Journal of Real-Time Image Processing (2019) 16:697–708
https://doi.org/10.1007/s11554-018-0811-y

SPECIAL ISSUE PAPER

Real-time reversible data hiding in encrypted images based on hybrid embedding mechanism

Wei Zhang[1] · Ping Kong[2] · Heng Yao[1] · Yu-Chen Hu[3] · Fang Cao[4]

Received: 8 April 2018 / Accepted: 23 July 2018 / Published online: 1 August 2018

Abstract
In this paper, we propose a novel real-time scheme of separable reversible data hiding in encrypted images, which consists of image encryption, data embedding, data extraction and image recovery. In image encryption phase, the content owner divides the original image into a number of non-overlapping blocks and encrypts blocks by stream cipher and permutation. During the data embedding phase, the data hider classifies encrypted blocks into smooth region and complex region according to the threshold and replaces the MSB layer of a part of pixels in blocks of smooth region with the secret data. Then, the LSB layers of other pixels are collected and compressed to generate a room for embedding the secret data again. When the receiver receives the marked image, he can divide the marked image into blocks and decrypt them by the encryption key to obtain a similar image with good quality. If the receiver only has the data hiding key, he can classify the blocks into smooth region and complex region according to the threshold and extract the embedded data by the data hiding key. If the receiver has both encryption key and data hiding key, he can extract the embedded data from the marked image and recover the original image perfectly. The proposed scheme can achieve satisfactory quality of decrypted image and high embedding rate. Experimental results demonstrate the effectiveness and computational efficiency of our scheme.

Keywords Image encryption · Reversible data hiding · Image decryption · Image recovery · Real-time

1 Introduction

Recent years, people have paid more attention on the privacy with the development of cloud and internet. Reversible data hiding in encrypted images (RDHEI) is proposed, this technology is useful for protecting the privacy. Now, RDHEI is widely used in many regions, such as military and medical images. For example, when you upload your photos to your cloud space, you can first encrypt the photos to ensure the image security and embed the information of the corresponding photo into the encrypted image for finding it later conveniently.

Data hiding is a useful technology in the field of information protection, so that the adversary can not detect the embedded information [1–4]. Now, digital image has become an important manner of communication, however, sometimes the images may disclose important information in military or medical fields. Based on these respects, people need to protect the security of images and effective manage images [5–8]. An improved dual-image RDH method using the selection strategy of shiftable pixels' coordinates with minimum distortion was proposed in [9].

✉ Fang Cao
fangcao@shmtu.edu.cn

Wei Zhang
wellzhang@yeah.net

Ping Kong
kongp@sumhs.edu.cn

Heng Yao
hyao@usst.edu.cn

Yu-Chen Hu
ychu@pu.edu.tw

[1] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

[2] Shanghai Key Laboratory for Molecular Imaging, Shanghai University of Medicine and Health Sciences, Shanghai 201318, China

[3] Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan

[4] College of Information Engineering, Shanghai Maritime University, No. 1550 Haigang Ave, Shanghai 201306, China

A color-image-dedicated reversible data hiding (RDH) algorithm [10] was proposed to improve embedding performance by applying a guided filtering predictor and an adaptive prediction-error expansion (PEE) scheme. Based on prediction-error expansion (PEE), a new reversible data hiding (RDH) technique for multiple histograms was proposed in [11]. However, the proposed scheme in [12] improved PEE by exploiting the correlations among prediction-errors. In [13], the authors revisited histogram shifting (HS) technique and proposed a framework of HS-based RDH. In [14], a novel prediction-based reversible data hiding scheme was proposed, which was based on image inpainting. By exploiting modification direction (EMD), a new scheme for reversible data hiding was proposed in [15]. Nevertheless, sometimes people do not want others to know the original image content, hence, reversible data hiding or compression in encrypted images was proposed. A novel lossy compression for encrypted images was proposed in [16]. The original image was encrypted by a modulo-256 addition and block permutation, and then the spatial correlation and quantization were used in compression phase. In [17], the content owner encrypted the image by a stream cipher, and the secret data were embedded by flipping the three LSB layers. The receiver can directly decrypt the marked image, extract the secret data and recover the original image. The method in [18] improved the scheme in [17] by the side-match technique, and the authors defined a new fluctuation function for each block. To further improve the performances of [17, 18], Liao et al. proposed a new, more precise function to calculate the complexity of the image blocks by considering the absolute mean difference of multiple neighboring pixels [19]. A resolution progressive compression scheme was proposed in [20], which was used in encrypted image. A new scheme for separable reversible data hiding in encrypted images was proposed in [21], the data hider embedded the secret data into spare room where compressed by matrix operation. In [21], if the receiver only had the encryption key, he can decrypt the marked image directly. If the receiver only had the data hiding key, he can extract the embedded data directly. If the receiver had both the encryption key and data hiding key, he can extract the embedded data and recover the original image perfectly. Base on [21], the method in [22] changed data embedding phase, the data hider divided the encrypted image into three sets, and pseudo-randomly permuted the encrypted pixels in each set. Then, the data hider divided the pixels in each set into segments and used the matrix operation to embed additional data. When the marked image was received, the receiver divided the image recovery procedure into three rounds to recover the image. The discrete Fourier transform (DFT) in the encrypted domain which based on the homomorphic properties of the underlying cryptosystem was proposed in [23]. In [24], the method of data hiding in encrypted image used MSB prediction to achieve high capacity, however, this method needed the pre-processing. The scheme in [25] used LDPC code to compress a part of encrypted image, and the rest were kept unchanged as the side information for image recovery. Based on the distributed source coding, a novel scheme of reversible data hiding in encrypted images was proposed in [26]. In data embedding phase, LDPC code was also used to embed the additional data into the encrypted image. In [27], the original image was encrypted by specific encryption, and then one secret bit was embedded into each block of the encrypted image. The receiver can decrypt the marked image by the encryption key. Based on decrypted image, the secret data can be extracted and the original image can further be recovered by the data hiding key and the smoothness function.

