Multimedia Tools and Applications
https://doi.org/10.1007/s11042-019-07874-w

Encrypted images-based reversible data hiding in Paillier cryptosystem

Cuiling Jiang[1] · Yilin Pang[1]

Received: 26 July 2018 / Revised: 16 April 2019 / Accepted: 10 June 2019
Published online: 08 September 2019

Abstract
Homomorphic public key technology effectively protects privacy, allowing algebraic operations directly in the cipher-text. Therefore, it has been extensively studied in the field of cloud computing. In this study, an encrypted image-based data hiding (EIRDH) algorithm with homomorphic public key cryptosystem is presented. The key contributions are these two sides. (1) An improved fast Paillier homomorphic public key cryptosystem system is proposed for encrypting image. It improves the efficiency of encryption operations greatly. (2) A difference expansion (DE) scheme is developed by exploiting the cover pixel to construct a new pair of pixels for data hiding. Compared with other methods, the experimental results show that, the proposed method has larger payload and higher stego-image quality. It accomplishes the image quality's increasing instead of general decreasing.

Keywords Reversible data hiding · Image encryption · Homomorphic public key cryptosystem · Difference expansion

1 Introduction

With the wide use of internet, information technologies change our lives and make us more convenient, but also result in many security problems. There are recent well-known information security events, such as PRISM and Xkeyscore [23]. Digital image is a kind of popular data widely used on the internet. How to use digital images to accomplish secure communication is still an important issues. Data hiding(DH) is an important branch of information security [23]. The goal of data hiding is to prevent the attacker from detecting a secret message

✉ Cuiling Jiang
cuilingjiang@ecust.edu.cn

Yilin Pang
yilinpang@ecust.edu.cn

[1] School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

in common communication. The data hider embeds the secret message into a chosen cover image to obtain the stego-image. Thus, attackers cannot visually detect the image with the embedded message. Typically, reversible data hiding (RDH) implies that the cover image can be recovered by the receiver [2, 19, 26] after extracting the secret message. RDH is very important when the cover image is significate such as military and medical images.

Recently, cloud storage is a popular and effective way for storing and sharing digital files, such as images, videos, and audios. More and more encrypted images are stored in cloud server. Encrypted image-based reversible data hiding (EIRDH) was introduced and many related algorithms emerged [11, 12, 33–35]. These developed EIRDH studies can provide efficient security techniques for many kinds of applications in cloud. In these EIRDH methods, the secret message is directly embedded into the encrypted cover-image without knowledge of the plaintext content. The cover-image can be completely restored after decryption and extraction of the secret message. EIRDH has a wide range of applications.

For instance, medical image owner information, shooting time, shooting location, and othercloud for rapid retrieval, or a design can first be encrypted, and then the copyright information and other content may be embedded for integrity authentication and copyright protection. EIRDH effectively solves the problem of encrypted image retrieval and privacy protection. Therefore, it has recently attracted a great deal of attention in the field of information security.

In the past decades, the images are mostly encrypted with a stream cipher [11, 15, 16, 18, 21, 33–37, 39], so that they can hardly be directly processed in the encryption domain. Besides, the security level will be degraded when the image contents are revealed in cloud computing. Therefore, this kind of EIRDH algorithms have some limitations. Homomorphic public key technology effectively protects privacy, allowing algebraic operations directly in encryption domain [13, 20]. Since it doesn't need to decrypt the cipher-text before processing them, the user privacy and confidentiality can be protected. It is very useful in current communication systems, such as voting systems trust, bank sensitivity accounts and error tracking, cloud computing and so on. Therefore, RDH based on the homomorphic public key has received much attention in recent years and has developed greatly [3–5, 14, 17, 22, 24, 25, 28–30].

The contributions of the work are as follows:

1) In this study, an EIRDH algorithm in the homomorphic encrypted domain is proposed. A new pair of pixels is constructed by using the cover pixel. Difference expansion(DE) is conducted for RDH.
2) Based on the Chinese remainder theorem(CRT), the proposed method presents an improved fast Paillier homomorphic public key cryptosystem, which improves the efficiency of encryption and decryption greatly.
3) The even or odd quality of the encrypted pair of pixels are exploited for EIRDH. Compared with the other methods, the experimental results show that, the proposed algorithm has significantly large payload up to 1.0 and higher image quality.
4) Comparison of some state-of-the-art algorithms, the stego-image quality of the proposed method realize increasing instead of the general decreasing.

The remainder of this paper is organized as follows. In Section 2, a review is provided of related studies on EIRDH and the improved fast Paillier homomorphic public key cryptosystem system. In Section 3, an algorithm is proposed using difference expansion based on the

improved homomorphic public key cryptosystem. In Section 4, experimental results are presented and discussed. Finally, the conclusion is presented in Section 5.

2 Related works

2.1 EIRDH

The first EIRDH algorithm was introduced by Zhang [33]. It divided an image into several blocks and individually processed each block to hide the secrete message. In this scheme, the secret message can be extracted from the plaintext image after decryption; however, the size of each block should be more than 32×32 for message extraction and image recovery without distortion. Subsequently, Zhang [34] presented a reversible information hiding method based on lossless compression in encrypted domain, which embedded the specified information by compressing the encrypted image. This method can not only recover the original secret information, but also improve the visual quality of the decrypted image. Hong [11] improved the scheme of Zhang [34]. He reduced the error of extraction and recovery by edge matching and adjustment of discriminant functions.

Furthermore, Zhang [35] proposed the use of low-density parity-check matrices to compress the low-order bits of the encrypted image, so that additional redundant space for data embedding may be created, achieving reversible data hiding for data extraction and image decryption. However the effective steganography capacity was limited. Zhang [36] improved his previous work and further presented the scheme of encrypted domain images. Furthermore, Zhang [39] proposed a lossless encryption image concealment method based on public-key encryption. In [39], he utilized wet paper code (WPC) in the encrypted image, and thus the embedded information could be extracted in the encrypted domain. However, the computational complexity of the algorithm is considerably higher. To obtain higher image quality, Qian [21] introduced distributed source coding (LDPC) coding for images encryption. Ma [18] proposed a novel method by reserving space before encryption using a traditional RDH algorithm. He could embed more than 10 times as large payloads as the previous methods. In [37], Zhang made further improvements based on estimation techniques. Moreover, several relative EIRDH methods were presented in [15, 16, 31, 32, 38].

