Xiang and Luo *EURASIP Journal on Advances in Signal Processing* (2017) 2017:59
DOI 10.1186/s13634-017-0496-6

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Efficient reversible data hiding in encrypted image with public key cryptosystem

Shijun Xiang* and Xinrong Luo

Abstract

This paper proposes a new reversible data hiding scheme for encrypted images by using homomorphic and probabilistic properties of Paillier cryptosystem. The proposed method can embed additional data directly into encrypted image without any preprocessing operations on original image. By selecting two pixels as a group for encryption, data hider can retrieve the absolute differences of groups of two pixels by employing a modular multiplicative inverse method. Additional data can be embedded into encrypted image by shifting histogram of the absolute differences by using the homomorphic property in encrypted domain. On the receiver side, legal user can extract the marked histogram in encrypted domain in the same way as data hiding procedure. Then, the hidden data can be extracted from the marked histogram and the encrypted version of original image can be restored by using inverse histogram shifting operations. Besides, the marked absolute differences can be computed after decryption for extraction of additional data and restoration of original image. Compared with previous state-of-the-art works, the proposed scheme can effectively avoid preprocessing operations before encryption and can efficiently embed and extract data in encrypted domain. The experiments on the standard image files also certify the effectiveness of the proposed scheme.

Keywords: Reversible data hiding, Image encryption, Public key cryptosystem, Homomorphic property, Probabilistic property, Paillier cryptosystem

1 Introduction

Reversible data hiding (RDH) is an efficient technology for cover authentication or content integrity verification [1–3] in which the hidden data can be extracted completely and the original carrier can be recovered losslessly after data extraction. With this property, reversible data hiding is widely applied in such areas as medical imagery, military imagery, and law forensics. Reversible data hiding algorithms are generally classified into three categories by using lossless compression [4, 5], difference expansion [6, 7], and histogram shifting [8, 9]. RDH technique is further developed by making good use of new prediction error algorithms and embedding strategies [10–12], which have higher payload and better image quality.

For the sake of information security and privacy protection, data are usually encrypted before upload and transmission. Encryption, which is one of the most powerful method for content protection, can convert original data into unintelligible ciphertexts. It is especially useful in scenarios that content owner is unwilling to expose their sensitive data or private information to untrusted channel and database. On the administrator side, additional data such as content-related keywords, information features, or authentication data are desired to be embedded directly into the encrypted data for the convenience of management and security protection. What is more, content owner still expects that the original cover can be restored perfectly since any distortion is tolerated in many applications. To this end, reversible data hiding in encrypted images (RDHEI) emerges. RDHEI is a technology which implements data hiding procedure in encrypted domain and can recover the original plaintext without error after decryption and data extraction. Owning to this merit, RDHEI has been a research hotspot in information security community recently. In [13], Zhang encrypted image with a standard stream cipher and further divided the encrypted image into blocks, each of which contains two

*Correspondence: shijun_xiang@qq.com
School of Information Science and Technology, Jinan University, Guangzhou 510632, China

groups. One bit can be embedded by flipping three LSBs of each pixel in the corresponding group. Zhang's method was improved by using a different estimation equation so as to better exploit the spatial correlation [14] and different flipping rates [15]. In order to extract the hidden data directly in encrypted domain, Zhang proposed a separable RDHEI which compresses the LSBs of encrypted image with a source coding method to accommodate additional data [16]. Besides, RDHEI for JPEG image has been presented in [17]. To improve the embedding performance, spare room for data hiding was created by implementing self-embedding or compression before encryption [18, 19]. In [20], Qian et al. vacated room after encryption with a distributed source encoding operation, which can avoid to pre-process original image before encryption and has a high capacity. The RDHEI methods mentioned above focus on symmetric cryptosystem. The data encrypted with stream cipher symmetric cryptosystem is hardly for the cloud computing.

Different from symmetric cryptosystem, public key cryptosystem with probabilistic and homomorphic properties, which allows us to conduct operations directly on ciphertexts to generate the expected result, is more powerful in secure signal processing. For example, mass data of an enterprise can be uploaded to the cloud after encryption for statistical analysis services from the third party, e.g., the enterprise can get corresponding annual turnover (sum of plain data) and growth (difference of plain data) from the cloud without any information leakage by utilizing Paillier cryptosystem [21]. Secure signal processing with public key cryptosystem can bring us great benefits.

In the literature, there are a few works on signal processing in encrypted domain with public key cryptosystem [22–26], and two of them have applied Paillier public key cryptosystem for reversible data hiding [25, 26]. In [25], Chen et al. first raised a novel RDHEI method with public key cryptography by dividing a pixel into two parts: LSB and the rest. Image owner encrypted these two parts with Paillier mechanism and transmitted them to data hider. In this way, the transmitted data in size will be twice of the directly encrypted image. One bit can be embedded in encrypted domain by modifying magnitude relationships between the encrypted LSB values of two adjacent pixels with homomorphic multiplication. Since the magnitude relationships can not be kept for the corresponding ciphertexts, the hidden data cannot be extracted before decryption. Aiming at retrieving the hidden data in encrypted domain, Zhang et al. proposed a new reversible data hiding algorithm with public key cryptosystem [26] by using a pre-procession operation to shrink histogram of original image to vacate room for data hiding. After encryption, data hider first embedded additional data into the reserved room so that additional data can be extracted in the plaintext domain. In order to extract data in the encrypted domain in [26], the authors ingeniously applied probabilistic property of Paillier cryptosystem with multilayer wet paper code (WPC) [27] strategy so that the additional data can be hidden into and extracted from the bit-planes of ciphertexts.

This paper proposes a new reversible data hiding scheme in the homomorphic encrypted domain with public key cryptosystem. The main idea is that groups of two selected pixels are encrypted in a way that two pixels in a group are encrypted with two selected parameters by exploiting the probabilistic property. In such a way, the absolute differences of groups of two pixels are reserved and can be retrieved from encrypted domain by implementing a modular multiplicative inverse method and looking for a mapping table. After that, additional data can be embedded into encrypted image by using the homomorphic property to shift histogram of the absolute differences. On the receiver side, legal receiver can extract the marked histogram in encrypted domain in the same way as data hiding procedure. The hidden data can be extracted from the marked histogram and the encrypted image without hiding data can be restored by using an inverse operations of histogram shifting. Besides, the marked absolute difference can be computed after decryption, and then data extraction and image restoration operations can be accomplished perfectly. By using standard example images as data set, extensive testing has been implemented for performance evaluation. The proposed scheme has the following highlights:

1. Preprocessing operations on original image for data hiding can be effectively avoided before encryption. This is beneficial to protect image content privacy in preprocessing procedure. More importantly, the proposed scheme is more scalable due to the avoidance of preprocessing operations.
2. The data can be directly inserted into and extracted from encrypted images with private key. This is useful for cloud computing services while protecting image privacy in the cloud. In the literature, it is a dilemma to extract the hidden data in encrypted domain with public key cryptosystem. In this paper, we have presented a solution.
3. Compared with the previous work [26], the hidden data in the proposed one has stronger security performance in encrypted domain because the data are embedded into plaintexts by using the homomorphic property instead of embedded into the information into bit-planes of ciphertexts [26].

This paper is very useful since the data hiding scheme can be used for both privacy protection and cloud computing due to homomorphic and probabilistic properties of Paillier cryptosystem.

The remainder of this paper is organized as follows. A brief introduction of Paillier cryptosystem is given in Section 2. Then, the details of the proposed reversible data hiding scheme is elaborated in Section 3. Experiment results and performance comparison are given in Section 4. Finally, we draw a conclusion in Section 5.