In this paper, we propose a new scheme of separable reversible data hiding in encrypted images and make full use of the relationship of adjacent pixels. The content owner divides the original image into non-overlapping blocks and encrypts them by stream cipher and permutation. The data hider classifies the blocks into smooth region and complex region and embeds the secret data into the encrypted image by two steps: (1) replace the MSB layer of a part of encrypted pixels in smooth region with the secret data, (2) compress the LSB layers of a part of the rest encrypted pixels. If the receiver only has the encryption key, he can decrypt the marked image directly to obtain an approximate image which has a good image quality. If the receiver only has the data hiding key, he can extract the secret data from the marked image. If the receiver has both the encryption key and data hiding key, he can extract the embedded data successfully and recover the image without any error according to the spatial correlation in natural image.

The rest of this paper is organized as follows. Section 2 describes the proposed RDHEI scheme detailedly. Experimental results and comparisons are given in Sect. 3. Section 4 concludes the paper.

2 Proposed scheme

In the proposed scheme, there are four phases: (1) image encryption; (2) data embedding; (3) data extraction; (4) image recovery. Figure 1 shows the framework of the proposed scheme. The content owner divides the original image into a number of non-overlapping blocks with a size of 2×2 and encrypts the blocks by stream cipher and permutation. Then, the data hider classifies the blocks into smooth region and complex region according to a pre-determined threshold T and embeds the data into the encrypted image using two methods; one is that the data hider compresses the least significant bits (LSB) of a part of pixels in the encrypted image using a data hiding key to create a sparse room for

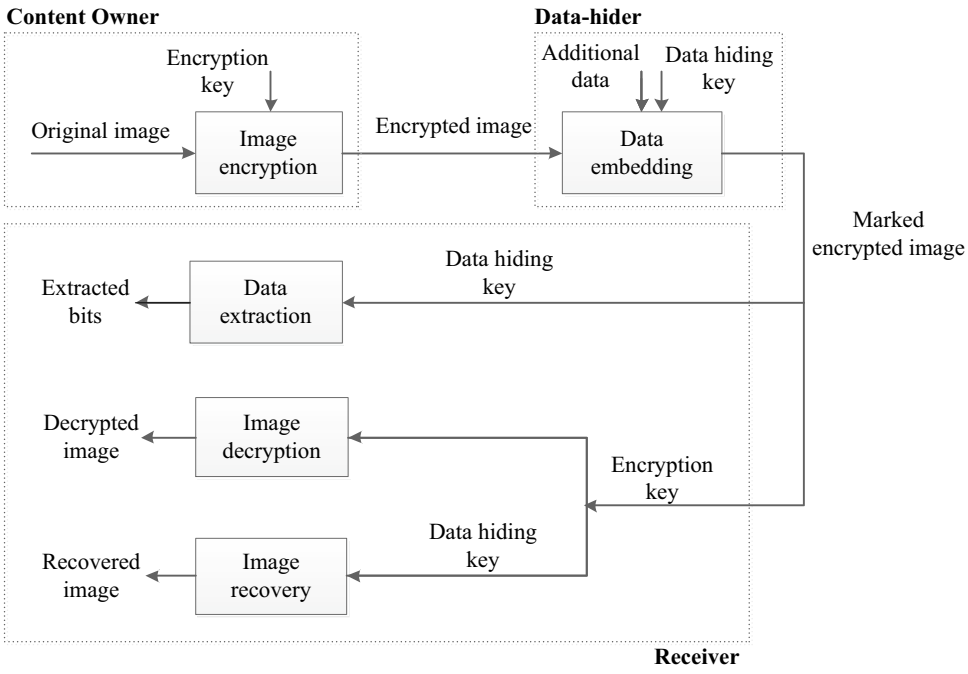


Fig. 1 Framework of the proposed scheme

the additional data, the other is that the most significant bit (MSB) of a part of pixels in smooth region are replaced by the additional data. The marked image is generated after the image encryption and data embedding phases. When the receiver received the marked image, he can decrypt the marked image only with encryption key to obtain an approximate image which has a good image quality. He can also extract the additional data in the marked image only according to the data hiding key. If the receiver has both the encryption key and data hiding key, he can extract the embedded data successfully and recover the image without any error according to the spatial correlation in natural image. The detail of the proposed method is described as follows.