Most image encryption techniques adopt XOR stream cipher, and the cipher-text cannot be processed. The homomorphic public-key cryptosystem is an asymmetric encryption system with higher security and allows algebra operations on cipher-text, which is more suitable for cloud computing and other third-party data processing. Paillier encryption [20] is an RSA-based public key encryption and has a valuable property of homomorphism. The fusion of signal processing and cryptography as a leading paradigm to protect the privacy of users was presented in [13].Then based on the Paillier encryption system, many related algorithms emerged for EIRDH.

Chen et al. [3] proposed a RDH algorithm. This method used Paillier encryption to obtain the encrypted image. Chen combined with RDH algorithm to realize EIRDH. Shiu [22] proposed embedding one-bit secret information using a pair of adjacent Paillier-encrypted pixels, and combined with RDH for differential expansion. This algorithm has lower computational complexity and higher embedding capacity,

whereas only has the maximum embedding rate 0.5. Xiang [28] proposed a new RDH algorithm for homomorphic encrypted images. The target pixel is first selected according to the given key, and it is embedded into other pixels by differential expansion. Compared with the algorithm in [39], the time complexity of the algorithm in [28] is reduced, but also has much higher computational complexity. Li [17] used cross division and additive homomorphism. The method did not cause data expansion and achieved EIRDH by difference histogram shifting. But it is limited to the embedding rate. Furthermore, Li [14] expanded the image histogram by applying the homomorphic encryption for data hiding. This method can achieve data embedding rate up to 1 bit per pixel (bpp) or larger in the lossy mode,but also has the limitation of high computational complexity and the distortion of the host image. In [14], Li's another lossless method can avoid the distortion of the host image, but the embedding rate is less than 0.15. When there is no message to be embedded, it still has pre-processing and result in over-operation. In [4], Di et al adopted a better scheme for vacating room before public key encryption. He used prediction-error expansion, in which the pixel predictor is utilized by interpolation technique. The payload in [4] can get to 0.74bpp, which is much higher than Chen [13] and Shius's [3] 0.5 bpp work, but also it is very limited. In [5] a novel RDH algorithm with image contrast enhancement is firstly proposed based on homomorphic public key cryptosystem. In [24], Tang et al presented an EIRDH scheme of block-based encryption. The EIRDH algorithm of shifting block histogram achieves efficient data embedding. It also has high payload up to the best case 0.68 and correct recovered image. Xiang [29] proposed to embed additional data directly into encrypted image, using homomorphic and probabilistic properties of Paillier cryptosystem. The method has much lower payload. In [30], Xiang presented a novel EIRDH algorithm by mirroring cipher-text group. Xiang's scheme has lower computation complexity, higher security performance, and better embedding performance, but has very limited payload. Tang [25] proposes a RDH algorithm with differential compression(DC) in encrypted image, which achieves high embedding capacity.

2.2 Improved fast Paillier homomorphic public key cryptosystem

Many encryption techniques have been developed to increase the security of information. Some focuses on scrambling the multimedia information such as cryptography. Other security techniques focused on hiding the information such as steganography. Adnan Gutub [10] proposed a new secret-sharing scheme based on parallel counting. Public key algorithm (RSA) is used in the crypto system to provide key encryption before key exchange. In [8], Symmetric key algorithms (DES and AES) are used in the crypto system to perform data encryption. In [1], the authors proposed a light weight cryptographic algorithm for the internet of things (IoT) applications. Montgomery modular inverse computation is needed in several public key cryptographic applications [6, 7, 9]. Generally, the bottleneck of RSA efficiency is big number finding and exponential modular computation. RSA has larger amount of calculation than that of CRT.

The Paillier cryptosystem [20] is an additive homomorphic public key. With Paillier cryptosystem, one of the advantages is that the encrypted image could be processed by the third party for cloud computing. Homomorphic public key

technology allows algebraic operations directly in the cipher-text. Thus, it effectively protects privacy and used in EIRDH [3–5, 14, 17, 22, 24, 28–30]. In practice, the property of additive homomorphism is widely used for multimedia data since it will not result for data expansion [24]. In this paper, we exploit the property of additive homomorphism to encrypt image and embed secret message into the encrypted image.

The Paillier cryptosystem first generates a random number g in the key generation phase, and then decides whether g is eligible or not. In this method, determining g is highly inefficient, and in the encryption and decryption process, a modulo operation should be performed after each square or multiplication. When the value of the secret key is large, the time required for such an operation is longer than the time required for a square or multiplication, significantly affecting the encryption speed of Paillier's algorithm [20]. An appropriate g can be quickly determined at the cost of reducing the range of the total probability of the public key. As g is randomly chosen, the reduction of the possible choice set does not affect the function of the Paillier cryptosystem. In the encryption and decryption processes, the Chinese remainder theorem(CRT) is used to convert high-order to low-order operations, greatly increasing speed. Therefore, CRT can also be applied in many cryptography fields, such as secret sharing and dynamic threshold signature [10].

CRT can be stated as follows:

If $\{m_1,m_2,\ldots,m_k\}$ is a set of relatively prime positive integers, then for any set $\{b_1,b_2,\ldots,b_k\}$ of integers,the following system of equations has a unique solution:

$$\left\{\begin{array}{l} x \bmod m_1 = b_1 \bmod m_1 \\ \ldots \\ x \bmod m_k = b_k \bmod m_k \end{array}\right\} \tag{1}$$

Let $m = m_1 m_2 \ldots m_k, M_i = \frac{m}{m_i}$,and $M_i^{'}$ be such that $M_i M_i^{'} \bmod m_i$ is equal to 1,then the solution of the *Eq.* (1) is described as *Eq.* (2).

$$x = \left(M_1^{'} M_1 b_1 + M_2^{'} M_2 b_2 + \ldots + M_k^{'} M_k b_k\right) \bmod m \tag{2}$$

(1) Key Generation. The key generation process is described in Algorithm 1.