2 Paillier cryptosystem and modular multiplicative inverse method

2.1 Paillier cryptosystem

Paillier cryptosystem [21], which has homomorphic and probabilistic properties, has been widely used in secure computation fields. In Paillier cryptosystem, a plaintext is encrypted with the same public key and different ciphertexts can be obtained since the parameter is selected randomly. The same plaintext, of course, can be retrieved after decrypting corresponding ciphertexts with private key. Hence, Paillier cryptosystem can achieve semantic security. Besides, homomorphic multiplication is permitted in Paillier cryptosystem, meaning that decrypted product of two ciphertexts is equal to the sum of two corresponding plaintexts. It provides an efficient approach to process the original data in encrypted domain. The following is the details of Paillier cryptography.

Key generation: Select two large primes p and q randomly. Compute $N = pq$ and $\lambda = lcm(p-1, q-1)$, where $lcm(\cdot)$ is a function for calculating the least common multiple of two inputs. And then, randomly select $g \in \mathbb{Z}^*_{N^2}$, which meanwhile satisfies

$$gcd\left(L\left(g^{\lambda} \bmod N^2\right), N\right) = 1 \tag{1}$$

and

$$L(x) = \frac{x-1}{N} \tag{2}$$

where $gcd(\cdot)$ is to derive the greatest common divisor of two inputs, $\mathbb{Z}^*_{N^2}$ is denoted as the subset of $\mathbb{Z}_{N^2}$ in which elements are all relatively prime with N^2 and $\mathbb{Z}_{N^2} = \left\{0, 1, 2, \ldots, N^2-1\right\}$. Finally, we get the public key (N, g) and corresponding private key λ respectively.

Encryption For each original data $m \in \mathbb{Z}_N$, select an integer $r \in \mathbb{Z}^*_N$ randomly. Denote encryption function as $E[\cdot]$. The corresponding ciphertext c can be obtained by

$$c = E[m, r] = g^m r^N \bmod N^2 \tag{3}$$

According to the property of Paillier cryptosystem, the ciphertext c is still in $\mathbb{Z}^*_{N^2}$.

Decryption Denote the decryption function as $D[\cdot]$. With corresponding private key λ, the original plaintext m can be derived by

$$m = D[c] = \frac{L\left(c^{\lambda} \bmod N^2\right)}{L(g^{\lambda} \bmod N^2)} \bmod N \tag{4}$$

Furthermore, the two important properties of Paillier cryptosystem (which have been applied in the proposed method) are introduced as follows:

Lemma one: The function $E[m, r]$ is bijective when g obeys Eq. (1). For two plaintexts $m_1, m_2 \in \mathbb{Z}_N$, compute corresponding ciphertexts c_1, c_2 with r_1, r_2 according to Eq. (1). The Eq. $c_1 = c_2$ holds if and only if $m_1 = m_2$ and $r_1 = r_2$.

Homomorphic multiplication: For $\forall r_1, r_2 \in \mathbb{Z}^*_N$, two plaintexts $m_1, m_2 \in \mathbb{Z}_N$ and corresponding ciphertexts $E[m_1, r_1], E[m_2, r_2] \in \mathbb{Z}^*_{N^2}$ satisfy:

$$E[m_1, r_1] \cdot E[m_2, r_2] = g^{m_1+m_2} \cdot (r_1 \cdot r_2)^N \bmod N^2 \tag{5}$$

and

$$D\left[E[m_1, r_1] \cdot E[m_2, r_2] \bmod N^2\right] = m_1 + m_2 \bmod N \tag{6}$$

2.2 Modular multiplicative inverse method

For better description, we introduce the principle of modular multiplicative inverse (*MMI*) method first. For two coprime integers y and z, there exists an integer Θ satisfying

$$\Theta \cdot y = 1 \bmod z, \tag{7}$$

where Θ is called as the modular multiplicative inverse of y. With a given y, the corresponding Θ can be computed efficiently via an Extended Euclidean algorithm [28]. According to the properties of MMI, the *division* operation can be implemented after modular operation. Assume another integer x and denote ν as the product of x and y to the modulo z. x can be derived from the product ν by multiplying y's corresponding modular multiplicative inverse Θ,

$$\nu \cdot \Theta = x \cdot y \cdot \Theta = x \bmod z. \tag{8}$$

3 Reversible data hiding scheme with public key cryptosystem

Figure 1 plots the sketch of the proposed scheme, which is composed of four main phases: image encryption, data hiding, data extraction and image restoration. Firstly, image owner selects two pixels as a group with a key K_g to form groups of two pixels for encryption with Paillier cryptosystem by using public key K_p with two selected parameters (r_1 and r_2) for a group. Two pixels in a group are encrypted with r_1 (randomly selected as descripted

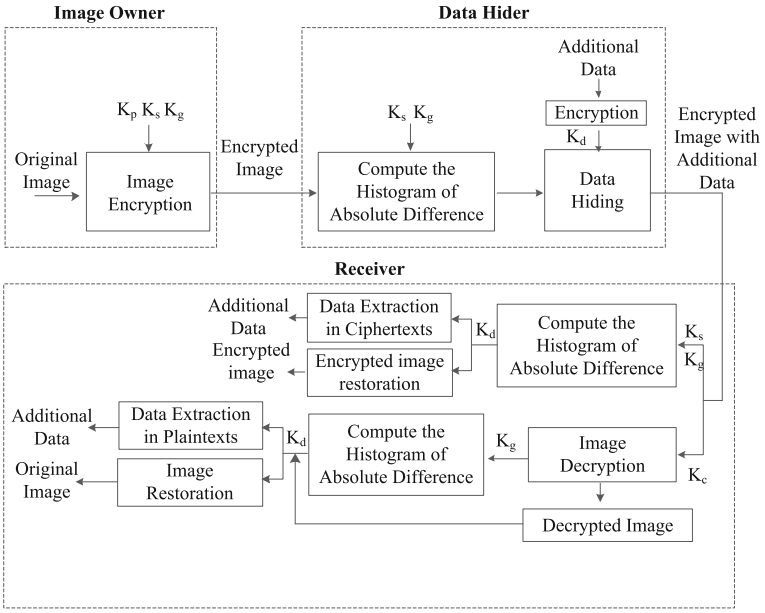

Fig. 1 Sketch of the proposed reversible data hiding scheme with public key cryptography

in Section 2) first. Then, one of the two pixels is further encrypted with r_2 (controlled by a secret key K_s). With K_g and K_s, data hider can calculate absolute differences of groups of two pixels by exploiting modular multiplicative inverse method and looking for a mapping table in encrypted domain. Then, additional data, after encryption, are embedded into the encrypted image by shifting histogram of the calculated absolute differences with a data hiding key K_d. With K_g, K_s, and K_d, receiver can extract the additional data from the histogram of absolute differences (which are computed in the same manner as data hiding procedure). With the private key K_c only, receiver can obtain an image similar to the original image after decryption. With both K_d and K_c, receiver can retrieve the hidden data from the histogram of absolute differences of pixel groups after decryption and recover the original image by implementing an inverse operation of histogram shifting. The remainder of this section elaborates the details of the proposed scheme.

3.1 Image encryption

According to the probabilistic property of Paillier cryptosystem, the parameter $r \in \mathbb{Z}_N^*$ is selected randomly for each plaintext to achieve semantic security. Since magnitude relationships among plaintexts can not be kept to the corresponding ciphertexts, it is still a dilemma to embed additional data directly into an encrypted image with such a cryptosystem. In this work, with the probabilistic property, we design a strategy to encrypt two pixels in a group by using a random parameter so as to reserve the difference of two plaintexts for data hiding in encrypted domain.

