

## Lossless and Reversible Data Hiding In Encrypted Images with Public Key Cryptography

**D.Chinna Kullayappa**

PG Scholar,

Department of ECE,

Chiranjeevi Reddy Institute of Engineering and  
Technology,

Anantapur , AP, India.

**T.Syed Akheel**

Associate Professor,

Department of ECE,

Chiranjeevi Reddy Institute of Engineering and  
Technology,

Anantapur , AP, India.

### ABSTRACT:

*This paper proposes a lossless, a reversible, and a combined data hiding schemes for ciphertext images encrypted by public key cryptosystems with probabilistic and homomorphic properties. In the lossless scheme, the ciphertext pixels are replaced with new values to embed the additional data into several LSB-planes of ciphertext pixels by multi-layer wet paper coding. Then, the embedded data can be directly extracted from the encrypted domain, and the data embedding operation does not affect the decryption of original plaintext image. In the reversible scheme, a preprocessing is employed to shrink the image histogram before image encryption, so that the modification on encrypted images for data embedding will not cause any pixel oversaturation in plaintext domain. Although a slight distortion is introduced, the embedded data can be extracted and the original image can be recovered from the directly decrypted image. Due to the compatibility between the lossless and reversible schemes, the data embedding operations in the two manners can be simultaneously performed in an encrypted image. With the combined technique, a receiver may extract a part of embedded data before decryption, and extract another part of embedded data and recover the original plaintext image after decryption.*

### INTRODUCTION

Encryption and data hiding are two effective means of data protection. While the encryption techniques convert plaintext content into unreadable ciphertext, the data hiding techniques embed additional data into cover media by introducing slight modifications. In

some distortion-unacceptable scenarios, data hiding may be performed with a lossless or reversible manner.

Although the terms “lossless” and “reversible” have a same meaning in a set of previous references, we would distinguish them in this work

We say a data hiding method is lossless if the display of cover signal containing embedded data is same as that of original cover even though the cover data have been modified for data embedding. For example, in [1], the pixels with the most used color in a palette image are assigned to some unused color indices for carrying the additional data, and these indices are redirected to the most used color. This way, although the indices of these pixels are altered, the actual colors of the pixels are kept unchanged. On the other hand, we say a data hiding method is reversible if the original cover content can be perfectly recovered from the cover version containing embedded data even though a slight distortion has been introduced in data embedding procedure. A number of mechanisms, such as difference expansion [2], histogram shift [3] and lossless compression [4], have been employed to develop the reversible data hiding techniques for digital images. Recently, several good prediction approaches [5] and optimal transition probability under payload-distortion criterion [6, 7] have been introduced to improve the performance of reversible data hiding.

Combination of data hiding and encryption has been studied in recent years. In some works, data hiding and encryption are jointed with a simple manner. For

example, a part of cover data is used for carrying additional data and the rest data are encrypted for privacy protection [8, 9]. Alternatively, the additional data are embedded into a data space that is invariable to encryption operations [10]. In another type of the works, data embedding is performed in encrypted domain, and an authorized receiver can recover the original plaintext cover image and extract the embedded data. This technique is termed as reversible data hiding in encrypted images (RDHEI). In some scenarios, for securely sharing secret images, a content owner may encrypt the images before transmission, and an inferior assistant or a channel administrator hopes to append some additional messages, such as the origin information, image notations or authentication data, within the encrypted images though he does not know the image content. For example, when medical images have been encrypted for protecting the patient privacy, a database administrator may aim to embed the personal information into the corresponding encrypted images.

Here, it may be hopeful that the original content can be recovered without any error after decryption and retrieve of additional message at receiver side. In [11], the original image is encrypted by an exclusive-or operation with pseudo-random bits, and then the additional data are embedded by flipping a part of least significant bits (LSB) of encrypted image. By exploiting the spatial correlation in natural images, the embedded data and the original content can be retrieved at receiver side. The performance of RDHEI can be further improved by introducing an implementation order [12] or a flipping ratio [13]. In [14], each additional bit is embedded into a block of data encrypted by the Advanced Encryption Standard (AES). When a receiver decrypts the encrypted image containing additional data, however, the quality of decrypted image is significantly degraded due to the disturbance of additional data. In [15], the data-hider compresses the LSB of encrypted image to generate a sparse space for carrying the additional data. Since only the LSB is changed in the data embedding phase, the quality of directly decrypted image is satisfactory.

Reversible data hiding schemes for encrypted JPEG images is also presented [16]. In [17], a sparse data space for accommodating additional data is directly created by compress the encrypted data. If the creation of sparse data space or the compression is implemented before encryption, a better performance can be achieved [18, 19].

While the additional data are embedded into encrypted images with symmetric cryptosystem in the above-mentioned RDHEI methods, a RDHEI method with public key cryptosystem is proposed in [20]. Although the computational complexity is higher, the establishment of secret key through a secure channel between the sender and the receiver is needless. With the method in [20], each pixel is divided into two parts: an even integer and a bit, and the two parts are encrypted using Paillier mechanism [21], respectively.

Then, the ciphertext values of the second parts of two adjacent pixels are modified to accommodate an additional bit. Due to the homomorphic property of the cryptosystem, the embedded bit can be extracted by comparing the corresponding decrypted values on receiver side. In fact, the homomorphic property may be further exploited to implement signal processing in encrypted domain [22, 23, 24]. For recovering the original plaintext image, an inverse operation to retrieve the second part of each pixel in plaintext domain is required, and then two decrypted parts of each pixel should be reorganized as a pixel.