Step 1: Let p and q be two large randomly selected primes, n be their product, and $\lambda(n) = lcm(p-1, q-1)$ be the least common multiple of $(p-1)$ and $(q-1)$.

Step 2: Randomly generate k that is smaller than and relatively prime with n, calculate $k^{'} = (k.\lambda^{-1}) \bmod n$, and let $g = k^{'} \cdot n + 1$.

Step 3: Output the public key (n,g) and save the private key $(p,g,\lambda(n))$. The random number g satisfies $g \in \left\{\omega \middle| Z_{n^2}^{*}\right\}$ and $\gcd(L(g^{\lambda} \bmod n^2), n) = 1$.

---

(2) Fast encryption:

To improve encryption speed, the data can be pre-processed. The fast encryption process is described in Algorithm 2.

Input： n, p, q, g, m

Output: $c = g^m \cdot r^N \bmod n^2$

Step 1: The modulo-inverse algorithm is first used to calculate $b_1 = (q^2)^{-1} \bmod p^2$,

$b_2 = (p^2)^{-1} \bmod q^2$, and $l_1 = b_1 \cdot q^2, l_2 = b_2 \cdot \mathrm{p}^2$.

Step 2: An integer r is randomly selected, and $r \in Z_N^*$.

Step 3: Calculate $k_1 = g^m \bmod p^2, k_2 = g^m \bmod q^2, k_3 = r^n \bmod p^2$, and $k_4 = r^n \bmod q^2$.

Step 4: Compute $g^m \bmod n^2$, and $r^n \bmod n^2$ by using CRT, then obtain c_1 and c_2 as follows:

$$\left\{\begin{array}{l} c_1 = \mathrm{g}^m \bmod n^2 = (k_1 \cdot l_1 + k_2 \cdot l_2) \bmod n^2 \\ c_2 = \mathrm{r}^n \bmod n^2 = (k_3 \cdot l_1 + k_4 \cdot l_2) \bmod n^2 \end{array}\right\}$$

Step 5: According to the nature of modular multiplication, $(a * b) \bmod n = (a \bmod n) * (b \bmod n)$, the cipher-text c is obtained as follows ：

$$c = g^m \cdot r^N \bmod n^2 = (c_1 \cdot c_2) \bmod n^2$$

Table 1 Comparison of the Paillier algorithm [20] and the improved algorithm in terms of key generation, encryption, and decryption for different key length (average runtime: ms)

Algorithms Key length	Paillier [20] key generation	Improved key generation	Paillier [20] Encryption	Improved Encryption	Paillier [20] Decryption	Improved Decryption
16	1.8*10⁶	6	1	2.5	0.5	0.5
32	2.8*10⁶	10	1	3	0.5	0.5
64	5.4*10⁶	12	2	3.5	1	0.5
128	/	13	3	4.5	1	1
256	/	14	9.5	7	2.5	3
512	/	23	28	18	14	12
1024	/	128	80	45	61	45
2048	/	835	312	175	342	273

(3) Fast decryption:

The fast decryption process is described in Algorithm 3.

Input: $c, n, p, q, \lambda, l_1, l_2, (L(g^{\lambda} \bmod n^2)^{-1}) \bmod n$

Output: Plain-text m

Step 1: Calculate $v_1 = c^{\lambda} \bmod p^2, v_2 = c^{\lambda} \bmod q^2$.

Step 2: By using CRT, calculate $v = c^{\lambda} \bmod n^2 = (v_1 \cdot l_1 + v_2 \cdot l_2) \bmod n^2$ and $L(v) = \frac{v-1}{n}$.

Step 3: Compute the plain-text: $m = (L(v).(L(g^{\lambda} \bmod n^2)^{-1}) \bmod n$

The testing software environment is Miccosoft Visual Studio 2015. The Paillier algorithm is tested on MIARCL. The testing platform is Win10 64bit, Intel i4 CPU@3.5GHz, and 8G memory. The length of p and q is 16, 32, 64, 128, 256, 512, 1024, and 2048, respectively. When the length of plaintext is 500 bit, the comparison is illustrated as Table 1.

Table 1 shows that, the improved Paillier algorithm is more efficient in key generation, encryption, and decryption for different key lengths compared with [20]. Specifically, it greatly improves the efficiency of key generation. In encryption and decryption, when the key length is up to 2048, the improved Paillier algorithm uses CRT to decrease the average runtime of encryption by nearly 44% and that of decryption by 20%. In the case of considerably longer key length for higher security, the efficiency of the improved Paillier algorithm can increase more greatly.

3 Image-based reversible data hiding algorithm with homomorphic public key cryptosystem

3.1 Framework of encrypted image-based reversible data hiding with homomorphic public key system

The proposed encrypted image-based RDH algorithm with homomorphic public key cryptosystem (called EIRDH-HP) consists of three entities: the image provider *P*, the data hider *H* and the receiver *R*, as shown in Fig. 1.

Figure 1 shows the flowchart of the proposed EIRDH-HP algorithm. *P* chooses the public/secret key pair, issues the public key, and keeps the private key secret. Then, *P* completes the encryption and transfers the encrypted image to *H*. *H* embeds the secret message into the encrypted image to obtain the stego-image. *R* uses the stego-image and the secret key to recover the cover image and extract the secret message.