Firstly, groups of two selected pixels are chosen from original image I by using a key K_g. Denote the two pixels in k^{th} group as $P_1(k)$ and $P_2(k)$, respectively. Data hider first randomly selects an integer $r_1(k) \in \mathbb{Z}_N^*$ and encrypts $P_1(k)$ and $P_2(k)$ with the public key (N, g) and Paillier cryptosystem.

$$c_1(k) = E\left[P_1(k), r_1(k)\right] = g^{P_1(k)} r_1(k)^N \bmod N^2 \tag{9}$$

$$c_2(k) = E\left[P_2(k), r_1(k)\right] = g^{P_2(k)} r_1(k)^N \bmod N^2 \tag{10}$$

where $c_1(k)$ and $c_2(k)$ are the corresponding ciphertexts of $P_1(k)$ and $P_2(k)$, respectively. To void the effect of using the same parameter for encrypting a group of two pixels, image owner selects another integer $r_2(k) \in \mathbb{Z}_N^*$ with a secret key K_s and implement a homomorphic multiplication operation on $c_2(k)$,

$$c_2(k)' = c_2(k) \cdot E\left[0, r_2(k)\right] = g^{P_2(k)} (r_1(k) \cdot r_2(k))^N \bmod N^2 \tag{11}$$

Let $D\left[c_2(k)'\right]$ be the decrypted version of $c_2(k)'$. According to Eqs. (5) and (6), $D\left[c_2(k)'\right]$ satisfies

$$D\left[c_2(k)'\right] = P_2(k) \bmod N \tag{12}$$

Therefore, $c_2(k)'$ is still an encrypted version of $P_2(k)$. Data hider obtains $c_1(k)$ and $c_2(k)'$ as the cyphertexts of the k^{th} group. Denote the encrypted image as $E[I]$.

3.2 Data hiding

With the encrypted image $E[I]$ and the group selection key K_g and the secret key K_s, data hider can retrieve ciphertext of the absolute difference of two plaintexts with a modular multiplicative inverse method, and then extract the absolute difference from the corresponding ciphertexts by looking for a mapping table (see Fig. 2) without private key K_c. Additional data can be embedded into $E[I]$ by shifting histogram of the absolute differences with a data hiding key K_d. Firstly, data hider divides $E[I]$ into groups in the same manner as image encryption procedure, and then retrieves the ciphertext group $\{c_1(k), c_2(k)'\}$.

3.2.1 Data embedding

With the property of MMI, ciphertext of the absolute difference between the two plaintexts $P_1(k)$ and $P_2(k)$ in the k^{th} group can be calculated in encrypted domain. As described in Section 2, ciphertext $c = E[m, r]$ can be obtained for each $m \in \mathbb{Z}_N$ according to Eq. (3) and $c \in \mathbb{Z}^*_{N^2}$. It means that $E[0, r_2(k)]$, $c_1(k)$ and $c_2(k)$ are coprime to N^2. Denote the modular multiplicative inverse of $E[0, r_2(k)]$ as Θ_{Er}. Data hider first calculates $\Theta_{Er}(k)$ with the Extended Euclidean algorithm and $\Theta_{Er}(k)$ satisfies

$$\Theta_{Er}(k) \cdot E[0, r_2(k)] = 1 \bmod N^2. \tag{13}$$

Since $c_2(k)'$ is the product of $c_2(k)$ and $E[0, r_2(k)]$ according to Eq. (11), data hider derives $c_2(k)$ by multiplying $c_2(k)'$ with $\Theta_{Er}(k)$,

$$c_2(k)' \cdot \Theta_{Er} = c_2(k) \cdot E[0, r_2(k)] \cdot \Theta_{Er}(k) = c_2(k) \bmod N^2. \tag{14}$$

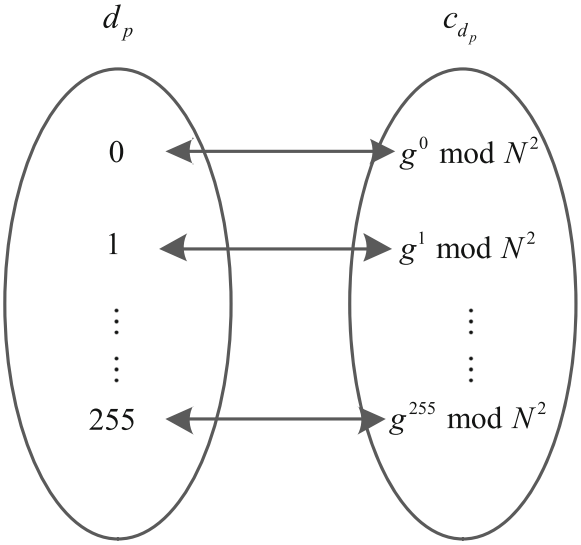

Fig. 2 Mapping table between an absolute difference and its ciphertext

Furthermore, data hider computes the corresponding modular multiplicative inverse of $c_1(k)$ and $c_2(k)$, denoted as $\Theta_1(k)$ and $\Theta_2(k)$ respectively. $\Theta_1(k)$ and $\Theta_2(k)$ satisfy

$$\begin{cases} \Theta_1(k) \cdot c_1(k) = \Theta_1 \cdot g^{P_1(k)} \cdot r_1(k)^N = 1 \bmod N^2 \\ \Theta_2(k) \cdot c_2(k) = \Theta_2 \cdot g^{P_2(k)} \cdot r_1(k)^N = 1 \bmod N^2 \end{cases} \tag{15}$$

With $\Theta_1(k)$ and $\Theta_2(k)$, data hider calculates $c_1 k \cdot \Theta_2(k)$, $c_2 k \cdot \Theta_1(k)$, and the results are denoted as $c_{d_1}(k)$, $c_{d_2}(k)$,

$$\begin{cases} c_{d_1}(k) = c_1(k) \cdot \Theta_2(k) = g^{P_1(k)} \cdot r_1(k)^N \cdot \Theta_2(k) \bmod N^2 \\ c_{d_2}(k) = c_2(k) \cdot \Theta_1(k) = g^{P_2(k)} \cdot r_1(k)^N \cdot \Theta_1(k) \bmod N^2 \end{cases} \tag{16}$$

According to Carmichael's theorem, Eq. (17) holds for any $\alpha \in \mathbb{Z}^*_{N^2}$ [21]

$$\alpha^{N\lambda} = 1 \bmod N^2, \tag{17}$$

where N is the product of two large primes p and q and $\lambda = lcm(p-1, q-1)$. Hence, in the Paillier cryptosystem described in Section (2), $g \in \mathbb{Z}^*_{N^2}$ and $r_1 \in \mathbb{Z}^*_N$ satisfy

$$\begin{cases} g^{N\lambda} = 1 \bmod N^2 \\ r_1^{N\lambda} = 1 \bmod N^2 \end{cases} \tag{18}$$

and

$$g^{N\lambda} \cdot r_1^{N\lambda} = 1 \bmod N^2 \tag{19}$$

Combining Eq. (15) with Eq. (19), we can derive specific expressions of $\Theta_1(k)$ and $\Theta_2(k)$

$$\begin{cases} \Theta_1(k) = g^{N\lambda - P_1(k)} \cdot r_1^{N(\lambda-1)} \bmod N^2 \\ \Theta_2(k) = g^{N\lambda - P_2(k)} \cdot r_1^{N(\lambda-1)} \bmod N^2 \end{cases} \tag{20}$$

With Eq. (18) and the expressions of $\Theta_1(k)$, $\Theta_2(k)$, Eq. (16) can be further simplified as (21) and (22) by considering the magnitude relation of $P_1(k)$ and $P_2(k)$,
if $P_1(k) \geq P_2(k)$

$$\begin{cases} c_{d_1}(k) = g^{P_1(k) - P_2(k)} \bmod N^2 \\ c_{d_2}(k) = g^{N\lambda + P_2(k) - P_1(k)} \bmod N^2 \end{cases} \tag{21}$$

else

$$\begin{cases} c_{d_1}(k) = g^{N\lambda + P_1(k) - P_2(k)} \bmod N^2 \\ c_{d_2}(k) = g^{P_2(k) - P_1(k)} \bmod N^2 \end{cases} \tag{22}$$