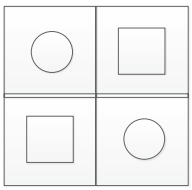
2.1 Image encryption

Assume the original image $\mathbf{I}$ with a size of $M \times N$ is an uncompressed grayscale image and each pixel can be represented by 8 bits. The content owner divides the original image $\mathbf{I}$ into a number of non-overlapping 2×2-sized blocks, and $M \times N/4$ blocks are generated. The blocks are denoted by $\mathbf{B}_{i,j}$ ($1 \le i \le M/2$ and $1 \le j \le N/2$), and the four pixels in $\mathbf{B}_{i,j}$ are ${B_{i,j}}^{(0,0)}$, ${B_{i,j}}^{(0,1)}$, ${B_{i,j}}^{(1,0)}$ and ${B_{i,j}}^{(1,1)}$. As shown in Fig. 2, the pixels ${B_{i,j}}^{(0,0)}$ and ${B_{i,j}}^{(1,1)}$ are represented by "○", ${B_{i,j}}^{(0,1)}$ and ${B_{i,j}}^{(1,0)}$ by "□", respectively.

The content owner decomposes the four pixel values in each block into 8 bits by:

$$b_{i,j}^{(x,y,u)} = \left\lfloor \frac{B_{i,j}^{(x,y)}}{2^u} \right\rfloor \bmod 2, \quad u = 0, 1, \ldots, 7, \tag{1}$$

Fig. 2 Four pixels in block $\mathbf{B}_{i,j}$



where the (i, j) is the block index, and $(x, y) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. In each block, the content owner first encrypts 8 bits of the square pixels and p least significant bits of the circle pixels by pseudo-random bits according to the encryption key:

$$b_{i,j}'^{(x,y,u)} = b_{i,j}^{(x,y,u)} \oplus f_{i,j}^{(x,y,u)}, \quad (x, y, u) \in \mathbf{\Psi}_1 \cup \mathbf{\Psi}_2, \tag{2}$$

$$\mathbf{\Psi}_1 = \{(x, y, u) | (x, y) \in \{(1, 0), (0,1)\} \text{ and } u = 0, 1, 2, \ldots, 7\}, \tag{3}$$

$$\mathbf{\Psi}_2 = \{(x, y, u) | (x, y) \in \{(0, 0), (1,1)\} \text{ and } u = 0, 1, 2, \ldots, p-1\}, \tag{4}$$

where p denotes the number of LSB layers and will be used in data embedding phase. Here, p is a small positive integer and smaller than 4. Then, the content owner needs to encrypt the rest of $(8-p)$ most significant bits of the circle pixels in each block. Another different pseudo-random binary sequence $A_{i,j}$ is generated by the content owner according to the encryption key, if the $(8-p)$ most significant bits of the circle pixels belong to $\mathbf{\Psi}_3$:

$$\begin{cases} b'^{(x,y,u)}_{i,j} = \overline{b^{(x,y,u)}_{i,j}}, & \text{if } A_{i,j} = 1, \\ b'^{(x,y,u)}_{i,j} = b^{(x,y,u)}_{i,j}, & \text{if } A_{i,j} = 0, \end{cases} \quad (x, y, u) \in \mathbf{\Psi}_3, \tag{5}$$

$$\mathbf{\Psi}_3 = \{(x, y, u) | (x, y) \in \{(0, 0), (1,1)\} \text{ and } u = p, p + 1, p + 2, \ldots, 7\}. \tag{6}$$

Then, the encrypted pixels are calculated by:

$$B^{(x,y)}_{i,j} = \sum_{u=0}^{7} [2^u \times b'^{(x,y,u)}_{i,j}]. \tag{7}$$

When the pixels in the image are all encrypted, the content owner permutes the blocks pseudo-randomly according to the encryption key. Thus, the final encrypted image $\mathbf{I}_e$ is generated.

2.2 Data embedding

During this stage, the data hider embeds the secret data into the encrypted image by two steps.

The first step: the encrypted image $\mathbf{I}_e$ is divided into a series of non-overlapping 2×2-sized blocks $\mathbf{E}_{i,j}$ ($1 \le i \le M/2$ and $1 \le j \le N/2$), which is the same with method in encryption phase. In each block $\mathbf{E}_{i,j}$, four pixels are denoted by $E^{(0,0)}_{i,j}$, $E^{(0,1)}_{i,j}$, $E^{(1,0)}_{i,j}$, $E^{(1,1)}_{i,j}$ from left to right, then from top to bottom. Then, the data hider calculates the absolute difference of the two circle pixels in each block, as illustrated in Fig. 2. For each block, the function is used to measure its smoothness:

$$d_{i,j} = \left| 2^p \cdot \left\lfloor E^{(0,0)}_{i,j} \Big/ 2^p \right\rfloor - 2^p \cdot \left\lfloor E^{(1,1)}_{i,j} \Big/ 2^p \right\rfloor \right|, \tag{8}$$

where $d_{i,j}$ represents the value of smoothness of corresponding block $\mathbf{E}_{i,j}$. Higher $d_{i,j}$ denotes the corresponding block is more complex. For classifying the smooth and complex regions, a threshold T is obtained. According to the threshold T, the data hider divides the blocks into smooth region and complex region:

$$\begin{cases} \mathbf{E}_{i,j} \in \mathbf{R}_1, & \text{if } d_{i,j} \le T, \\ \mathbf{E}_{i,j} \in \mathbf{R}_2, & \text{if } d_{i,j} > T, \end{cases} \tag{9}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ represent smooth region and complex region, respectively. In each encrypted block of smooth region $\mathbf{R}_1$, the data hider collect the square pixels $E^{(0,1)}_{i,j}$ and $E^{(1,0)}_{i,j}$ to generate a group $\mathbf{G}_e$. Then, the most significant bit (MSB) of each pixel in $\mathbf{G}_e$ is replaced by secret data b_e:

$$E'^{(x,y)}_{i,j} = \left(E^{(x,y)}_{i,j} \mod 128 \right) + b_e \times 128, \quad (x, y) \in \mathbf{\Psi}_4, \tag{10}$$

$$\mathbf{\Psi}_4 = \{(x, y) | (x, y) \in \{(0, 1), (1,0)\}\}. \tag{11}$$

When the pixels in group $\mathbf{G}_e$ are all embedded, the data hider puts them into their original position. We embed the secret data into the smooth region due to the full exploitation of spatial correlation in natural image. Thus, the first step is completed. Since the blocks belonging to smooth region are used to hide the additional data by MSB replacement in this step, thus, if the threshold T is higher, more blocks would belong to smooth region and more additional bits can be embedded into the encrypted image.

The second step: the data hider collects the p least significant bits of two circle pixels ($E^{(0,0)}_{i,j}$, $E^{(1,1)}_{i,j}$) in each block $\mathbf{E}_{i,j}$ and divides them into a number of groups, each of which has q pixels. The number of pixels which are used for embedding in second step is $M \times N/2$. In each group, denote them as $v(k, 1)$, $v(k, 2)$, …, $v(k, p{\cdot}q)$. Here, k denotes the group index. A matrix $\mathbf{M}$ sized $(p{\cdot}q - s) \times p{\cdot}q$ is obtained by the data hider according to the data hiding key, which consist of two parts:

$$\mathbf{M} = [\mathbf{I}_{p \cdot q - s}, \mathbf{Q}], \tag{12}$$

where the matrix $\mathbf{I}$ is an $(p{\cdot}q - s) \times (p{\cdot}q - s)$ identity matrix, and the matrix $\mathbf{Q}$ sized $(p{\cdot}q - s) \times s$ is a pseudo-random binary matrix. In each group, the data hider compresses the bits by Eq. (13):

$$\begin{bmatrix} v'(k,\ 1) \\ v'(k,\ 2) \\ \vdots \\ v'(k,\ pq - s) \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} v(k,\ 1) \\ v(k,\ 2) \\ \vdots \\ v(k,\ pq) \end{bmatrix}. \tag{13}$$

The arithmetic in Eq. (13) is modulo-2. After calculated by function (13), $[v(k, 1), v(k, 2), \ldots, v(k, p{\cdot}q)]$ becomes $[v'(k, 1), v'(k, 2), \ldots, v'(k, p{\cdot}q - s)]$, that is, $p{\cdot}q$ bits are compressed into $(p{\cdot}q - s)$ bits. Thus, the data hider embeds the data into $[v'(k, p{\cdot}q - s + 1), v'(k, p{\cdot}q - s + 2), \ldots, v'(k, p{\cdot}q)]$ in each group. Combine $[v'(k, 1), v'(k, 2), \ldots, v'(k, p{\cdot}q - s)]$ and $[v'(k, p{\cdot}q - s + 1), v'(k, p{\cdot}q - s + 2), \ldots, v'(k, p{\cdot}q)]$ to generate a new $[v'(k, 1), v'(k, 2), \ldots, v'(k, p{\cdot}q)]$. Substitute the $[v(k, 1), v(k, 2), \ldots, v(k, p{\cdot}q)]$ with $[v'(k, 1), v'(k, 2), \ldots, v'(k, p{\cdot}q)]$ and put them into their original position. After the two steps, an encrypted image with embedded data $\mathbf{I}_m$ is generated. However, the order of the two steps can be changed and known by the receiver.

2.3 Data extraction and image recovery

During this phase, three cases may happened: (1) receiver only has the encryption key, (2) receiver only has the data hiding key, (3) receiver has both encryption key and data hiding key. In our proposed scheme, the threshold T is a fixed number and known by the content owner, data hider and receiver.

If the receiver only has the encryption key, he/she can decrypt the marked image to obtain an image with high

image quality which is similar to original image. First, the receiver divides the marked image $\mathbf{I}_{\mathrm{m}}$ into $M\times N/4$ non-overlapping blocks sized 2×2 and classifies them into smooth region $\mathbf{R}_1$ and complex region $\mathbf{R}_2$ by the threshold T. Then, the pixels in each block are decomposed into 8 bits. Decrypting the four pixels in each block using exclusive OR operation according to the corresponding encryption key, as described in Sect. 2.1. After that, all the $M\times N/4$ blocks are inversely permuted back to their original locations in the image by the encryption key. Thus, a basic image $\mathbf{I}'_{\mathrm{m}}$ is obtained. To improve the directly decrypted image's quality, the receiver needs to further recover the MSB of square pixels of each block in smooth region $\mathbf{R}_1$. Because $(8-p)$ MSB layers of the circle pixels in $\mathbf{I}'_m$ are all unchanged, the receiver can generate a function to obtain estimated gray value:

$$\hat{p}_{m,n}=\frac{\lfloor p_{m-1,n}/2^p\rfloor+\lfloor p_{m+1,n}/2^p\rfloor+\lfloor p_{m,n-1}/2^p\rfloor+\lfloor p_{m,n+1}/2^p\rfloor}{4}\cdot 2^p+2^{p-1}, \tag{14}$$