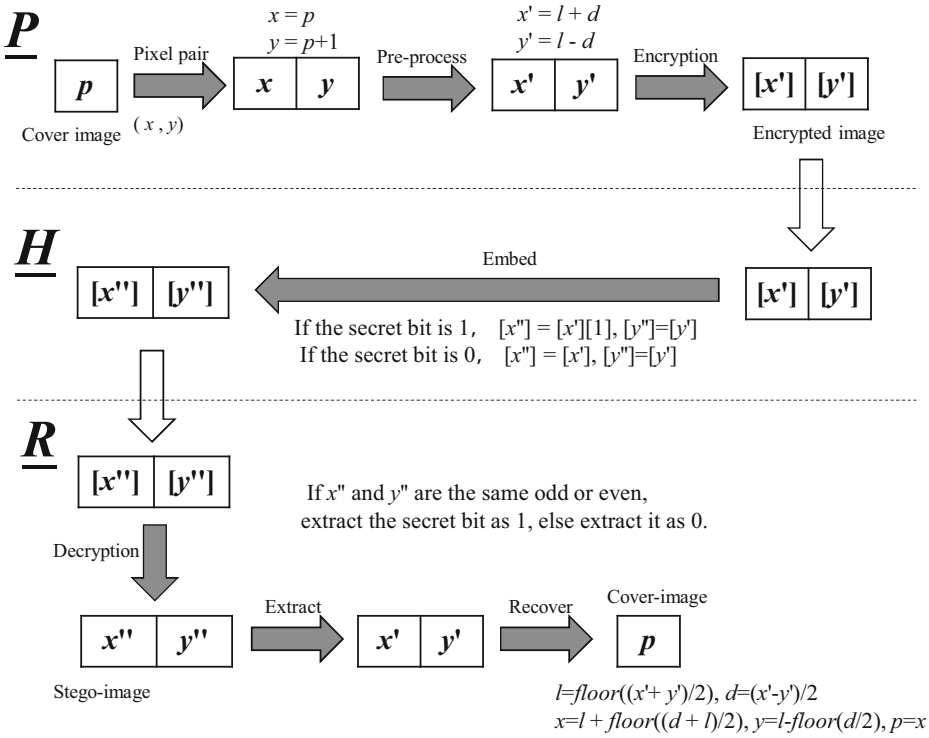


Fig. 1 The proposed EIRDH-HP algorithm

(a) Original Lena (b) Encrypted image

(c) Encrypted image with DH (d) Recovered Lena

Fig. 2 Test on Lena. (**a**)Original Lena; (**b**)Encrypted image; (**c**)Encrypted image with DH; (**d**)Recovered Lena

3.2 Reversible data hiding algorithm in encrypted image with homomorphic public key cryptosystem

The proposed EIRDH-HP algorithm utilizes the improved Paillier cryptosystem to implement fast encryption of images. Then, difference expansion is used to implement the reversible hiding of the secret message and the recovery of the cover image. The algorithm consists of the following procedures.

3.2.1 Image encryption

The image provider *P* chooses the public/secret key pair, releases the public key, keeps the private key secret, and completes the encryption. The process is described as Algorithm 4.

Input: cover image

A pair of pixels (x, y) is constructed and encrypted.

Encrypted pair of pixels is used the improved Paillier encryption algorithm.

Output: the encrypted image *EI*.

Step 1. *P* sets the key pair using the improved Paillier encryption algorithm as illustrated in Section 2.2, including the public key (N, g) and the private key (λ), and keeps (λ) secret.

Step 2. Assume that the current pixel value is p. Then, p is pre-processed and a pixel pair (x, y) is generated with ($x=p$, $y=x+1$).

Step 3. Perform the difference expansion by *Eq*.(8):

$$\begin{cases} l = \lfloor floor((x+y)/2) \rfloor = x \\ d = x - y = -1 \\ x' = l + d \\ y' = l - d \end{cases} \tag{8}$$

Step 4. The pixel pair (x', y') of difference expansion is obtained and they are both odd or even .

if (p=0 and x' = -1)

then location map is used to record :

Location (i, j)= - 1, x' = 0;

if (p=255 and y' = 256)

then location map is used to record :

Location (i, j)= 1, y' = 255;

Step 5. $[x']$ and $[y']$ are generated by encrypting (x', y') by using the public key in the improved Paillier encryption algorithm.

3.2.2 Data hiding

The data hider *H* receives the encrypted image from *P*. By using the public key and the embedding algorithm, *H* embeds the secret message *M* into *EI* to obtain the encrypted image with the embedded message *EIM*, as stated in Algorithm 5.

Input: encrypted image *EI*.

Output: encrypted image with the embedded message *EIM*.

Step 1. The data hider *H* uses his encrypting method to change the chosen secret message *M* into *SM*.

Step 2. For a bit *b* of *SM*, do

if (b= 1)

then $[x''] = [x'][1] = [x'+1], [y''] = [y']$;

x'' and y'' will have the different parity.

else $[x''] = [x'], [y''] = [y']$;

x'' and y'' will be both odd or both even.

3.2.3 Image decryption and recover.

The receiver *R* obtains *EIM* from *H* and decrypts the image using the private key. This step will extract the secret message and recover the cover-image, as described in Algorithm 6.

Input: the encrypted image *EIM*.

Output: the recovered cover-image and secret message.

Step 1. If （x'', y''）are both odd or both even

then the extracted secret bit is 0 , and ($x' = x''$, $y' = y''$).

else the extracted secret bit is 1 , and ($x' = x''-1, y' = y''$).

Step 2. Compute *Eq*. (9):

$$\begin{cases} l = \lfloor floor((x'+y')/2) \rfloor \\ d = (x'-y')/2 \\ x = l + \lfloor floor((d+1)/2) \rfloor \end{cases} \qquad y = l - \lfloor floor(d/2) \rfloor \tag{9}$$

Step 3. The cover image is recovered according to *Eq*. (10):

$$\begin{cases} \text{if } Location(i,j)=0, \quad p=x; \\ \text{if } Location(i,j)=-1, \quad p=0; \\ \text{if } Location(i,j)=1, \quad p=255 \end{cases} \tag{10}$$

We give an example below for describing the above procedures in detail.