This paper proposes a lossless, a reversible, and a combined data hiding schemes for public-key-encrypted images by exploiting the probabilistic and homomorphic properties of cryptosystems. With these schemes, the pixel division/reorganization is avoided and the encryption/decryption is performed on the cover pixels directly, so that the amount of encrypted data and the computational complexity are lowered. In the lossless scheme, due to the probabilistic property, although the data of encrypted image are modified for data embedding, a direct decryption can still result in the original plaintext image while the embedded data

can be extracted in the encrypted domain. In the reversible scheme, a histogram shrink is realized before encryption so that the modification on encrypted image for data embedding does not cause any pixel oversaturation in plaintext domain. Although the data embedding on encrypted domain may result in a slight distortion in plaintext domain due to the homomorphic property, the embedded data can be extracted and the original content can be recovered from the directly decrypted image. Furthermore, the data embedding operations of the lossless and the reversible schemes can be simultaneously performed in an encrypted image. With the combined technique, a receiver may extract a part of embedded data before decryption, and extract another part of embedded data and recover the original plaintext image after decryption.

### LOSSLESS DATA HIDING SCHEME

In this section, a lossless data hiding scheme for public-key-encrypted images is proposed. There are three parties in the scheme: an image provider, a data-hider, and a receiver. With a cryptosystem possessing probabilistic property, the image provider encrypts each pixel of the original plaintext image using the public key of the receiver, and a data-hider who does not know the original image can modify the ciphertext pixel-values to embed some additional data into the encrypted image by multi-layer wet paper coding under a condition that the decrypted values of new and original cipher-text pixel values must be same. When having the encrypted image containing the additional data, a receiver knowing the data hiding key may extract the embedded data, while a receiver with the private key of the cryptosystem may perform decryption to retrieve the original plaintext image. In other words, the embedded data can be extracted in the encrypted domain, and cannot be extracted after decryption since the decrypted image would be same as the original plaintext image due to the probabilistic property. That also means the data embedding does not affect the decryption of the plaintext image. The sketch of lossless data hiding scheme is shown in Figure 1.

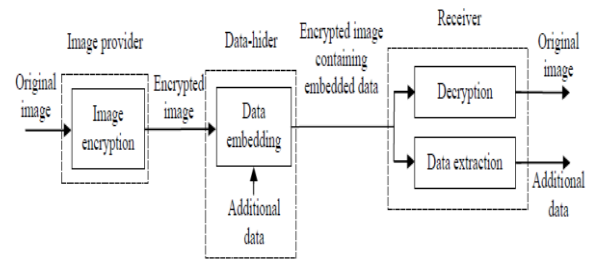


Figure 1. Sketch of lossless data hiding scheme for public-key-encrypted images

#### 2.1.1. Image encryption

In this phase, the image provider encrypts a plaintext image using the public key of probabilistic cryptosystem  $P_k$ . For each pixel value  $m(i, j)$  where  $(i, j)$  indicates the pixel position, the image provider calculates its ciphertext value,

$$c(i, j) = E[p_k, m(i, j), r(i, j)] \quad (1)$$

where  $E$  is the encryption operation and  $r(i, j)$  is a random value. Then, the image provider collects the ciphertext values of all pixels to form an encrypted image.

Actually, the proposed scheme is capitable with various probabilistic public-key cryptosystems, such as Paillier [18] and Damgard-Jurik cryptosystems [25]. With Paillier cryptosystem [18], for two large primes  $p$  and  $q$ , calculate  $n = p \cdot q$ ,  $\lambda = \text{lcm}(p-1, q-1)$ , where  $\text{lcm}$  means the least common multiple. Here, it should meet that  $\text{gcd}(n, (p-1) \cdot (q-1)) = 1$ , where  $\text{gcd}$  means the greatest common divisor. The public key is composed

of  $n$  and a randomly selected integer  $g$  in  $Z_{n^2}^*$ , while the private key is composed of  $\lambda$  and

$$\mu = \left( L(g^\lambda \bmod n^2) \right)^{-1} \bmod n \quad (2)$$

where

$$L(x) = \frac{(x-1)}{n} \quad (3)$$

In this case, (1) implies

$$c(i, j) = g^{m(i,j)} \cdot (r(i, j))^n \bmod n^2 \quad (4)$$

where  $r(i, j)$  is a random integer in  $Z_n^*$ . The plaintext pixel value can be obtained using the private key,

$$m(i, j) = L\left(\left(c(i, j)\right)^d \bmod n^2\right) \cdot \mu \bmod n \quad (5)$$

As a generalization of Paillier cryptosystem, Damgard-Jurik cryptosystem [25] can be also used to encrypt the plaintext image. Here, the public key is composed of  $n$  and an element  $g$  in  $Z_{n^2}^*$  such that  $g = (1+n)^j \cdot x \bmod n^2$  for a known  $j$  relatively prime to  $n$  and  $x$  belongs to a group isomorphic to  $Z_n^*$ , and we may choose  $d$  as the private key when meeting  $d \bmod n \in Z_n^*$  and  $d = 0 \bmod \lambda$ . Then, the encryption in (1) can be rewritten as

$$c(i, j) = g^{m(i, j)} \cdot (r(i, j))^{n^2} \bmod n^{s+1} \quad (6)$$

where  $r(i, j)$  is a random integer in  $Z_{n^{s+1}}^*$ . By applying a recursive version of Paillier decryption, the plaintext value can be obtained from the ciphertext value using the private key. Note that, because of the probabilistic property of the two cryptosystems, the same gray values at different positions may correspond to different ciphertext values.