As a result, $c_{d_1}(k)$ in Eq. (21) and $c_{d_2}(k)$ in Eq. (22) are the encrypted version of the absolute difference between $P_1(k)$ and $P_2(k)$. Denote probable value of the absolute difference as d_p. Since data hider has the public key $K_p = (N, g)$ and d ranges from 0 to 255, a one-to-one mapping table between d_p and $c_d = g^{d_p}$ can be obtained by

$$c_{d_p} = g^{d_p} \bmod N^2, d_p = 0, 1, \ldots, 255 \tag{23}$$

The mapping table is given in Fig. 2. Without the private key K_c, data hider can look for this table to get a match of $c_{d_1}(k)$ if $P_1(k) \geq P_2(k)$ or $c_{d_2}(k)$ if $P_1(k) < P_2(k)$. Denote

the matched result as $d(k)$, which is the corresponding absolute difference between the two plaintexts $P_1(k)$ and $P_2(k)$. Besides, data hider can get the magnitude relationships between $P_1(k)$ and $P_2(k)$. If $c_{d_1}(k)$ is matched with the mapping table, we have $P_1(k) \geq P_2(k)$; Otherwise, we have $P_2(k) > P_1(k)$. Consequently, a histogram of the absolute differences between groups of two plaintexts is obtained as shown in Fig. 3a.

Additional data can be embedded into the encrypted image $E[I]$ by shifting histogram of the absolute differences in several rounds. Assume n_e rounds are implemented and EP is the embedding bin in each round in referring to data hiding key K_d. To achieve a better performance, we suggest to select the peak point of the histogram as EP in each round. Data hiding procedure is accomplished as follows:
If $P_1(k) \geq P_2(k)$,

$$c_1^w(k) = \begin{cases} c_1(k) \cdot g^w = g^{P_1(k)+w} \cdot r_1^N, & if\ d(k) = EP \\ c_1(k) \cdot g^1 = g^{P_1(k)+1} \cdot r_1^N, & if\ d(k) \geq EP+1 \\ c_1(k), & else \end{cases} \tag{24}$$

$$c_2^w(k) = c_2(k) \tag{25}$$

else

$$c_1^w(k) = c_1(k) \tag{26}$$

$$c_2^w(k) = \begin{cases} c_2(k) \cdot g^w = g^{P_2(k)+w} \cdot r_1^N, & if\ d(k) = EP \\ c_2(k) \cdot g^1 = g^{P_2(k)+1} \cdot r_1^N, & if\ d(k) \geq EP+1 \\ c_2(k), & else \end{cases} \tag{27}$$

where $c_1^w(k)$ and $c_2^w(k)$ are the ciphertexts after data hiding in k^{th} group and w is one bit of the additional data. Denote decrypted versions of $c_1^w(k)$ and $c_2^w(k)$ as $P_1^w(k)$ and $P_2^w(k)$ respectively. The effect of data hiding on plaintexts is to change $P_1(k)$, $P_2(k)$ to $P_1^w(k)$, $P_2^w(k)$:
If $P_1(k) \geq P_2(k)$,

$$P_1^w(k) = \begin{cases} P_1(k) + w, & if\ d(k) = EP \\ P_1(k) + 1, & if\ d(k) \geq EP+1 \\ P_1(k), & else \end{cases} \tag{28}$$

$$P_2^w(k) = P_2(k) \tag{29}$$

else

$$P_1^w(k) = P_1(k) \tag{30}$$

$$P_2^w(k) = \begin{cases} P_2(k) + w, & if\ d(k) = EP \\ P_2(k) + 1, & if\ d(k) \geq EP+1 \\ P_2(k), & else \end{cases} \tag{31}$$

Here, we denote $P_1^w(k)$ and $P_2^w(k)$ as the marked versions of $P_1(k)$ and $P_2(k)$. Obviously, we keep the relative magnitude of two pixels in each group unchanged after data hiding: $P_1^w(k) \geq P_2^w(k)$ if $P_1(k) \geq P_2(k)$ or $P_1^w(k) < P_2^w(k)$ if $P_1(k) < P_2(k)$. Denote the absolute difference after data hiding as $d_w(k)$. According to Eqs. (28), (29), (30) and (31), when $d(k) \geq EP+1$, let $d_w(k) = d(k)+1$ create room for data hiding, as shown in Fig. 3b. Let $d_w(k) = d(k) + w$ hide the additional data if $d(k) = EP$, as shown in Fig. 3c. If the receiver has the EP in each round, he/she can extract the hidden data and recover the original pixels with the inverse operation of embedding procedure.

Finally, data hider multiplies $c_2^w(k)$ with corresponding $E[0, r_2(k)]$ to encrypt the absolute difference properties according to K_s,

$$c_2^w(k)' = c_2^w(k) \cdot E[0, r_2(k)] \mod N^2 \tag{32}$$

Denote the encrypted image with hidden data as $E[I_w]$.

By employing Paillier cryptosystem to encrypt a group of two pixels with two selected parameters, we have successfully reserved the absolute differences of plaintexts

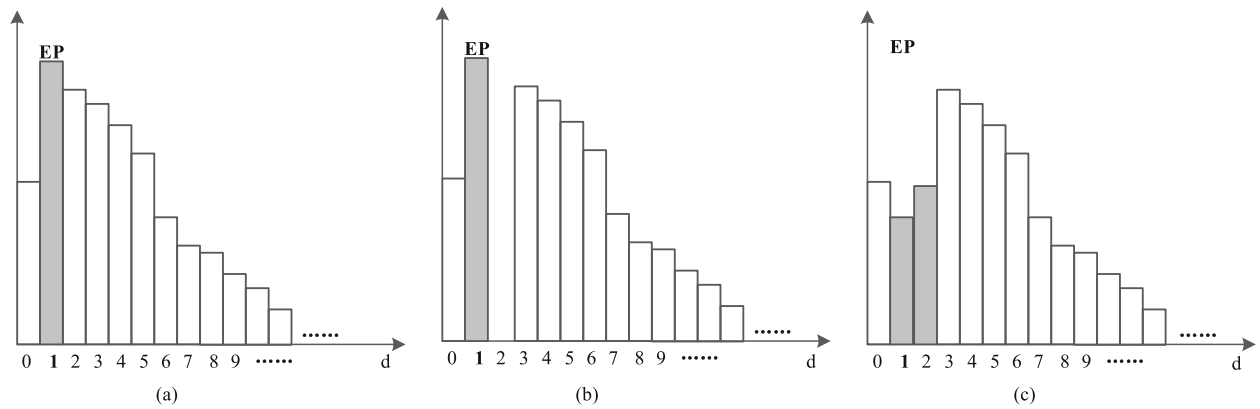

Fig. 3 Data hiding by shifting the histogram of absolute differences. **a** Histogram of absolute difference. **b** Creating rooms for data hiding. **c** Data hiding with histogram shifting

in encrypted domain. Data hider can retrieve the absolute differences and embed additional data directly into the encrypted image by shifting histogram of the absolute differences.

3.2.2 Security of image content in encrypted domain

Security of image content in encrypted domain is an important aspect that users are most concerned about. Here, we discussed the security problem by considering whether an attacker is with the secret key K_s:

Case one: If the user does not have K_s, there is no extra information can be extracted from the encrypted image since $P_1(k)$ and $P_2(k)$ were encrypted with two random integers $r_1(k)$ and $r_1(k) \cdot r_2(k)$. In this case, the proposed image encryption strategy has the same security performance as the standard encryption method in Eq. (3).