Example 1. Let $p = 123$, $x = p = 123$, $y = x + 1 = 124$, and secret bit $b = 1$.Then, a new pair of pixels is constructed as (x, y). Then, $\underline{P}$ computes $d = x - y = -1$, $x' = l + d = 122$, $y' = l - d = 124$ where $l = floor((x + y)/2) = x$. Next, $\underline{P}$ generates the encrypted pair of pixels,[x'] and [y'], as shown in Section 3.2.1. According to Algorithm 5, $\underline{H}$ generates [23] and computes[x''] = [x' + 1] = [122][1] = [123],[y''] = [y'] = [124]. Finally, $\underline{R}$ recovers the cover pixel p, by decrypting [[x''], [[y'']].He finds that $x'' = 123$,$y'' = 124$ have the different parity, and easily extracts the secret bit $b = 1$. He can recover$x' = 122$ and$y' = 124$. According to *Eq.*(9), he computes $l = floor((x' + y')/2) = 123$, $d = (x' - y')/2 = -1$, and then recovers $x = l + floor((d + 1)/2) = 123$.The cover image can be recovered.

4 Experimental results and analysis

4.1 Data hiding results

To evaluate the performance of the proposed method, experiments regarding image quality and capacity were conducted. Chen et al. 'method [13], Shiu et al. 'method in [3], Li's lossy method [14], Di et al [4], Tang et al [24], and the proposed method were compared under the same conditions and on the same image database [27] . In this section, the experimental results of these methods are conducted with 150 color images, each having 512×512 pixels. For space limitation, some typical visual examples is presented here, as shown in Figs. 2, 3 and 4.

As the stego-image quality and data hiding capacity are the most important criteria in evaluating an EIRDH method, the experiments focused on these two criteria. The peak signal to noise ratio(PSNR) was used to evaluate image quality. The PSNR of a gray-level image is defined by:

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} dB \tag{10}$$

The mean square error(*MSE*) for an $N \times N$gray-level image is defined as follows:

$$MSE = \left(\frac{1}{N}\right)^2 \sum_{i=1}^{N} \sum_{j=1}^{N} \left(X_{ij} - \overline{X_{ij}}\right)^2 \tag{11}$$

where X_{ij} and $\overline{X_{ij}}$ represent the pixel values of the cover image and the stego-image at (i, j), respectively. In general, the bigger the PSNR is, the better the visual quality of the image is.

SSIM is used to evaluate the similarity between the source image and distorted image. SSIM takes advantage of characteristics of the human visual system (HVS). Suppose X and Y represent the two images. The SSIM function are defined as followed,

$$SSIM(X, Y) = \frac{(2\mu_X\mu_Y + C_1).(2\sigma_X\sigma_Y + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{12}$$

where $\mu_X^2, \mu_Y^2, \sigma_X, \sigma_Y$ are the means and variances of X and Y. C_1 and C_2 are the small constants near zero. In practice, a bigger SSIM means better visual quality of the image. The maximum value of SSIM is 1.

Payload (called embedding rate) is defined by Eq. (13) .

$$Payload(bpp) = \frac{\text{Total number of embedded bits of secret message}}{\text{Total number of pixels in cover image}} \tag{13}$$

(a) Lena (b) Aerial (c) Airplane

(d) Baboon (e) House (f) Peppers

Fig. 3 Cover-images: (**a**) Lena; (**b**) Aerial; (**c**) Airplane; (**d**) Baboon; (**e**) House; (**f**) Peppers

Table 2 shows that the payload in encrypted images are difficult to reach 1 bpp by using the conventional RDH algorithms [3, 4, 13, 14, 17, 18, 24, 39],i.e.,0.8,0.5,0.5,0.5,0.6,1.0,0.13,0.74,0.68. The payload in the conventional methods [11, 33–36] are smaller than 0.1.The proposed method are much higher than that of others and has the same payload 1.0 as Li's lossless method's [14]. Li's lossless method [14] can get the payload up to 1 bpp,but has the limitation of high computational complexity and the distortion of the host image during the histogram shifting.

As shown in Fig. 2, when the payload is 1 bpp, Fig.2a is the original standard test image Lena with size of 512 × 512 pixels. Figure 2b and c are encrypted image and encrypted image with data hiding(DH), respectively. They are both chaotic images. Figure 2d is the final recovered image, which is exactly same as the cover image Fig. 2a.

Figures 3 and 4 shows the six over images (Lena, Aerial, Airplane, Baboon, House, and Peppers, with 512 × 512 pixels) and the corresponding stego-images. Figures 3 and 4 demonstrate that our stego-image are almost identical to the cover images virtually.

Table 3 shows the performance of our EIRDH-HP. Visual qualities of our directly decrypted images are good enough. The average PSNR of decrypted images is close to 57 and their SSIM are near to 1. Since our recovered images are very identical to their original cover images, all PSNRs tend to $+\infty$ and all SSIM are 1.

4.2 Security analysis

To analyze security of our proposed scheme, some widely used statistical metrics are evaluated, such as Shannon entropy, χ^2 test, and NPCR (number of pixel change

(a) Stego-Lena (b) Stego-Aerial (c) Stego-Airplane

(d) Stego-Baboon (e) Stego-House (f) Stego-Peppers

Fig. 4 Stego-images by the proposed method: (**a**) Stego-Lena; (**b**) Stego-Aerial; (**c**) Stego-Airplane; (**d**) Stego-Baboon; (**e**) Stego- House; (**f**) Stego-Peppers

rate). Definitions of Shannon entropy, χ^2test, NPCR and UACI are defined as follows:

1. Shannon entropy

$$H(I) = -\sum_{i=0}^{255} P(x_i)\log_2(P(x_i)) \tag{14}$$

where I is an image with 256 Gy levels and $P(x_i)$ is the probability of a gray level x_i ($0 \leq i \leq 255$). The maximum theoretical value of $H(I)$ is $\log_2 256$ and equal to 8. A bigger Shannon entropy means more secure image.

2. χ^2 test

$$\chi^2 = 256(M \times N)\sum_{i=0}^{255}\left(P(x_i) - \frac{1}{256}\right)^2 \tag{15}$$

where 1/256 is the theoretical probability of a gray level x_i ($0 \leq i \leq 255$) for a chaotic image. $M \times N$ are the image size. Note that χ^2 indicates the deviation between a test image and its theoretical chaotic image. Clearly, a smaller χ^2 value means a more secure image.