### 2.1.2 Data embedding

When having the encrypted image, the data-hider may embed some additional data into it in a lossless manner. The pixels in the encrypted image are reorganized as a sequence according to the data hiding key. For each encrypted pixel, the data-hider selects a random integer  $r'(i, j)$  in  $Z_n^*$  and calculates

$$c'(i, j) = c(i, j) \cdot (r'(i, j))^n \bmod n^2 \quad (7)$$

if Paillier cryptosystem is used for image encryption, while the data-hider selects a random integer  $r'(i, j)$  in  $Z_{n^{s+1}}^*$  and calculates

$$c'(i, j) = c(i, j) \cdot (r'(i, j))^{n^s} \bmod n^{s+1} \quad (8)$$

if Damgard-Jurik cryptosystem is used for image encryption. We denote the binary representations of  $c(i, j)$  and  $c'(i, j)$  as  $b_k(i, j)$  and  $b'_k(i, j)$ , respectively,

$$b_k(i, j) = \left\lfloor \frac{c(i, j)}{2^{k-1}} \right\rfloor \bmod 2, \quad k = 1, 2, \dots \quad (9)$$

$$b'_k(i, j) = \left\lfloor \frac{c'(i, j)}{2^{k-1}} \right\rfloor \bmod 2, \quad k = 1, 2, \dots \quad (10)$$

Clearly, the probability of  $b_k(i, j) = b'_k(i, j)$  ( $k = 1, 2, \dots$ ) is  $1/2$ . We also define the sets

$$\begin{aligned} S_1 &= \{(i, j) | b_1(i, j) \neq b'_1(i, j)\} \\ S_2 &= \{(i, j) | b_2(i, j) \neq b'_2(i, j), b_1(i, j) = b'_1(i, j)\} \\ &\dots \\ S_K &= \{(i, j) | b_K(i, j) \neq b'_K(i, j), b_k(i, j) = b'_k(i, j), k = 1, 2, \dots, K-1\} \end{aligned} \quad (11)$$

By viewing the  $k$ -th LSB of encrypted pixels as a wet paper channel (WPC) [26] and the  $k$ -th LSB in  $S_k$  as “dry” elements of the wet paper channel, the data-hider may employ the wet paper coding [26] to embed the additional data by replacing a part of  $c(i, j)$  with  $c'(i, j)$ . The details will be given in the following.

Considering the first LSB, if  $c(i, j)$  are replaced with  $c'(i, j)$ , the first LSB in  $S_1$  would be flipped and the rest first LSB would be unchanged. So, the first LSB of the encrypted pixels can be regarded as a WPC, which includes changeable (dry) elements and unchangeable (wet) elements. In other words, the first LSB in  $S_1$  are dry elements and the rest first LSB are wet positions. By using the wet paper coding [26], one can represent on average  $N_d$  bits by only flipping a part of dry elements where  $N_d$  is the number of dry elements. In this scenario, the data-hider may flip the dry elements by replacing  $c(i, j)$  with  $c'(i, j)$ . Denoting the number of pixels in the image as  $N$ , the data-hider may embed on average  $N/2$  bits in the first LSB-layer using wet paper coding. Considering the second LSB (SLSB) layer, we call the SLSB in  $S_2$  as dry elements and the rest SLSB as wet elements. Note that the first LSB of ciphertext pixels in  $S_1$  have been determined by replacing  $c(i, j)$  with  $c'(i, j)$  or keeping  $c(i, j)$  unchanged in the first LSB-layer embedding, meaning that the SLSB in  $S_1$

are unchangeable in the second layer. Then, the data-hider may flip a part of SLSB in  $S_2$  by replacing  $c(i, j)$  with  $c'(i, j)$  to embed on average  $N/4$  bits using wet paper coding.

Similarly, in the  $k$ -th LSB layer, the data-hider may flip a part of  $k$ -th LSB in  $S_k$  to embed on average  $N/2^k$  bits. When the data embedding is implemented in  $K$  layers, the total  $N \cdot (1 - 1/2^K)$  bits, on average, are embedded. That implies the embedding rate, a ratio between the number of embedded bits and the number of pixels in cover image, is approximately  $(1 - 1/2^K)$ . That implies the upper bound of the embedding rate is 1 bit per pixel. The next subsection will show that, although a part of  $c(i, j)$  is replaced with  $c'(i, j)$ , the original plaintext image can still be obtained by decryption.

### 2.1.3 Data extraction and image decryption

After receiving an encrypted image containing the additional data, if the receiver knows the data-hiding key, he may calculate the  $k$ -th LSB of encrypted pixels, and then extract the embedded data from the  $K$  LSB-layers using wet paper coding. On the other hand, if the receiver knows the private key of the used cryptosystem, he may perform decryption to obtain the original plaintext image. When Paillier cryptosystem is used, Equation (4) implies

$$c(i, j) = g^{m(i, j)} \cdot (r(i, j))^n + \alpha \cdot n^2 \quad (12)$$

where  $\alpha$  is an integer. By substituting (12) into (7), there is

$$c'(i, j) = g^{m(i, j)} \cdot (r(i, j) \cdot r'(i, j))^n \text{ mod } n^2 \quad (13)$$

Since  $r(i, j) \cdot r'(i, j)$  can be viewed as another random integer in  $Z_n^*$ , the decryption on  $c'(i, j)$  will result in the plaintext value,

$$m(i, j) = L((c'(i, j))^{\lambda} \text{ mod } n^2) \cdot \mu \text{ mod } n \quad (14)$$

Similarly, when Damgard-Jurik cryptosystem is used,

$$c'(i, j) = g^{m(i, j)} \cdot (r(i, j) \cdot r'(i, j))^{n^f} \text{ mod } n^{s+1} \quad (15)$$

The decryption on  $c'(i, j)$  will also result in the plaintext value. In other words, the replacement of ciphertext pixel values for data embedding does not affect the decryption result.

### REVERSIBLE DATA HIDING SCHEME

This section proposes a reversible data hiding scheme for public-key-encrypted images. In the reversible scheme, a preprocessing is employed to shrink the image histogram, and then each pixel is encrypted with additive homomorphic cryptosystem by the image provider. When having the encrypted image, the data-hider modifies the ciphertext pixel values to embed a bit-sequence generated from the additional data and error-correction codes. Due to the homomorphic property, the modification in encrypted domain will result in slight increase/decrease on plaintext pixel values, implying that a decryption can be implemented to obtain an image similar to the original plaintext image on receiver side. Because of the histogram shrink before encryption, the data embedding operation does not cause any overflow/underflow in the directly decrypted image. Then, the original plaintext image can be recovered and the embedded additional data can be extracted from the directly decrypted image. Note that the data-extraction and content-recovery of the reversible scheme are performed in plaintext domain, while the data extraction of the previous lossless scheme is performed in encrypted domain and the content recovery is needless. The sketch of reversible data hiding scheme is given in Figure 2.