Case two: If someone (like the data hider) has the secret key K_s, he or she can extract the absolute differences of groups of two plaintext pixels. In this case, the differences of groups of two adjacent pixels may expose the high-frequency information of original image, as shown in Fig. 4b. To solve this problem, we apply the following two group selection strategies:

Strategy 1: Select two adjacent pixels as a group for encryption. This is followed by a permutation operation on groups of two encrypted pixels. With the permutation operation, the high-frequency information can be effectively covered, as shown in Fig. 4c. Correspondingly, an inverse operation of permutation need to be implemented during the image restoration procedure.

Strategy 2: Randomly select two pixels as a group for encryption. The permutation operation in the *strategy 1* is not needed. By randomly select two pixels as a group, there is no any information leaked from the absolute differences, as shown in Fig. 4d.

3.3 Data extraction and image restoration

In the proposed scheme, data extraction and image restoration can be completed together by inverse operations of histogram shifting. For a receiver, there are two ways to extract the hidden data and restore original ciphertext image or plaintext image.

3.3.1 Extract hidden data and restore original ciphertext image in encrypted domain

With the data hiding key K_d and secret key K_s, receiver can retrieve the hidden data w and recover the directly

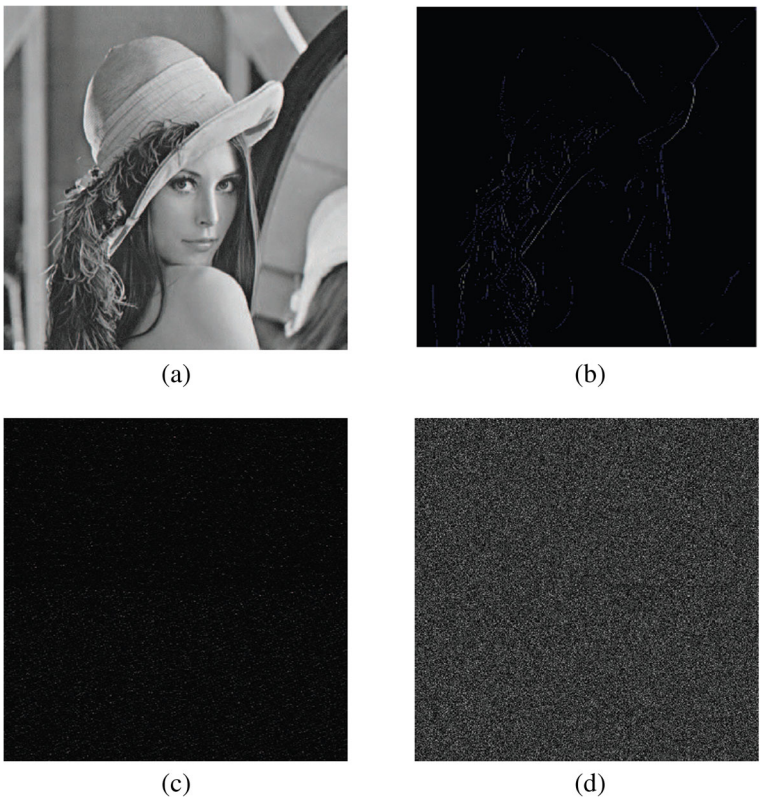

Fig. 4 a Original image. **b** Absolute differences of groups of two adjacent pixels. **c** Absolute differences of groups of two adjacent pixels after a permutation operation on the encrypted image. **d** Absolute differences of groups of two randomly selected pixels

encrypted image $E[I]$ from the received image $E[I_w]$. The extraction and restoration procedures are described as follows:

1. For a legal receiver, the same group selection operation is first implemented on the received ciphertext image $E[I_w]$. Denote the k^{th} ciphtext group by $\left(c_1^w(k), c_2^w(k)'\right)$, which contains the hidden data w.
2. With the secret key K_s, receiver can get the corresponding parameter $r_2(k)$, and then compute $E[0, r_2(k)]$ referring to Eq. (3). Furthermore, $\Theta_{Er}(k)$, which is modular multiplicative inverse of $E[0, r_2(k)]$, is calculated with Extended Euclidean algorithm. Referring to Eqs. (13) and (32), $c_2^w(k)$ (which is the ciphertext of $P_2^w(k)$) is derived by

$$c_2^w(k) = c_2^w(k)' \cdot \Theta_{Er}(k) \bmod N^2 \tag{33}$$

Using the same method, receiver further computes $\Theta_1^w(k)$ and $\Theta_2^w(k)$, which are the corresponding modular multiplicative inverse of $c_1^w(k)$ and $c_2^w(k)$. Similar to extract the absolute difference in the data hiding procedure, receiver calculates $c_{d_1}^w(k)$ and $c_{d_2}^w(k)$ by referring to Eq. (16)

$$\begin{cases} c_{d_1}^w(k) = c_1^w(k) \cdot \Theta_2^w(k) \bmod N^2 \\ c_{d_2}^w(k) = c_2^w(k) \cdot \Theta_1^w(k) \bmod N^2, \end{cases} \tag{34}$$

and computes the mapping table beforehand by referring to Eq. (23). After searching $c_{d_1}^w(k)$ and $c_{d_2}^w(k)$ in the mapping table, we can retrieve $d^w(k)$, which is the absolute difference of the k^{th} marked group $P_1^w(k)$ and $P_2^w(k)$). This can be done to obtain the histogram of the absolute differences of all the groups. Besides, the magnitude relationships between $P_1^w(k)$ and $P_2^w(k)$) are obtained by: If $c_{d_1}^w(k)$ is matched with the table, $P_1^w(k) \geq P_2^w(k)$; Otherwise, $P_2^w(k) > P_1^w(k)$.
3). With the data hiding key K_d, receiver can obtain the rounds of embedding n_e and the embedding point in each round EP. According to the corresponding EP and the magnitude relationships between $P_1^w(k)$ and $P_2^w(k)$, additional data w can be extracted from the histogram,

$$w = \begin{cases} 0, \ if \ d^w(k) = EP \\ 1, \ if \ d^w(k) = EP + 1 \end{cases} \tag{35}$$

Accompanied with data extraction, the directly encrypted image $E[I]$ can be recovered without any error. Receiver calculates the modular multiplicative inverse of g via Extended Euclidean algorithm, denoted as Θ_g. Θ_g satisfies

$$\Theta_g \cdot g = 1 \bmod N^2 \tag{36}$$

According to the property of MMI, ciphertexts $c_1(k)$ and $c_2(k)$ before hiding data can be restored as follows:
If $P_1^w(k) \geq P_2^w(k)$,

$$c_1(k) = \begin{cases} c_1^w(k) \cdot \Theta_g \bmod N^2 & if \ d^w(k) \geq EP + 1 \\ c_1^w(k), & else \end{cases} \tag{37}$$

$$c_2(k) = c_2^w(k) \tag{38}$$

else

$$c_1(k) = c_1^w(k) \tag{39}$$

$$c_2(k) = \begin{cases} c_2^w(k) \cdot \Theta_g \bmod N^2, & if \ d^w(k) \geq EP + 1 \\ c_2^w(k), & else \end{cases} \tag{40}$$

where $c_1^w(k) \cdot \Theta_g$ in Eq. (37) is equal to $g^{P_1^w(k)-1} \cdot r_1(k)^N$ and $c_2^w(k) \cdot \Theta_g$ in Eq. (39) is equal to $g^{P_2^w(k)-1} \cdot r_1(k)^N$.Therefore, inverse operations of histogram shifting can be implemented.
4). After n_e rounds of data extraction and inverse operations of histogram shifting, the hidden data can be extracted perfectly and the original ciphertext image $E[I]$ can be restored.

3.3.2 *Extract the hidden data and restore the original image after decryption*

With the private key K_c and data hiding key K_d, receiver can extract the hidden data and recover the original image after decryption.