Table 2 Comparison of payload from different methods

Methods	Zhang [39]	Ma [18]	Chen [3]	Shiu [22]	Li [17]	Li's lossy method [14]	Li's lossless method [14]	Di [4]	Tang [24]	Proposed method
Payload (bpp)	0.8	0.5	0.5	0.5	0.6	1.0	0.13	0.74	Best case: 0.68; Worst case:0.02	1.0

3. NPCR

$$\mathrm{NPCR} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} F(i,j)}{M \times N} \times 100\% \tag{16}$$

where $F(i,j)$ is defined as:

$$F(i,j) = \begin{cases} 0, if I(i,j) = I^{'}(i,j) \\ \\ 1, \text{otherwise.} \end{cases} \tag{17}$$

The maximum theoretical value of NPCR is 100%. Much bigger the NPCR is, the more secure the test image is.

Table 4 illustrates the results of the used statistical metrics. It is found that the entropies of the original cover images are far away from the theoretical maximum value 8, whereas the entropies of their encrypted images and encrypted images with DH are all close to 8. This indicates that our encrypted images and encrypted images with DH are secure in terms of entropy. In Table 4, square root value of χ^2 of original images are much bigger than those of the encrypted images and encrypted images with DH. In the view of χ^2 test metric, the value of our encrypted images and their encrypted images with DH are very similar. Thus, they are also secure. Additionally, all the NPCR values are nearly equal to the theoretical value 100%. It indicates that our encrypted images and marked encrypted images are secure enough. From these analyses, it can be concluded that our encrypted image and encrypted images with DH are both secure.

Figure 5 presents the comparison of six methods: Chen et al [13], Shiu et al [3], Li's lossy method [14], Di et al [4], Tang et al [24], and the proposed method. It shows that, when the payload increases, the proposed method provides higher stego-quality than Chen et al [13], Shiu et al method [3], Di et al [4], and Tang et al [24]. Generally, when payload increases, PSNRs of stego-

Table 3 Our statistical performance on PSNR and SSIM under payload 1.0 bpp

Images		Lena	Aerial	Airplane	Baboon	House	Peppers
Decrypted image	PSNR	56.8250	56.8357	56.8439	56.8349	56.8450	56.8452
	SSIM	0.9995	0.9998	0.9994	0.9998	0.9996	0.9996
Recovered image	PSNR	+∞	+∞	+∞	+∞	+∞	+∞
	SSIM	1	1	1	1	1	1

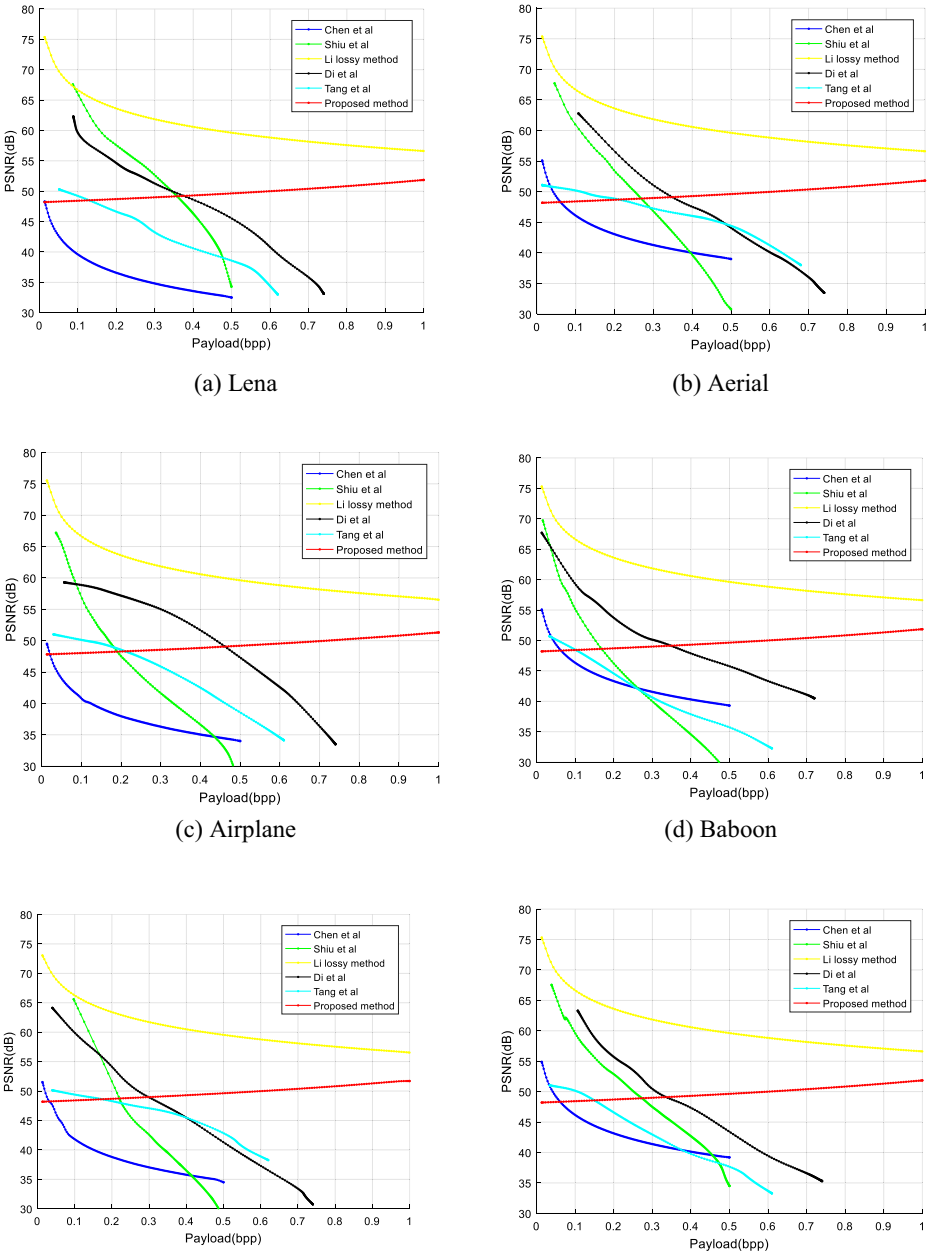
Table 4 Results of the used statistical metrics