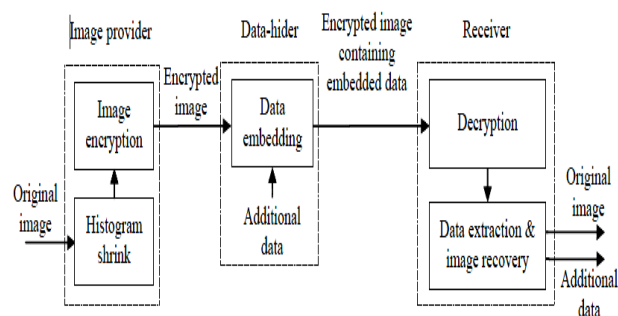


Figure 2. Sketch of reversible data hiding scheme for public-key-encrypted images

**3.1. Histogram shrink and image encryption**

In the reversible scheme, a small integer  $\delta$  shared by the image provider, the data-hider and the receiver will be used, and its value will be discussed later. Denote the number of pixels in the original plaintext image with gray value  $v$  as  $h_v$ , implying

$$\sum_{v=0}^{255} h_v = N \tag{16}$$

where  $N$  is the number of all pixels in the image. The image provider collects the pixels with gray values in  $[0, \delta+1]$ , and represent their values as a binary stream  $BS_1$ . When an efficient lossless source coding is used, the length of  $BS_1$

$$l_1 \approx \sum_{v=0}^{\delta+1} h_v \cdot H \left( \frac{h_0}{\sum_{v=0}^{\delta+1} h_v}, \frac{h_1}{\sum_{v=0}^{\delta+1} h_v}, \dots, \frac{h_{\delta+1}}{\sum_{v=0}^{\delta+1} h_v} \right) \tag{17}$$

where  $H(\cdot)$  is the entropy function. The image provider also collects the pixels with gray values in  $[255-\delta, 255]$ , and represent their values as a binary stream  $BS_2$  with a length  $l_2$ . Similarly,

$$l_2 \approx \sum_{v=255-\delta}^{255} h_v \cdot H \left( \frac{h_{255-\delta}}{\sum_{v=255-\delta}^{255} h_v}, \frac{h_{255-\delta+1}}{\sum_{v=255-\delta}^{255} h_v}, \dots, \frac{h_{255}}{\sum_{v=255-\delta}^{255} h_v} \right) \tag{18}$$

Then, the gray values of all pixels are enforced into  $[\delta+1, 255-\delta]$ ,

$$i_s(i, j) = \begin{cases} 255 - \delta & , \text{ if } m(i, j) \geq 255 - \delta \\ m(i, j) & , \text{ if } \delta + 1 < m(i, j) < 255 - \delta \\ \delta + 1 & , \text{ if } m(i, j) \leq \delta + 1 \end{cases} \tag{19}$$

Denoting the new histogram as  $h'_v$ , there must be

$$h'_v = \begin{cases} 0 & , v \leq \delta \\ \sum_{v=0}^{\delta+1} h_v & , v = \delta + 1 \\ h_v & , \delta + 1 < v < 255 - \delta \\ \sum_{v=255-\delta}^{255} h_v & , v = 255 - \delta \\ 0 & , v > 255 - \delta \end{cases} \tag{20}$$

The image provider finds the peak of the new histogram,

$$V = \underset{\delta+1 \leq v \leq 255-\delta}{\text{arg max}} h'_v \tag{21}$$

The image provider also divides all pixels into two sets: the first set including  $(N-8)$  pixels and the second set including the rest 8 pixels, and maps each bit of  $BS_1$ ,  $BS_2$  and the LSB of pixels in the second set to a pixel in the first set with gray value  $V$ . Since the gray values close to extreme black/white are rare, there is

$$h'_V \geq l_1 + l_2 + 16 \tag{22}$$

when  $\delta$  is not too large. In this case, the mapping operation is feasible. Here, 8 pixels in the second set cannot be used to carry  $BS_1/BS_2$  since their LSB should be used to carry the value of  $V$ , while 8 pixels in the first set cannot be used to carry  $BS_1/BS_2$  since their LSB should be used to carry the original LSB of the second set. So, a total of 16 pixels cannot be used for carrying  $BS_1/BS_2$ . That is the reason that there is a value 16 in (22). The experimental result on 1000 natural images shows (22) is always right when  $\delta$  is less than 15. So, we recommend the parameter  $\delta < 15$ .

Then, a histogram shift operation is made,

$$m_T(i, j) = \begin{cases} m_S(i, j) & , \text{ if } m_S(i, j) > V \\ V & , \text{ if } m_S(i, j) = V \text{ and the corresponding bit is } 0 \\ V - 1 & , \text{ if } m_S(i, j) = V \text{ and the corresponding bit is } 1 \\ m_S(i, j) - 1 & , \text{ if } m_S(i, j) < V \end{cases} \tag{23}$$

In other word,  $BS_1$ ,  $BS_2$  and the LSB of pixels in the second set are carried by the pixels in the first set. After this, the image provider represents the value of  $V$  as 8 bits and maps them to the pixels in the second set in a one-to-one manner. Then, the values of pixels in the second set are modified as follows,

$$m_r(i, j) = \begin{cases} m_s(i, j) & , \text{ if LSB of } m_s(i, j) \text{ is same as the corresponding bit} \\ m_s(i, j) - 1 & , \text{ if LSB of } m_s(i, j) \text{ differs from the corresponding bit} \end{cases} \quad (24)$$

That means the value of  $V$  is embedded into the LSB of the second set. This way, all pixel values must fall into  $[\delta, 255 - \delta]$ .