where the estimated gray value is generated from the neighbors in the basic image $\mathbf{I}'_m$, and (m, n) denotes the position of the pixels ($1\le m\le M$ and $1\le n\le N$). For the square pixels in smooth region $\mathbf{R}_1$, the receiver calculates:

$$p'_{m,n}=\begin{cases}128+\mathrm{mod}(p_{m,n},128), & \text{if } \left|128+\mathrm{mod}(p_{m,n},128)-\hat{p}_{m,n}\right|<\left|\mathrm{mod}(p_{m,n},128)-\hat{p}_{m,n}\right|,\\ \mathrm{mod}(p_{m,n},128), & \text{otherwise.}\end{cases} \tag{15}$$

According to the function (15), the receiver further recovers the MSB of the square pixels in smooth region. Because the data hider embeds a part of data into MSB layer in the smooth region, the result of further recovery is good. Thus, final decrypted image $\mathbf{I}_{\mathrm{d}}$ is generated.

If the receiver only has the data hiding key, he can extract the embedded data from the marked image directly. First, the receiver divides the marked image into $M\times N/4$ non-overlapping blocks sized 2×2 and classifies them into smooth region $\mathbf{R}_1$ and complex region $\mathbf{R}_2$ by the threshold T. After that the embedded data in MSB of the square pixels of the smooth region $\mathbf{R}_1$ can be extracted by the data hiding key. Then, the receiver collects p bits of circle pixels in each block and divides them into k groups. In each group, s embedded bits can be extracted according to the data hiding key. Thus, the receiver extracts all the embedded data without known the original image.

If the receiver has both the encryption key and data hiding key, he can extract the embedded data and recover the original image. In the beginning, the receiver divides the marked image into $M\times N/4$ non-overlapping blocks, and the values of parameters p, q and s are obtained according to the data hiding key. $M\times N/4$ non-overlapping blocks are classified into smooth region and complex region by the threshold T. Thus, the embedded data can be extracted from the marked image. Then the receiver decrypts the marked image to generate decrypted image $\mathbf{I}_{\mathrm{d}}$ according to the encryption key. In decrypted image $\mathbf{I}_{\mathrm{d}}$, p LSB layers of the circle pixels in each block need to recover. In each pixel group, the receiver can get the vector $[v'(k, 1), v'(k, 2), \ldots, v'(k, p\cdot q-s)]$ in (13). The original bits vector of each group $[v(k, 1), v(k, 2), \ldots, v(k, p\cdot q)]$ must be one in function (16):

$$\mathbf{\Theta}=\left[v'(k,\ 1), v'(k,\ 2), \ldots, v'(k,\ pq-s), 0, 0, \ldots, 0\right]+\mathbf{\Gamma}\cdot\mathbf{H}, \tag{16}$$

$$\mathbf{H}=[\mathbf{Q}', \mathbf{I}_s], \tag{17}$$

where $\mathbf{\Gamma}$ is an arbitrary binary vector sized $1\times s$, and $\mathbf{H}$ is a matrix sized $s\times pq$ and consisted of the transpose of $\mathbf{Q}$ and an $s\times s$ identity matrix. Thus, for recovering the vector $[v(k, 1), v(k, 2), \ldots, v(k, p\cdot q)]$, there are 2^s possibilities. The receiver puts the elements in each vector $\mathbf{\Theta}$ into their original position and decrypts the pixel group to generate decrypted pixel group O_k according to the encryption key, and then calculates the total difference of the decrypted and estimated pixels in each group. Here, $\tilde{p}_{m,n}$ is generated from the neighbors in the directly decrypted image $\mathbf{I}_{\mathrm{d}}$, and $r_{m,n}$ denotes the pixel values in decrypted pixel group O_k:

$$\tilde{p}_{m,n}=\frac{p_{m-1,n}+p_{m+1,n}+p_{m,n-1}+p_{m,n+1}}{4}, \tag{18}$$

$$D=\sum_{(m,\,n)\in O_k}\left|r_{m,n}-\tilde{p}_{m,n}\right|. \tag{19}$$

Thus, 2^s possible D corresponding to 2^s decrypted pixel group O_k. 2^s decrypted pixel groups must contain the original segment which has the smallest D, because of the spatial correlation in natural image. That is if the receiver find the smallest D, he can regard the corresponding pixel group as the original segment. When the pixel groups are all recovered, the original image is recovered.

In our scheme, the adjacent pixels are helped each other to recover the image. We first use $(8-p)$ MSB layers of circle pixels to recover the MSB layer of square pixels in decryption phase. Then, the recovered square pixels are used to further recover the p LSB layers of circle pixels. Finally,

the original image is recovered without any error. This way can improve the embedding rate and the visual quality of decrypted image significantly. The parameters of *p, q* and *s* can be transmitted as the additional data.