Images	Entropy	square root of χ^2	NPCR (%)
Lena			
Original image	7.3937	430.8245	–
Encrypted image	7.9872	66.2637	99.61
Encrypted image with DH	7.9202	62.850	99.77
Aerial			
Original image	6.9940	664.7239	–
Encrypted image	7.9868	66.7713	99.58
Encrypted image with DH	7.9373	62.851	99.92
Airplane			
Original image	6.3127	963.3561	–
Encrypted image	7.9846	72.1085	99.6
Encrypted image with DH	7.9380	62.824	99.93
Baboon			
Original image	7.6955	284.4855	–
Encrypted image	7.9886	61.1406	99.57
Encrypted image with DH	7.9404	62.809	99.86
House			
Original image	7.0687	666.705	–
Encrypted image	7.9877	63.2959	99.57
Encrypted image with DH	7.9208	62.861	99.91
Peppers			
Original image	7.2142	560.1206	–
Encrypted image	7.9878	63.2342	99.56
Encrypted image with DH	7.9383	62.886	99.91

images will decrease, as the other five methods illustrated in Fig. 5. Whereas, stego-image quality of the proposed method does not degrade and shows a slight increase. That is to say, the stego-image does not degrade. As seen in Fig. 5, stego-images and cover-images of the proposed method have the same visual quality. When the proposed method generates a pixel pair ($x = p, y = x + 1$) during image pre-processing, it is assumed that the original pixel value is x. Then, according to *Eq.* (8), the pixel value of the stego-image is $x' = l + d = x - 1$. Therefore, when no secret message is embedded, each pixel value of the stego-image must differ from that of the cover-image by 1.Whereas, after embedding the secret bit 1, we have$[x'']=[x'][1]=[x'+1]=[x]$. That is to say, the pixel value of the stego-image will be same as the cover-image's. Consequently, when the payload of the proposed method increases, the stego-images goes up instead of decreasing, meanwhile still maintain considerably higher image quality than the other four methods.

As seen in Fig. 5, Chen et al 'method [13], Shiu et al 'method [3] and Li's lossy method [14] have higher stego-image quality when the payload is low. However, as payload increases, stego-image quality obviously degenerates. Moreover, the maximum payload in theory for these two algorithms in [3, 13] is 0.5. The maximum payload of Di et al [4] and Tang et al' method [24] is 0.74 and 0.68 under the block size of 9×9, respectively. The maximum payload for Li's lossy method [14] could reach 1 bpp. Although Li's lossy method [14] has slightly higher stego-image quality than ours under the same payload, it can result the distortion of the host image. By contrast, the payload for the proposed method is up to 1 and do not have changes of the host image.

Computation cost is an important metric in the homomorphic encrypted domain. In the proposed method, the computation complexity of data hiding is O(k), where k is the number of



(a) Lena (b) Aerial

(c) Airplane (d) Baboon

(e) House (f) Peppers

Fig. 5 Comparison of PSNR and payload. (**a**) Stego-Lena; (**b**) Stego-Aerial; (**c**) Stego-Airplane; (**d**) Stego-Baboon; (**e**) Stego- House; (**f**) Stego-Peppers

hidden bits. In [39], the computation complexity is $O(k^3)$ due to the use of multi-layer wet paper coding(WPC) operation. Because the other methods apply homomorphic multiplication operation to embed data in encrypted domain, they have the same complexity $O(k)$, as shown in Table 5.

Table 5 Analysis of time efficiency among different algorithms

Algorithms	Zhang [39]	Chen [3]	Shiu [22]	Li [14]	Di [4]	Tang [24]	Proposed method
Homomorphism	WPC	additive	additive	additive	additive	additive	additive
Computation complexity	$O(k^3)$	$O(k)$	$O(k)$	$O(k)$	$O(k)$	$O(k)$	$O(k)$
Extra data expansion	No	Yes	Yes	No	No	No	Yes

5 Conclusion

An EIRDH algorithm based on homomorphic public key cryptosystem was proposed. By using CRT, this paper presents an improved fast Paillier public key cryptosystem to achieve the image encryption and decryption, which greatly improves the computational efficiency. A new pair of pixels is constructed with DE to embed the secret message into the encrypted image. The even or odd quality of the encrypted pair of pixels is exploited for EIRDH. Compared with some state-of-the-art algorithms, the proposed scheme has significantly large payload up to 1.0 and higher image quality. It accomplishes the image quality's increasing instead of the general decreasing. Since the user privacy and data integrity can get more security by using homomorphic encryption, the proposed scheme are more applicable in the cloud computing.

Acknowledgements The authors are grateful for the anonymous reviewers' insightful comments and valuable suggestions sincerely, which can substantially improve the quality of this study. This work is partially supported by the National Natural Science Foundation of China (No.61371150).

References

1. Alassaf N, Gutub A, Parah SA, Al Ghamdi M (2018) Enhancing speed of SIMON: a light-weight-cryptographic algorithm for IoT applications. Multimed Tools Appl. https://doi.org/10.1007/s11042-018-6801-z
2. Caldelli R, Filippini F, Becarelli R (2010) Reversible watermarking techniques: an overview and a classification. EURASIP Journal on Information Security
3. Chen YC, Shiu CW, Horng G (2014) Encrypted signal-based reversible data hiding with public key cryptosystem. J Vis Commun Image Represent 25:1164–1170
4. Di F, Duan J, Zhang M, Liu J (2018) Encrypted image-based reversible data hiding with public key cryptography from interpolation-error expansion. Adv Internetw Data Web Technol: 138–149: doi: https://doi.org/10.1007/978-3-319-59463-7_14
5. Di F, Duan J, Zhang M, Zhang Y, Liu J (2018) Reversible image data hiding with homomorphic encryption and contrast enhancement. Advances in Internetworking, Data & Web Technologies:150–159. doi: https://doi.org/10.1007/978-3-319-59463-7_15
6. Gutub AA-A (2007) High speed hardware architecture to compute galois fields GF(p) Montgomery inversion with scalability features. IET Comput Digit Technol 1(4):389–396
7. Gutub AA-A, Ferreira Tenca A. (2003) Efficient scalable hardware architecture for Montgomery inverse computation in GF(p),2003, IEEE Workshop on Signal Processing Systems (SIPS'03): 93–98. Seoul, Korea, August 27–29
8. Gutub AA-A, Khan FA-A (2012) Hybrid Crypto Hardware Utilizing Symmetric-Key & Public-Key Cryptosystems. International Conference on Advanced Computer Science Applications and Technologies: 116–121
9. Gutub AA-A, Ferreira Tenca A, Koç ÇK (2002) Scalable VLSI Architecture for GF(p) Montgomery Modular Inverse Computation, Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI.02): 46–51
10. Gutub A, Al-Juaid N, Khan E (2017) Counting-based secret sharing technique for multimedia applications. Multimed Tools Appl. https://doi.org/10.1007/s11042-017-5293-6