At last, the image provider encrypts all pixels using a public key cryptosystem with additive homomorphic property, such as Paillier and Damgard-Jurik cryptosystems. When Paillier cryptosystem is used, the ciphertext pixel is

$$c(i, j) = g^{m_T(i, j)} \cdot (r(i, j))^n \bmod n^2 \quad (25)$$

And, when Damgard-Jurik cryptosystem is used, the ciphertext pixel is

$$c(i, j) = g^{m_T(i, j)} \cdot (r(i, j))^{n^2} \bmod n^{s+1} \quad (26)$$

Then, the ciphertext values of all pixels are collected to form an encrypted image.

### 3.2. Data embedding

With the encrypted image, the data-hider divides the ciphertext pixels into two set: Set A including  $c(i, j)$  with odd value of  $(i+j)$ , and Set B including  $c(i, j)$  with even value of  $(i+j)$ . Without loss of generality, we suppose the pixel number in Set A is  $N/2$ . Then, the data-hider employs error-correction codes expand the additional data as a bit-sequence with length  $N/2$ , and maps the bits in the coded bit-sequence to the ciphertext pixels in Set A in a one-to-one manner, which is determined by the data-hiding key. When Paillier cryptosystem is used, if the bit is 0, the corresponding ciphertext pixel is modified as

$$c'(i, j) = c(i, j) \cdot g^{-\delta} \cdot (r'(i, j))^n \bmod n^2 \quad (27)$$

where  $r'(i, j)$  is a integer randomly selected in  $Z^*_{n^2}$ . If the bit is 1, the corresponding ciphertext pixel is modified as

$$c'(i, j) = c(i, j) \cdot g^{\delta} \cdot (r'(i, j))^n \bmod n^2 \quad (28)$$

When Damgard-Jurik cryptosystem is used, if the bit is 0, the corresponding ciphertext pixel is modified as

$$c'(i, j) = c(i, j) \cdot g^{n^{s+1} - \delta} \cdot (r'(i, j))^{n^2} \bmod n^{s+1} \quad (29)$$

where  $r'(i, j)$  is a integer randomly selected in  $Z^*_{n^{s+1}}$ . If the bit is 1, the corresponding ciphertext pixel is modified as

$$c'(i, j) = c(i, j) \cdot g^{\delta} \cdot (r'(i, j))^{n^2} \bmod n^{s+1} \quad (30)$$

This way, an encrypted image containing additional data is produced. Note that the additional data are embedded into Set A. Although the pixels in Set B may provide side information of the pixel-values in Set A, which will be used for data extraction, the pixel-values in Set A are difficult to be precisely obtained on receiver side, leading to possible errors in directly extracted data. So, the error-correction coding mechanism is employed here to ensure successful data extraction and perfect image recovery.

### 3.3 Image decryption, data extraction and content recovery

After receiving an encrypted image containing additional data, the receiver firstly performs decryption using his private key. We denote the decrypted pixels as  $m'(i, j)$ . Due to the homomorphic property, the decrypted pixel values in Set A meet

$$m'(i, j) = \begin{cases} m_T(i, j) + \delta & , \text{ if the corresponding bit is 1} \\ m_T(i, j) - \delta & , \text{ if the corresponding bit is 0} \end{cases} \quad (31)$$

On the other hand, the decrypted pixel values in Set B are just  $m_T(i, j)$  since their ciphertext values are unchanged in data embedding phase. When  $\delta$  is small, the decrypted image is perceptually similar to the original plaintext image.

Then, the receiver with the data-hiding key can extract the embedded data from the directly decrypted image. He estimates the pixel values in Set A using their neighbors,

$$\bar{m}_T(i, j) = \frac{m_T(i-1, j) + m_T(i, j-1) + m_T(i+1, j) + m_T(i, j+1)}{4} \quad (32)$$

and obtain an estimated version of the coded bit-sequence by comparing the decrypted and estimated pixel values in Set A.

That means the coded bit is estimated as 0 if  $\bar{m}_T(i, j) > m'(i, j)$  or as 1 if  $\bar{m}_T(i, j) \leq m'(i, j)$ . With the estimate of coded

Bit-sequence, the receiver may employ the error-correction method to retrieve the original coded bit-sequence and the embedded additional data. Note that, with a larger  $\delta$ , the error rate in the estimate of coded bits would be lower, so that more additional data can be embedded when ensuring successful error correction and data extraction. In other words, a smaller  $\delta$  would result in a higher error rate in the estimate of coded bits, so that the error correction may be unsuccessful when excessive payload is embedded. That means the embedding capacity of the reversible data hiding scheme is depended on the value of  $\delta$ .

After retrieving the original coded bit-sequence and the embedded additional data, the original plaintext image may be further recovered. For the pixels in Set A,  $m_T(i, j)$  are retrieved according to the coded bit-sequence,

$$m_T(i, j) = \begin{cases} m'(i, j) - \delta & , \text{ if the corresponding bit is 1} \\ m'(i, j) + \delta & , \text{ if the corresponding bit is 0} \end{cases} \quad (33)$$

For the pixels in Set B, as mentioned above,  $m_T(i, j)$  are just  $m'(i, j)$ . Then, divides all  $m_T(i, j)$  into two sets: the first one including  $(N-8)$  pixels and the second one including the rest 8 pixels. The receiver may obtain the value of  $V$  from the LSB in the second set, and retrieve  $m_S(i, j)$  of the first set,

$$m_S(i, j) = \begin{cases} m_T(i, j) & , \text{ if } m_T(i, j) > V \\ V & , \text{ if } m_T(i, j) = V \text{ or } V-1 \\ m_T(i, j)+1 & , \text{ if } m_T(i, j) < V-1 \end{cases} \quad (34)$$

Meanwhile, the receiver extracts a bit 0 from a pixel with  $m_T(i, j) = V$  and a bit 1 from a pixel with  $m_T(i, j) = V-1$ . After decomposing the extracted data into  $BS_1$ ,  $BS_2$  and the LSB of  $m_S(i, j)$  in the second set, the receiver retrieves  $m_S(i, j)$  of the second set,

$$m_S(i, j) = \begin{cases} m_T(i, j) & , \text{ if LSB of } m_S(i, j) \text{ and } m_T(i, j) \text{ are same} \\ m_T(i, j)+1 & , \text{ if LSB of } m_S(i, j) \text{ and } m_T(i, j) \text{ are different} \end{cases} \quad (35)$$

Collect all pixels with  $m_S(i, j) = \delta+1$ , and, according to  $BS_1$ , recover their original values within  $[0, \delta+1]$ . Similarly, the original values of pixels with  $m_S(i, j) = 255-\delta$  are recovered within  $[255-\delta, 255]$  according to  $BS_2$ . This way, the original plaintext image is recovered.