3 Experimental analysis and results

3.1 Security analysis

For the schemes of reversible data hiding in encrypted image, the security of encryption is important. Our encryption method consists of stream encryption and permutation. After the stream encryption, we randomly permute the $M \times N/4$ non-overlapping blocks. In the permutation phase, there are at most $(M \times N/4)!$ possibilities of the permuted patterns. However, the value of $(M \times N/4)!$ is too large to rearrange the permuted blocks as original. To illustrate, we test the image *lake* in this phase. Figure 3a–d shows the original image, encrypted image only with stream encryption, final encrypted image and decrypted image, respectively. In Fig. 4a, b, we show the histograms of original image and final encrypted image; however, the two histograms are quite different. In Fig. 4c, d, the x-axis and y-axis denote the position of corresponding pixel of the image, and the z-axis represents the pixel values. In addition, distributions of the pixel values of the original image and final encrypted image are shown in Fig. 4c, d. Clearly, the distribution of pixel values in the final encrypted image is uniform. Generally, the security of our encryption method is guaranteed.

3.2 Comparisons with state-of-the-art schemes

In this phase, the sizes of test images are 512×512, and the threshold T is 1. In Fig. 5, we use image *Airplane* to show the results, here, the parameters of *q, p* and *s* are 1600, 1 and 1, respectively. Figure 5a shows the original image *Airplane*. The encrypted image and marked image with embedding rate 0.1171 bpp are shown in Fig. 5b, c. Figure 5d illustrates the directly decrypted image with PSNR of 57.4 dB.

The results of four images *Airplane, Lake, Lena* and *Crowd* with different embedding rates are shown in Fig. 6.

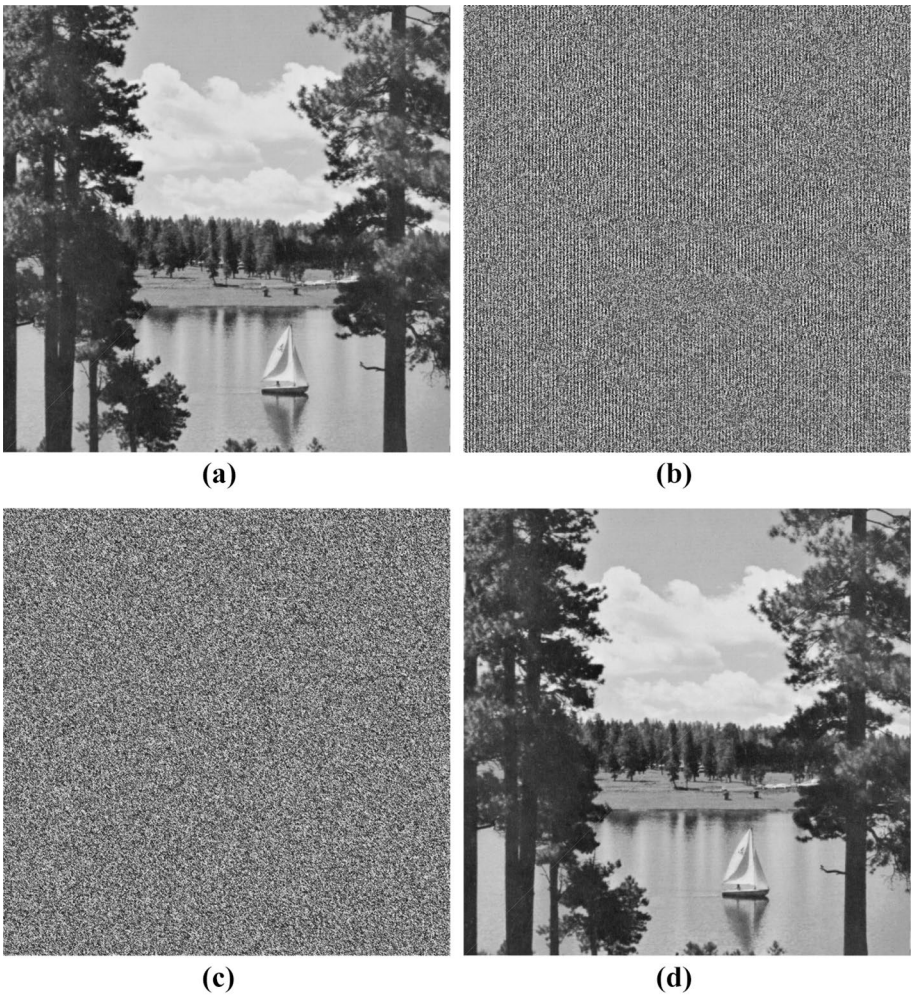


Fig. 3 **a** Original image, **b** encrypted result with stream encryption, **c** final encrypted result with block permutation after stream encryption, **d** decrypted image