As described in Section 2, the ciphertext group $\left\{c_1^w(k), c_2^w(k)\right\}$ in the received encrypted image $E[I_w]$ can be decrypted by private key $K_c = \lambda$, formulated as

$$P_1^w(k) = D\left[c_1^w(k)\right] = \frac{L\left(c_1^w(k)^\lambda \bmod N^2\right)}{L\left(g^\lambda \bmod N^2\right)} \bmod N \tag{41}$$

and

$$P_2^w(k) = D\left[c_2^w(k)\right] = \frac{L\left(c_2^w(k)^\lambda \bmod N^2\right)}{L(g^\lambda \bmod N^2)} \bmod N \tag{42}$$

where $P_1^w(k)$ and $P_2^w(k)$ are the corresponding plaintexts of $c_1^w(k)$ and $c_2^w(k)$. This can be done for all the groups to obtain a decrypted version of $E[I_w]$, denoted as I_w. In the applications, there are two possible scenarios on the decrypted result I_w.

Scenario 1: The receiver extracts the hidden data and recover the original image immediately. Firstly, the receiver divide I_w into groups in the same manner as in the image encryption procedure and get $P_1^w(k)$ and $P_2^w(k)$. Then, $d^w k$ (the absolute difference between $P_1^w(k)$ and $P_2^w(k)$) is calculated

$$d^w(k) = \left|P_1^w(k) - P_2^w(k)\right| \tag{43}$$

In this way, the receiver can obtain histogram of the absolute differences. With the data hiding key K_d, the receiver can get n_e (embedding rounds) and PZ (embedding point in each round). Referring to PZ, the hidden data w can be retrieved by

$$w = \begin{cases} 0, \ if \ d^w(k) = EP \\ 1, \ if \ d^w(k) = EP + 1. \end{cases} \tag{44}$$

Meanwhile, original image pixels can be recovered by the following formulations.
If $P_1^w(k) \geq P_2^w(k)$,

$$P_1(k) = \begin{cases} P_1^w(k) - 1, \ if \ d^w(k) \geq EP + 1 \\ P_1^w(k), \qquad else \end{cases} \tag{45}$$

$$P_2(k) = P_2^w(k) \tag{46}$$

else

$$P_2(k) = \begin{cases} P_2^w(k) - 1, \ if \ d^w(k) \geq EP + 1 \\ P_2^w(k), \qquad else \end{cases} \tag{47}$$

$$P_1(k) = P_1^w(k) \tag{48}$$

After n_e rounds of data extraction and image restoration, the hidden data can be retrieved perfectly and the original image can be restored without any error.

Scenario 2: Receiver wants to store or deliver a marked version of plaintext image. If needed, the hidden data can be extracted and the original image can be recovered. This case is possible in cloud computing application. For example, a company may upload the encrypted image to the cloud for watermarking service from the third party. After decryption, the company can deliver the watermarked image to legal users for business. The user can get the original image according to the data hiding key K_d and the hidden data can be extracted for copyright authentication.

Due to the effect of data hiding, pixel oversaturation in plaintext domain could be caused. Denote $P^w(i,j)$ by a plaintext in I_w, where (i,j) indicates the position. Receiver can scans I_w in order to find overflowed pixels (if $P^w(i,j) \geq 255$). Then, the overflowed part of the $P^w(i,j)$ can be calculated as

$$o(l) = P^w(i,j) - 255 \tag{49}$$

where $P^w(i,j)$ is the l^{th} plaintext which is greater than or equal to 255, and $o(l)$ is the overflowed part of $P^w(i,j)$.

Since the effect of data hiding in each round is to add 1 or a binary bit to the corresponding plaintexts, receiver uses L $\left(L = \lceil log_2(n_e) \rceil\right)$ bits to represent $o(l)$ as $o_1(l), o_2(l), \ldots, o_L(l)$,

$$o_i(l) = \left\lfloor \frac{o(l)}{2^{i-1}} \right\rfloor \bmod 2, i = 1, 2, \ldots, L. \tag{50}$$

When there are Q overflowed pixels, we can get a L_{S_a} ($L_{S_a} = L \times Q$) bits of sequence, denoted by S_a.

For the overflow problem, we can modify pixels greater than 255 as 255 and restore their position information. All the appendix information including L, S_a and the positions of the overflowed pixels, can be reversibly embedded into I_w to generate I_w'. In this way, receiver could deliver I_w' to his clients for business. For copyright problem, the receiver can extract the inserted appendix information and reconstruct I_w. With the data hiding key, the hidden data w can be extracted from I_w and the original image I can be restored, as described in *Scenario 1* above.

4 Experimental results

In Section 3.2.2, we have presented two group selection strategies. The two group selection strategies is applied to form the proposed two reversible data hiding methods:

Method 1: We choose two adjacent pixels as a group for encryption. This is followed by a permutation operation on the encrypted image so as to avoid the difference of the adjacent pixels to leak image content.

Method 2: We randomly select two pixels as a group for encryption. The permutation operation in the *Method 1* is not needed.

In this section, we will discuss the performance of the proposed reversible data hiding methods with several standard images (see Fig. 5). Then, the embedding distortion of the proposed methods is evaluated by using *Lena* as example image. Last part is a comprehensive comparison of the proposed two methods against previous schemes with Paillier cryptosystem.

4.1 Embedding distortion

Embedding distortion in the proposed scheme is from histogram shifting operation in encrypted domain. As embedding capacity increases, the more groups are needed and more rounds of histogram shifting will be implemented for data embedding. As a result, the distortion increases.

With the Method 1, the PSNR value of *Lena* is 44.914 dB for 40,960 bits of additional information, and the distortion is imperceptible, as shown in Fig. 6b. The basic reason is that a great number of the absolute differences of groups

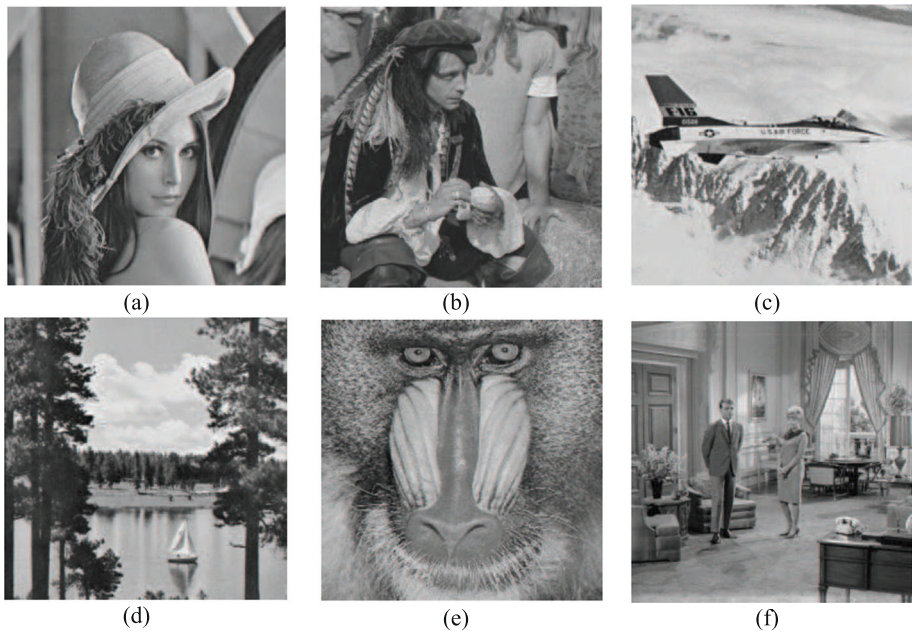

Fig. 5 Six standard test images sized by 512 × 512. **a** Lena. **b** Man. **c** Plane. **d** Lake. **e** Baboon. **f** Room

of two adjacent pixels range from 0 to 4 due to strong correlation among neighboring pixels. In the Method 2, since two randomly selected pixels in magnitude are often different, only a small number of the absolute differences are smaller than 4 in magnitude. For 4960 bits of additional data, the PSNR value with the Method 2 is 41.754 dB, as shown in Fig. 6c.