### 4.3 COMBINED DATA HIDING SCHEME

As described in Sections 3 and 4, a lossless and a reversible data hiding schemes for public-key-encrypted images are proposed. In both of the two schemes, the data embedding operations are performed in encrypted domain. On the other hand, the data extraction procedures of the two schemes are very different. With the lossless scheme, data embedding does not affect the plaintext content and data extraction is also performed in encrypted domain.

With the reversible scheme, there is slight distortion in directly decrypted image caused by data embedding, and data extraction and image recovery must be performed in plaintext domain. That implies, on receiver side, the additional data embedded by the lossless scheme cannot be extracted after decryption, while the additional data embedded by the reversible scheme cannot be extracted before decryption. In this section, we combine the lossless and reversible



schemes to construct a new scheme, in which data extraction in either of the two domains is feasible. That means the additional data for various purposes may be embedded into an encrypted image, and a part of the additional data can be extracted before decryption and another part can be extracted after decryption.

In the combined scheme, the image provider performs histogram shrink and image encryption as described in Subsection 3.A. When having the encrypted image, the data-hider may embed the first part of additional data using the method described in Subsection 3.B. Denoting the ciphertext pixel values containing the first part of additional data as  $c'(i, j)$ , the data-hider calculates

$$c''(i, j) = c'(i, j) \cdot (r''(i, j))^n \bmod n^2 \quad (36)$$

or

$$c''(i, j) = c'(i, j) \cdot (r''(i, j))^{n^s} \bmod n^{s+1} \quad (37)$$

where  $r''(i, j)$  are randomly selected in  $Z_n^*$  or  $Z_{n^s}^*$  for Paillier and Damgard-Jurik cryptosystems, respectively. Then, he may employ wet paper coding in several LSB-planes of ciphertext pixel values to embed the second part of additional data by replacing a part of  $c'(i, j)$  with  $c''(i, j)$ .

In other words, the method described in Subsection 2.B is used to embed the second part of additional data. On receiver side, the receiver firstly extracts the second part of additional data from the LSB-planes of encrypted domain. Then, after decryption with his private key, he extracts the first part of additional data and recovers the original plaintext image from the directly decrypted image as described in Subsection 3.C.

The sketch of the combined scheme is shown in Figure 3. Note that, since the reversibly embedded data should be extracted in the plaintext domain and the lossless embedding does not affect the decrypted result, the lossless embedding should implemented after the reversible embedding in the combined scheme.

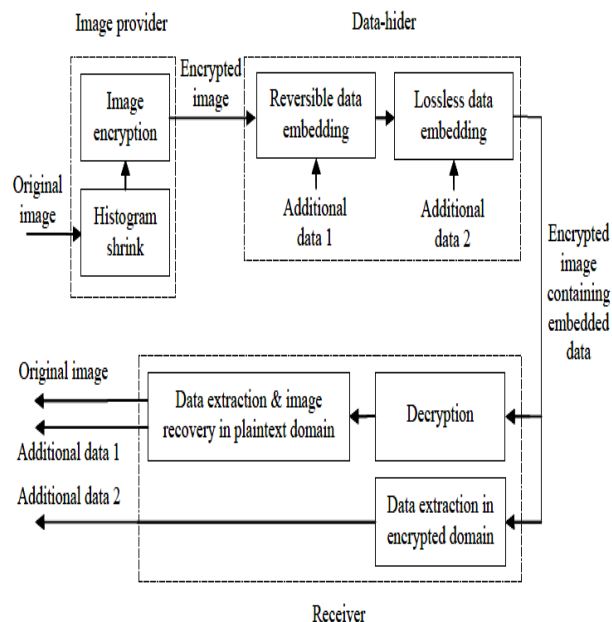


Figure 3. Sketch of combined scheme

Four gray images sized 512×512, Lena, Man, Plane and Crowd, shown in Figure 4, and 50 natural gray images sized 1920×2560, which contain landscape and people, were used as the original plaintext covers in the experiment. With the lossless scheme, all pixels in the cover images were firstly encrypted using Paillier cryptosystem, and then the additional data were embedded into the LSB-planes of ciphertext pixel-values using multi-layer wet paper coding as in Subsection 2.B. Table 1 lists the average value of embedding rates when K LSB-planes were used for carrying the additional data in the 54 encrypted images. In fact, the average embedding rate is very close to  $(1-1/2^k)$ . On receiver side, the embedded data can be extracted from the encrypted domain. Also, the original plaintext images can be retrieved by direct decryption. In other word, when the decryption was performed on the encrypted images containing additional data, the original plaintext images were obtained.