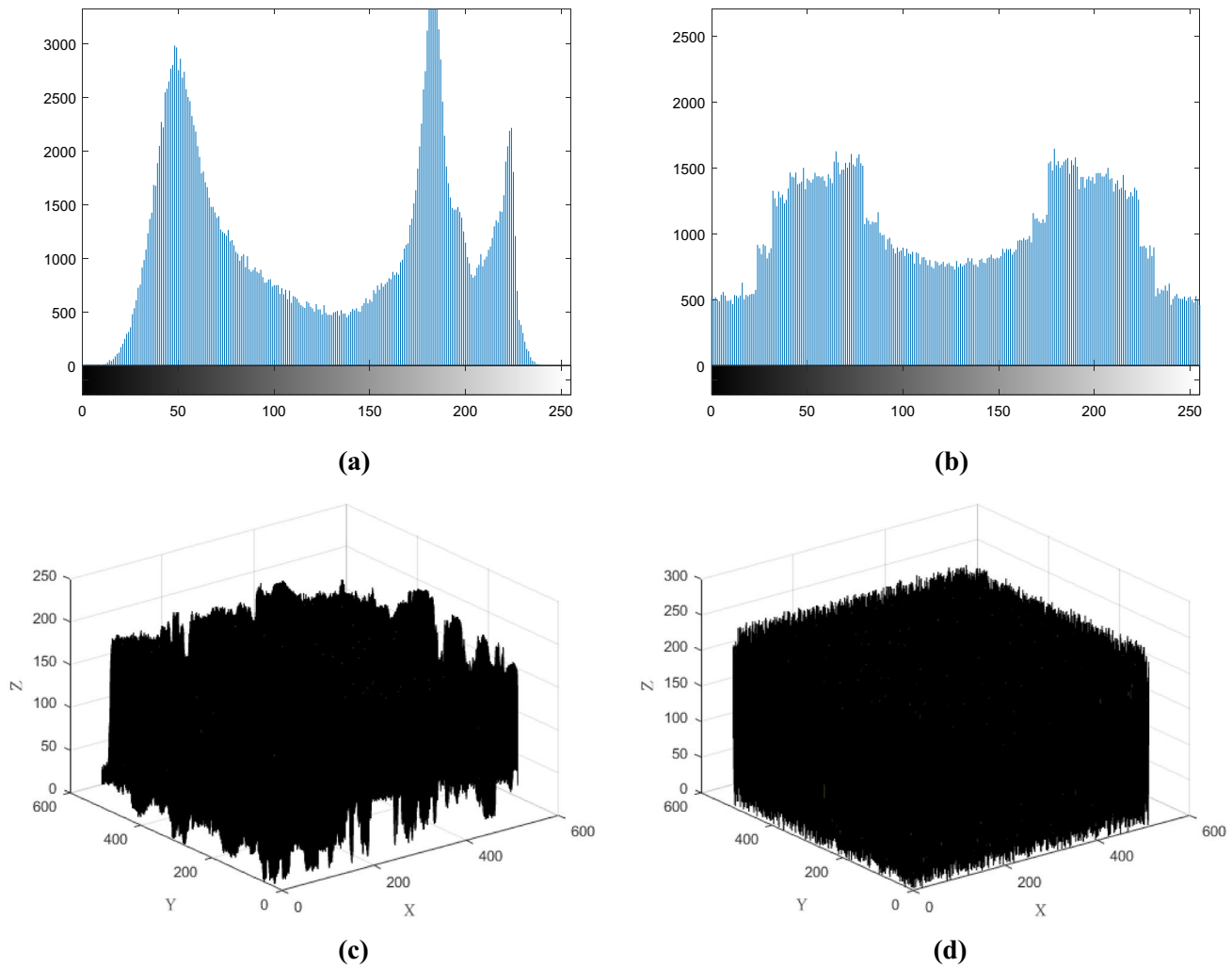


Fig. 4 Histograms and distributions of original image *Lake* and its final encrypted version. **a** Histogram of original image, **b** histogram of encrypted image, **c** distribution of the pixel values of the original image, **d** distribution of the pixel values of the final encrypted image

They all have good image quality with big embedding capacity. Our proposed scheme is more suitable for the smooth image because the smooth region and complex region in the image are used.

Table 1 illustrates the results of the four images with different parameters *q, p* and *s* in detail. From Table 1, we can see that when the parameter *p* is high, the corresponding embedding rate is also improved. However, due to more LSB layers are changed, the corresponding PSNR in directly decrypted images are declined a little. In addition, as illustrated in Table 1, the image quality of final recovered image is good, and mostly can be recovered without any error.

Figures 7 and 8 show the comparisons between the propose scheme and [17, 19, 21]. Here, the parameter *p* of method in [19] is 1. Figure 7 illustrates the PSNR of directly decrypted image with different embedding rate comparisons. Both in PSNR and embedding rate, our propose scheme is well beyond other three methods. Figure 8 illustrates the SSIM of directly decrypted image with different embedding rate comparisons [28]. The results of SSIM and embedding rate of our proposed scheme are also very good. In our experiments, we set the threshold *T* is 1. If higher embedding capacity is required, the user can set higher value of the threshold *T*.

3.3 Analysis of computation complexity

The computation complexity of our proposed scheme mainly depends on the embedding phase including two embedding ways and the iteration algorithm in image recovery phase. As a result, the real-time performance will be affected, if more bits are embedded into the encrypted image or more groups are obtained. In the proposed scheme, the encryption method, embedding method and decryption method are all have low computation complexity. Here, we set parameters of *q, p* and *s* are 1600, 1 and 1, respectively. The image *Lena*