4.2 Performance comparison

In the literature, there are two existing works to reversibly hide data into Homomorphic encrypted image [25, 26] by using Paillier cryptosystem (which is a public key cryptosystem with both homomorphic and probabilistic properties). With Paillier cryptosystem, one of the advantages is that the encrypted image can be processed by the third party in encrypted domain.

Table 1 lists a comprehensive comparison results of the proposed methods against the two previous schemes by considering (1) if a preprocessing operation is needed before encryption, (2) if the image after encryption is bigger than the directly encrypted image, (3) if data extraction and image restoration is separable, and (4) security of hidden data in encrypted domain, which are described as follows:

1. To reserve room before encryption for data hiding in encrypted domain is good idea proposed by Ma et al. in [18]. In [25, 26], preprocessing image before encryption is important step so that additional can be hidden into encrypted image. In the proposed scheme, the use of two selected pixels as a group is directly encrypted with a genuine encryption strategy

Fig. 6 Embedding distortion (in PSNR) with two group selection methods. **a** Lena. **b** Embedding 40,960 bits with Method 1 (44.914 dB). **c** Embedding 4960 bits with Method 2 (41.754 dB)

Table 1 Comprehensive comparison between the proposed schemes and two previous methods

	Features			
Methods	Preprocessing	Extra data expansion	Separable ?	Protection mechanism of the hidden data
Method [25]	Yes	Yes	No	Homomorphic and probabilistic mechanism
Method [26]	Yes	No	Yes	WPC mechanism
The proposed schemes	No	No	Yes	Homomorphic and probabilistic mechanism

to transmit the absolute differences of groups of the two plaintext pixels to encrypted domain for data hiding. In other words, without any preprocessing operations are needed in the proposed scheme. This is beneficial to reduce risk of image content exposed. Besides, the scheme without preprocessing has better scalability.

2. The method [25] introduced an extra data expansion due to the fact that original pixel has been divided into two parts (LSB and the rest) for encryption.

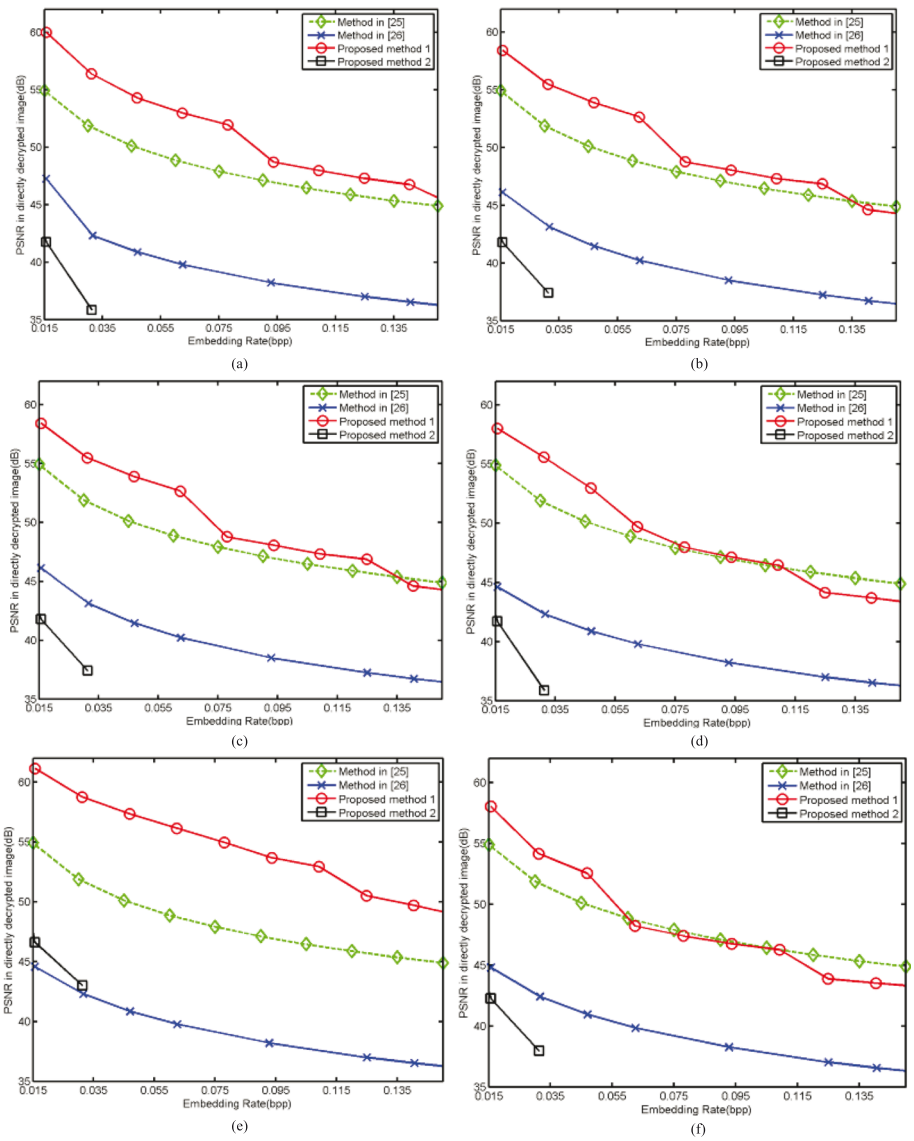

Fig. 7 Comparison of Embedding capacity versus embedding distortion in six different images against two existing works. **a** Lena. **b** Man. **c** Plane. **d** Lake. **e** Baboon. **f** Room

Such a way will make delivered ciphertext data double. The method [26] and the proposed scheme has no extra data expansion introduced.

3. The proposed method embedded additional data by shifting histogram of the absolute differences of groups of two pixels in encrypted domain. Since the histogram can be derived in encrypted domain, data extraction is separable from the content decryption. The method in [25] is on the opposite side, where the data can not be extracted before encryption. In [26], additional data were inserted twice in a reversible way and a lossless way, so that the data can be extracted in both encrypted and plaintext domains.
4. The security of hidden data in encrypted domain is one of the features that people are most concerned about. Method in [26] utilized WPC and probabilistic properties of Paillier cryptosystem to embed bits into bit-planes of the ciphertexts, where WPC mechanism was applied to protect the hidden data. Since the bit-planes of ciphertext are exposed in encrypted domain, there exists a risk that an illegal client may embed fake information into the bit-planes. Besides, when the same data are embedded into different images, the hidden data could be found in statistical way. In the proposed scheme, additional data are embedded into histogram of the absolute differences of groups of two selected pixels by using homomorphic properties in encrypted domain. Since the hidden data are capsuled with homomorphic and probabilistic properties of Paillier cryptosystem, an illegal receiver without the data hiding key can hardly crack those encrypted pixels, meaning that the hidden data in the proposed scheme is secure in encrypted domain.

Finally, we report the PSNR values of the proposed scheme with two group selection methods and the previous two schemes at different embedding rates. With six standard images (as shown in Fig. 5), we plot the corresponding comparison results of embedding capacity versus embedding distortion. We can see from Fig.7 that the proposed method 1 with the group selection strategy 1 (by selecting two adjacent pixels as a group for encryption followed by a permutation operation) can provide the best performance since the use of two adjacent pixels as a group is beneficial to generate a great number of the differences with smaller magnitude. Also, we can see from this figure that the proposed method 2 with the group selection strategy 2 (by randomly selecting two pixels as a group for encryption) has lower PSNR value at the same embedding rate due to the fact that only a small number of the differences with smaller magnitude can be achieved. Furthermore, we compute the average PSNR values of 50 standard test images at different embedding rates, as shown in Fig. 8. We can see from this figure that the simulation results are similar.