With the reversible scheme, all pixels were encrypted after histogram shrink as in Subsection 3.A. Then, a

half of ciphertext pixels were modified to carry the additional data as in Subsection 3.B, and after decryption, we implemented the data extraction and image recovery in the plaintext domain. Here, the low-density parity-check (LDPC) coding was used to expand the additional data as a bit-sequence in data embedding phase, and to retrieve the coded bit-sequence and the embedded additional data on receiver side. Although the error-correction mechanism was employed, an excessive payload may cause the failure of data extraction and image recovery. With a larger value of  $\delta$ , a higher embedding capacity could be ensured, while a higher distortion would be introduced into the directly decrypted image. For instance, when using Lena as the cover and  $\delta = 4$ , a total of  $4.6 \times 10^4$  bits were embedded and the value of PSNR in directly decrypted image was 40.3 dB. When using  $\delta = 7$ , a total of  $7.7 \times 10^4$  bits were embedded and the value of PSNR in directly decrypted image was 36.3 dB. In both of the two cases, the embedded additional data and the original plaintext image were extracted and recovered without any error. Figure 5 gives the two directly decrypted images. Figure 6 shows the rate-distortion curves generated from different cover images and various values of  $\delta$  under the condition of successful data-extraction/image-recovery. The abscissa represents the pure embedding rate, and the ordinate is the PSNR value in directly decrypted image. The rate-distortion curves on four test images, Lena, Man, Plane and Crowd, are given in Figures 6, respectively. We also used 50 natural gray images sized  $1920 \times 2560$  as the original plaintext covers, and calculated the average values of embedding rates and PSNR values, which are also shown as a curve marked by asterisks in the figure. Furthermore, Figure 7 compares the average rate-PSNR performance between the proposed reversible scheme with public-key cryptosystems and several previous methods with symmetric cryptosystems under a condition that the original plaintext image can be recovered without any error using the data-hiding and encryption keys. In [11] and [12], each block of encrypted image with given size is used to carry one additional bit. So, the embedding rates of the two works are fixed and low.

With various parameters, we obtain the performance curves of the method in [15] and the proposed reversible scheme, which are shown in the figure. It can be seen that the proposed reversible scheme significantly outperforms the previous methods when the embedding rate is larger than 0.01 bpp.

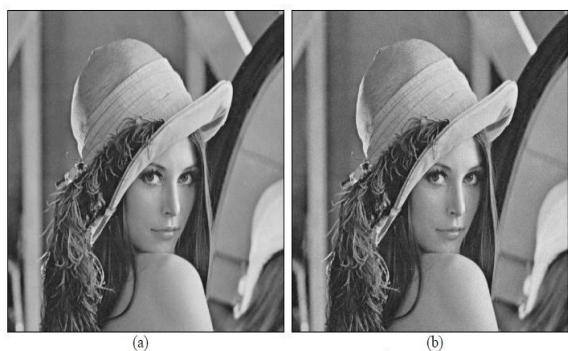
With the combined scheme, we implemented the histogram shrink operation with a value of parameter  $\delta$ , and encrypted the pixels using Paillier cryptosystem. Then, we embedded the first part of additional data into the ciphertext pixel values by the reversible embedding method, and embedded the second part of additional data into the K LSB-planes of the ciphertext pixel values by the lossless embedding method. When having the encrypted image containing the additional data, we firstly extracted the second part of additional data from the LSB-planes of ciphertext pixel values. After decryption, we further extracted the first part of additional data and recovered the original plaintext image in the plaintext domain. Here, the payloads of the two parts of additional data are same as the payloads of reversible and lossless schemes, respectively, and the quality of directly decrypted image is same as that of reversible scheme.



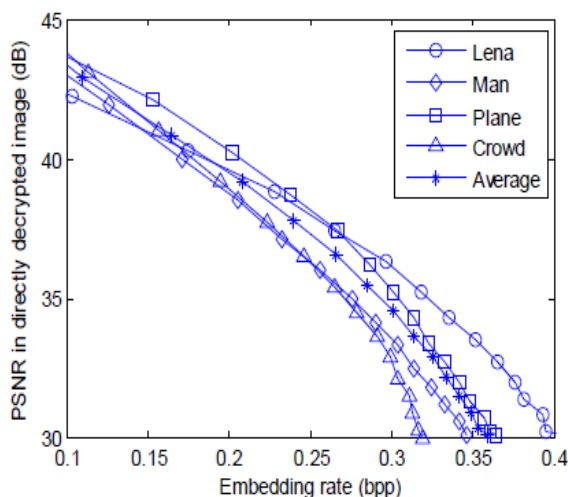
**Figure 4. Cover images (a) Lena, (b) Man, (c) Plane and (d) Crowd**

Table 1. Average payload of lossless scheme with respect to different K

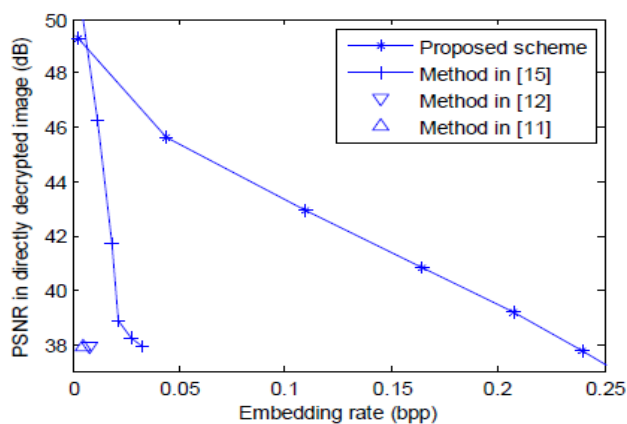
K	1	2	3	4	5
Average embedding rate (bits per pixel) with Paillier cryptosystem	0.499	0.749	0.875	0.937	0.968



**Figure 5. Directly decrypted Lena of reversible scheme (a)  $\delta = 4$ , a total of  $4.6 \times 10^4$  bits embedded and PSNR = 40.3 dB, (b)  $\delta = 7$ , a total of  $7.7 \times 10^4$  bits embedded and PSNR = 36.3 dB**