11. Hong W, Chen TS, Wu HY (2011) An improved reversible data hiding in encrypted images using side match. IEEE Signal Process Lett 9(4):199–202
12. Janani S, P Poorani S (2014) In-dependable data hiding in an encrypted image using FCM-DH algorithm[J]. IJRCCT 3(2):223–225
13. Lagendijk RL, Zekeriya E, Barni M (2013) Encrypted signal processing for privacy protection: conveying the utility of homomorphic encryption and multiparty computation. IEEE Signal Process 30(1):82–105
14. Li M, Li Y (2017) Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding. Signal Process 130:190–196
15. Li M, Xiao D, Peng Z, Nan HA (2014) Modified reversible data hiding in encrypted images using random diffusion and accurate prediction [J]. ETRI J 36(2):325–328
16. Li M, Xiao D, Kulsoom A, Zhang Y (2015) Improved reversible data hiding for encrypted images using full embedding strategy. Electron Lett 51(9):690–691
17. Li M, Xiao D, Zhang Y, Nan H (2015) Reversible data hiding in encrypted images using cross division and additive homomorphism. Signal Process: Image Commun 39:234–248
18. Ma K, Zhang W, Zhao X, Yu N, Li F (2013) Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans Inform Forensics Sec 8(3):553–562
19. Ou B, Li X, Zhao Y, Ni R (2013) Reversible data hiding based on PDE predictor. J Syst Softw 86(10):2700–2709
20. Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. Proceeding of the Advances Cryptology, EUROCRYPT99,LNCS 1592:223–238
21. Qian Z, Zhang X (2015) Reversible data hiding in encrypted image with distributed source encoding. IEEE Transactions on Circuits and Systems for Video Technology
22. Shiu C-W, Chen Y-C, Hong W (2015) Encrypted image-based reversible data hiding with public key cryptography from difference expansion. Signal Process Image Commun 39:226–233
23. Tang Z, Wang F, Zhang XQ (2017) Image encryption based on random projection partition and chaotic system. Multimed Tools Applic 76(6):8257–8283
24. Tang Z, Xu S, Ye D, Wang J, Zhang X, Yu C (2018) Real-time reversible data hiding with shifting block histogram of pixel differences in encrypted image. J Real-Time Image Proc. https://doi.org/10.1007/s11554-018-0838-0
25. Tang Z, Xu S, Yao H, Qin C (2018) Reversible data hiding with differential compression in encrypted image. Multimed Tools Appl. https://doi.org/10.1007/s11042-018-6567-3
26. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circ Syst Video Technol 3(8):890–896
27. USC-SIPI image database ⟨http://sipi.usc.edu/database⟩
28. Xiang SJ, Luo XR (2016) Reversible data hiding in encrypted image based on homomorphic public key cryptosystem. Ruan Jian Xue Bao/Journal of Software 27(6):1592–1601 (in Chinese). http://www.jos.org.cn/1000-9825/5007.htm
29. Xiang S, Luo X (2017) Efficient reversible data hiding in encrypted image with public key cryptosystem. EURASIP J Adv Signal Process 2017:59
30. Xiang S, Luo X (2018) Reversible data hiding in homomorphic encrypted domain by mirroring Ciphertext group. IEEE Trans Circ Syst Video Technol 28(11):3099–3110
31. Xiao D, Chen S (2014) Separable data hiding in encrypted image based on compressive sensing [J]. Electron Lett 50(8):598–600
32. Yin Z., Luo B, Hong W (2014) Separable and error-free reversible data hiding in encrypted image with high payload [J]. Sci World J
33. Zhang X (2011) Reversible data hiding in encrypted image. IEEE Signal Process Lett 18(4):255–258
34. Zhang X (2011) Lossy compression and iterative reconstruction for encrypted image. IEEE Trans Inform Forensics Sec 6(1):53–58
35. Zhang X (2012) Separable reversible data hiding in encrypted image. IEEE Trans Inform Forensics Sec 7(2):826–832
36. Zhang X (2014) Reversibility improved data hiding in encrypted images. Signal Process 94(0):118–127
37. Zhang W, Ma K, Yu N (2014) Reversibility improved data hiding in encrypted images. Signal Process 94:118–127
38. Zhang S, Gao T, Sheng G (2014) A joint encryption and reversible data hiding scheme based on integer-DWT and arnold map permutation [J]. Journal of Applied Mathematics
39. Zhang XP, Long J, Wang Z, Cheng H (2016) Lossless and reversible data hiding in encrypted images with public key cryptography. IEEE Trans Circ Syst Video Technol 26(9):1622–1631

Cuiling Jiang received the Ph.D. degree in pattern identification and intelligent system from East University of Science and Technology, China, in 2009. She is with the school of Information Science and Engineering, East China University of Science and Technology, Shanghai, China. Her current research interests include data hiding and images processing. She is a reviewer of some reputable journals, such as the Circuits, Systems, and Signal Processing, the Journal of Visual Communication and Image Representation, Journal of Electronic Imaging.

Yilin Pang is pursuing the Ph.D. degree in computer science from East University of Science and Technology, Shanghai, China. He is with the school of Information Science and Engineering, East China University of Science and Technology. His current research interests include image processing and data mining.