption method in the tree structured Haar transform domain," *Signal Processing: Image Communication*, vol. 26, no. 1, pp. 1–12, 2011.

[13] X. Zhang, Z. Qian, G. Feng, and Y. Ren, "Efficient reversible data hiding in encrypted images," *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 322–328, 2014.

[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] 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.

[16] W. M. Zhang, K. D. Ma, and N. H. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, no. 1, pp. 118–127, 2014.

[17] 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 Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

[18] H. Y. Zheng, Z. Gao, D. Xiao et al., "Novel reversible data embedding algorithm for encrypted image," *Computer Engineering and Applications*, vol. 50, no. 7, pp. 186–189, 2014.

[19] X. P. Zhang, Z. X. Qian, G. Feng, and Y. Ren, "Efficient reversible data hiding in encrypted images," *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 322–328, 2014.

[20] B. Zhao, W. D. Kou, H. Li, L. Dang, and J. Zhang, "Effective watermarking scheme in the encrypted domain for buyer-seller watermarking protocol," *Information Sciences*, vol. 180, no. 23, pp. 4672–4684, 2010.

[21] J. Yu, G. Zhu, X. L. Li, and J. Q. Yang, "An improved algorithm for reversible data hiding in encrypted image," in *Proceedings of the 11th International Conference on Digital Forensics and Watermaking (IWDW '12), Shanghai, China, October-November 2012*, vol. 7809 of *Lecture Notes in Computer Science*, pp. 384–394, Springer, 2013.

[22] D. Xiao, M.-M. Deng, and Y.-S. Zhang, "Robust and separable watermarking algorithm in encrypted image based on compressive sensing," *Journal of Electronics and Information Technology*, vol. 37, no. 5, pp. 1248–1254, 2015.

[23] D. Xiao, K. Bai, and H. Y. Zheng, "Reversible data-hidng algorithm in encrypted image for security application in cloud computing," *Application Research of Computers*, vol. 32, no. 12, pp. 3702–3713, 2015.

[24] Z. Liu, H. Chen, T. Liu et al., "Image encryption by using gyrator transform and Arnold transform," *Journal of Electronic Imaging*, vol. 20, no. 1, Article ID 013020, pp. 13–20, 2011.

[25] D. Xu and R. Wang, "Context adaptive binary arithmetic coding-based data hiding in partially encrypted H.264/AVC videos," *Journal of Electronic Imaging*, vol. 24, no. 3, Article ID 033028, 2015.

[26] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Annual ACM Symposium on Theory of Computing*, pp. 169–178, Bethesda, Md, USA, 2009.

[27] K. Schmidt-Samoa and T. Takagi, "Paillier's cryptosystem modulo p^2q and its applications to trapdoor commitment schemes," in *Progress in Cryptology—Mycrypt 2005*, E. Dawson and S. Vaudenay, Eds., vol. 3715 of *Lecture Notes in Computer Science*, pp. 296–313, Springer, Berlin, Germany, 2005.

[28] S. D. Galbraith, "Elliptic curve paillier schemes," *Journal of Cryptology*, vol. 15, no. 2, pp. 129–138, 2002.

[29] D. Boneh, E. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proceedings of the Second Theory of Cryptography Conference (TCC '05), Cambridge, Mass, USA, February 2005*, vol. 3378 of *Lecture Notes in Computer Science*, pp. 325–341, Springer, Berlin, Germany, 2005.

[30] A. Klein, "Attacks on the RC_4 stream cipher," *Designs, Codes and Cryptography*, vol. 48, no. 3, pp. 269–286, 2008.

[31] M. S. Abdul Karim and K. Wong, "Universal data embedding in encrypted domain," *Signal Processing*, vol. 94, no. 1, pp. 174–182, 2014.

[32] S. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*, John Wiley & Sons, New York, NY, USA, 2008.