5 Conclusions

In this paper, we proposed a new reversible data hiding scheme in encrypted domain with public key cryptosystem. In the proposed method, groups of two pixels are selected for encryption in a way that two pixels in the same group are encrypted with a random integer r_1 to obtain two ciphertexts first. Then, one of the two ciphertexts is further encrypted with another parameter r_2 (controlled by a key) so that their difference can be transmitted to encrypted domain. Owing to the encryption strategy, data hider can retrieve absolute differences of groups of two plaintext pixels by combining a modular multiplicative inverse method and a mapping table. And, additional data can be embedded into encrypted image by shifting histogram of the retrieved absolute differences by using the homomorphic property. With the proposed embedding strategy, there is no preprocessing on original image before encryption. On the receiver side, legal receiver can extract the modified histogram in encrypted domain with the same method as data hiding procedure. The hidden data can be further extracted from the marked histogram by performing inverse operations of histogram shifting. After decryption, the marked absolute differences can be computed in plaintext domain. Then, data extraction and image restoration can be accomplished perfectly. Experimental results have shown superiority of the proposed scheme by comparing with two previous schemes.

Also, we have discussed the security of original image and the hidden data in encrypted domain, and the conclusion is satisfied. Since the proposed scheme with Paillier public key cryptosystem is separable and without any preprocessing operations on original image, we suggest that

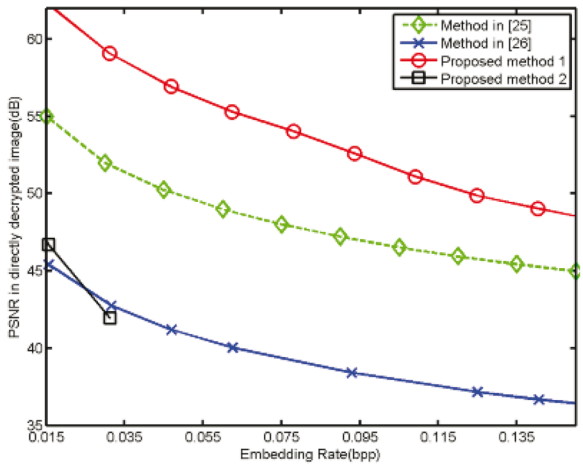

Fig. 8 Comparison of embedding capacity versus average embedding distortion in 50 standard test images against two existing works

the proposed scheme is useful for privacy protection and cloud computing.

Funding
This work was partially supported by the NSFC project (No. 61772234 and 61272414) , co-funded by the State Key Laboratory of Information Security (No. 2016-MS-07).

Authors' contributions
All the authors have participated in writing the manuscript. Both authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 13 February 2017 Accepted: 10 August 2017
Published online: 22 August 2017

References

1. CW Honsinger, P Jones, M Rabbani, JC Stoffel, Lossless recovery of an original image containing embedded data. (US Patent 6278791, 2001)
2. B Macq, in *Proc. the European Signal Processing Conf,*. Lossless multiresolution transform for image authenticating watermarking, (Tampere, 2000), pp. 533–536
3. Y Hu, H Lee, K Chen, J Li, Difference expasion based reversible data hiding using two embedding directions. IEEE Trans. Multimedia. **10**(8), 1500–1512 (2010)
4. J Fridrich, M Goljan, R Du, Lossless data embedding—new paradigm in digital watermarking. Eur.Assoc. Signal Process. J. Appl. Signal Process. **2002**(2), 185–196 (2002)
5. M Celik, G Sharma, A Tekalp, E Saber, Lossless generalized-LSB data embedding. IEEE Trans. Image Process. **14**(2), 253–266 (2005)
6. J Tian, Reversible data embedding using a difference expansion. IEEE Trans. Circ. Syst. Video Technol. **13**(8), 890–896 (2003)
7. DM Thodi, JJ Rodriguez, Expansion embedding techniques for reversible watermarking. IEEE Trans. Image Process. **16**(3), 721–730 (2007)
8. Z Ni, YQ Shi, N Ansari, W Su, Reversible data hiding. IEEE Trans. Circ. Syst. Video Technol. **16**(3), 354–362 (2006)
9. SK Lee, YH Suh, YS Ho, in *Proc. IEEE Int. Conf. Multimedia Expo*. Reversible image authentication based on watermarking, (Toronto, 2006), pp. 1321–1324
10. W Hong, TS Chen, CW Shiu, Reversible data hiding for high quality images using modification of prediction errors. J. Syst. Softw. **82**(11), 1833–1842 (2009)
11. L Luo, et al., Reversible image watermarking using interpolation technique. IEEE Trans. Inf. Forensic Secur. **5**(1), 187–193 (2010)
12. XL Li, B Yang, TY Zeng, Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. IEEE Trans. Image Process. **20**(12), 3524–3533 (2011)
13. X Zhang, Reversible data hiding in encrypted image. IEEE Signal Process. Lett. **18**(4), 255–258 (2011)
14. W Hong, TS Chen, HY Wu, An improved reversible data hiding in encrypted images using side match. IEEE Signal Process. Lett. **19**(4), 199–202 (2012)
15. W Hong, TS Chen, J Chen, YH Kao, Reversible data embedment for encrypted cartoon images using unbalanced bit flipping. Advances on Swarm Intell. Lect. Notes Comput. Sci. **7929**, 208–14 (2013)
16. X Zhang, Separable reversible data hiding in encrypted image. IEEE Trans. Inf. Forensic Secur. **7**(2), 526–532 (2012)
17. Z Qian, X Zhang, S Wang, Reversible data hiding in encrypted JPEG bitstream. IEEE Trans. Multimedia. **16**(5), 1486–1491 (2014)
18. K Ma, W Zhang, et al., Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans. Inf. Forensic Secur. **8**(3), 553–562 (2013)
19. W Zhang, Ma K, N Yu, Reversibility improved data hiding in encrypted images by reserving images. Signal Process. **94**, 174–182 (2014)
20. Z Qian, X Zhang, Reversible data hiding in encrypted image with distributed source encoding. IEEE Trans. Circ. Syst. for Video Technol. **26**(4), 636–646 (2016)
21. P Paillier, Public-key cryptosystems based on composite degree residuosity classes. Proceeding of the, Advances Cryptology. EUROCRYPT99, LNCS. **1592**, 223–238 (1999)
22. T Bianchi, A Piva, M Barni, On the implementation of the discrete fourier transform in the encrypted domain. IEEE Trans. Inf. Forensic Secur. **4**(1), 86–97 (2009)
23. T Bianchi, A Piva, M Barni, Composite signal representation for fast and storage-efficient processing of encrypted signals. IEEE Trans. Inf. Forensic Secur. **5**(1), 180–187 (2010)
24. P Zheng, J Huang, Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain. IEEE Trans. Image Process. **22**(6), 2455–2468 (2013)
25. YC Chen, CW Shiu, G Horng, Encrypted signal-based reversible data hiding with public key cryptosystem. J. Vis. Commun. Image Represent. **25**, 1164–1170 (2014)
26. X Zhang, J Long, Z Wang, H Cheng, Lossless and reversible data hiding in encrypted images with public key cryptography. IEEE Trans. Circ. Syst. for Video Technol. **26**(9), 1622–1631 (2016)
27. J Fridrich, M Goljan, P Lisonek, D Soukal, Writing on wet paper. IEEE Trans. Signal Proc. **53**(10), 3923–3935 (2005)
28. D Knuth, *The art of computer programming*, 3rd edition Chapter 4, vol. 2. (Addison-Wesley, 1997)