**Figure 6. Embedding rate-distortion performance of reversible scheme on different cover images**



**Figure 7. Comparison of rate-PSNR performance between the proposed reversible scheme and previous methods**

**CONCLUSION**

This work proposes a lossless, a reversible, and a combined data hiding schemes for cipher-text images encrypted by public key cryptography with probabilistic and homomorphic properties. In the lossless scheme, the ciphertext pixel values are replaced with new values for embedding the additional data into the LSB-planes of ciphertext pixels. This way, the embedded data can be directly extracted from the encrypted domain, and the data embedding operation does not affect the decryption of original plaintext image. In the reversible scheme, a preprocessing of histogram shrink is made before encryption, and a half of ciphertext pixel values are modified for data embedding. On receiver side, the additional data can be extracted from the plaintext domain, and, although a slight distortion is introduced in decrypted image, the original plaintext image can be recovered without any error. Due to the compatibility of the two schemes, the data embedding operations of the lossless and the reversible schemes can be simultaneously performed in an encrypted image. So, the receiver may extract a part of embedded data in the encrypted domain, and extract another part of embedded data and recover the original plaintext image in the plaintext domain.

**REFERENCES**

[1] N. A. Saleh, H. N. Boghdad, S. I. Shaheen, A. M. Darwish, "High Capacity Lossless Data Embedding Technique for Palette Images Based on Histogram Analysis," *Digital Signal Processing*, 20, pp. 1629–1636, 2010.

[2] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Trans. on Circuits and Systems for Video Technology*, 13(8), pp. 890–896, 2003.

[3] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible Data Hiding," *IEEE Trans. on Circuits and Systems for Video Technology*, 16(3), pp. 354–362, 2006.

- [4] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless Generalized-LSB Data Embedding," *IEEE Trans. on Image Processing*, 14(2), pp. 253–266, 2005.
- [5] X. Hu, W. Zhang, X. Li, and N. Yu, "Minimum Rate Prediction and Optimized Histograms Modification for Reversible Data Hiding," *IEEE Trans. on Information Forensics and Security*, 10(3), pp. 653–664, 2015.
- [6] X. Zhang, "Reversible Data Hiding with Optimal Value Transfer," *IEEE Trans. on Multimedia*, 15(2), 316–325, 2013.
- [7] W. Zhang, X. Hu, X. Li, and N. Yu, "Optimal Transition Probability of Reversible Data Hiding for General Distortion Metrics and Its Applications," *IEEE Trans. on Image Processing*, 24(1), pp. 294–304, 2015.
- [8] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative Encryption and Watermarking in Video Compression," *IEEE Trans. on Circuits and Systems for Video Technology*, 17(6), pp. 774–778, 2007.
- [9] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. Natale, and A. Neri, "A Commutative Digital Image Watermarking and Encryption Method in the Tree Structured Haar Transform Domain," *Signal Processing: Image Communication*, 26(1), pp. 1–12, 2011.
- [10] X. Zhang, "Commutative Reversible Data Hiding and Encryption," *Security and Communication Networks*, 6, pp. 1396–1403, 2013.
- [11] X. Zhang, "Reversible Data Hiding in Encrypted Image," *IEEE Signal Processing Letters*, 18(4), pp. 255–258, 2011.
- [12] W. Hong, T.-S. Chen, and H.-Y. Wu, "An Improved Reversible Data Hiding in Encrypted Images Using Side Match," *IEEE Signal Processing Letters*, 19(4), pp. 199–202, 2012.
- [13] J. Yu, G. Zhu, X. Li, and J. Yang, "An Improved Algorithm for Reversible Data Hiding in Encrypted Image," *Proceeding of the 11th International Workshop on Digital-Forensics Watermark (IWDW 2012)*, Shanghai, China, Oct. 31–Nov. 02, 2012, *Lecture Notes in Computer Science*, 7809, pp. 358–367, 2013.
- [14] W. Puech, M. Chaumont, and O. Strauss, "A Reversible Data Hiding Method for Encrypted Images," *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, Proc. SPIE*, 6819, 2008.
- [15] X. Zhang, "Separable Reversible Data Hiding in Encrypted Image," *IEEE Trans. Information Forensics & Security*, 7(2), pp. 526–532, 2012.
- [16] Z. Qian, X. Zhang, and S. Wang, "Reversible Data Hiding in Encrypted JPEG Bitstream," *IEEE Trans. on Multimedia*, 16(5), pp. 1486–1491, 2014.
- [17] M. S. A. Karim, and K. Wong, "Universal Data Embedding in Encrypted Domain," *Signal Processing*, 94, pp. 174–182, 2014.
- [18] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption," *IEEE Trans. Information Forensics & Security*, 8(3), pp. 553–562, 2013.
- [19] W. Zhang, K. Ma, and N. Yu, "Reversibility Improved Data Hiding in Encrypted Images," *Signal Processing*, 94, pp. 118–127, 2014.
- [20] Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted Signal-Based Reversible Data Hiding with Public Key Cryptosystem," *Journal of Visual Communication and Image Representation*, 25, pp. 1164–1170, 2014.

[21] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," Proceeding of the Advances Cryptology, EUROCRYPT'99, LNCS, 1592, pp. 223-238, 1999.

[22] T. Bianchi, A. Piva, and M. Barni, "On the Implementation of the Discrete Fourier Transform in the Encrypted Domain," IEEE Trans. Information Forensics and Security, 4(1), pp. 86-97, 2009.

[23] T. Bianchi, A. Piva, and M. Barni, "Composite Signal Representation for Fast and Storage-Efficient Processing of Encrypted Signals," IEEE Trans. Information Forensics and Security, 5(1), pp. 180-187, 2010.

[24] P. Zheng, and J. Huang, "Discrete Wavelet Transform and Data Expansion Reduction in Homomorphic Encrypted Domain," IEEE Trans. Image Processing, 22(6), pp. 2455-2468, 2013.

[25] I. Damgård, and M. Jurik, "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System," Public Key Cryptography, pp. 119-136, 2001.

[26] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on Wet Paper," IEEE Trans. Signal Processing, 53(10), pp. 3923-3935, 2005.