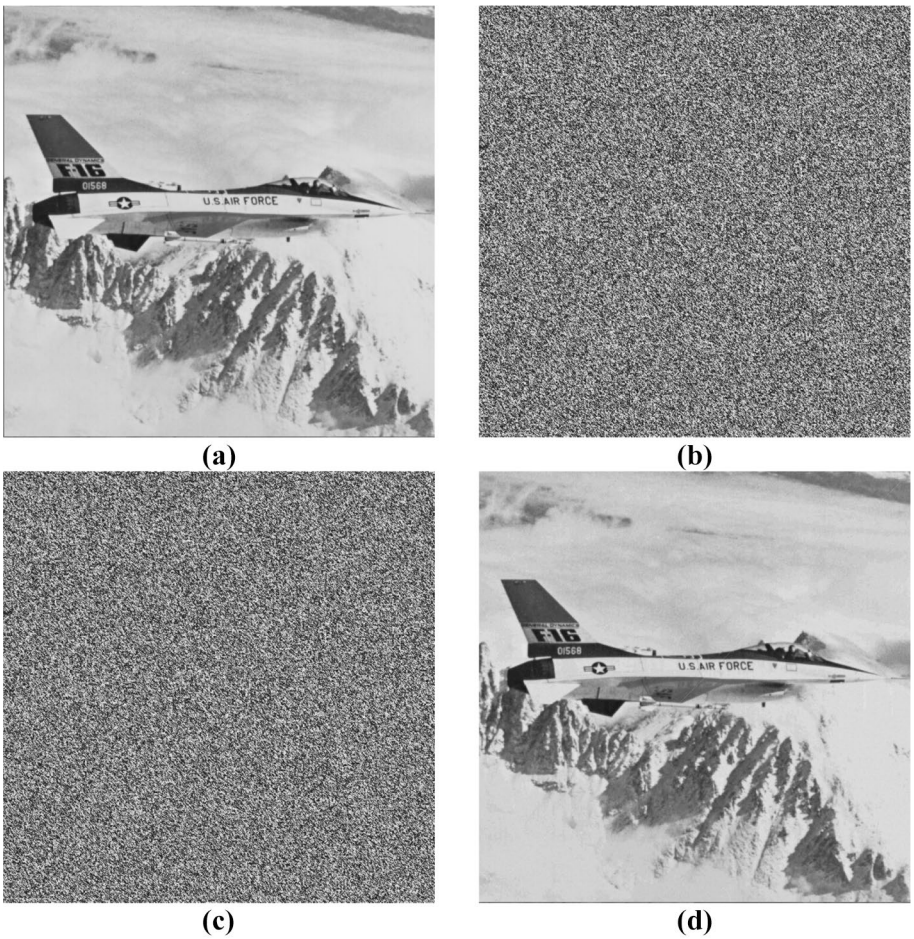


Fig. 5 **a** Original image, **b** encrypted image, **c** marked image (embedding rate 0.1171 bpp), **d** directly decrypted image with PSNR of 57.4 dB

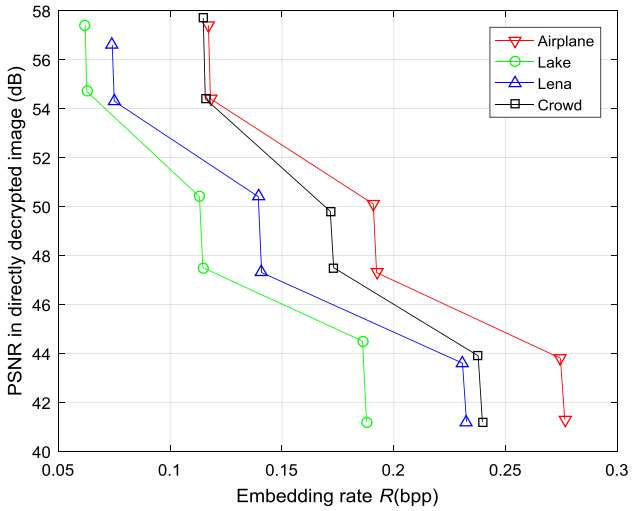


Fig. 6 PSNR of directly decrypted image with different embedding rates

sized 512 by 512 was used as an example. The execution time of image encryption, data embedding, direct decryption and image recovery are 0.819, 2.570, 0.824 and 1.863 s, respectively, which demonstrates the real-time performance of our scheme and the feasibility in real applications.

4 Conclusions

In this work, a real-time reversible data hiding scheme in encrypted image is presented, which uses mutual aid of the adjacent pixels to improve the embedding capacity and image quality of decrypted image. In the encryption phase, the content owner divides the original image into blocks and uses the stream cipher and permutation to encrypt the blocks. When the data hider obtains the encrypted image, he can embed the secret data into the encrypted image without known the image content. In detail, the data hider first classifies the encrypted blocks into smooth region and complex region and embeds the secret data into the MSB layer of square pixels in blocks of smooth region. Then, he collects the LSB layers of the circle pixels in all blocks and compresses them to make a room for embedding the secret data again. Finally, the marked image with secret data is generated. If he receiver only has the encryption key, he can divides the marked image into blocks and decrypts

Table 1 Embedding rate (R), PSNR of directly decrypted images and PSNR of recovered images under different in parameters q, p and s

	q	p	$s=1$	$s=2$	$s=3$	$s=4$
Airplane	1600	1	0.1171, 57.4, $+\infty$	0.1174, 55.3, $+\infty$	0.1177, 54.8, $+\infty$	0.1180, 54.4, $+\infty$
	1000	2	0.1908, 50.1, $+\infty$	0.1913, 48.3, $+\infty$	0.1918, 47.6, $+\infty$	0.1923, 47.3, $+\infty$
	800	3	0.2745, 43.8, $+\infty$	0.2751, 41.9, $+\infty$	0.2757, 41.7, $+\infty$	0.2764, 41.3, $+\infty$
Lake	2000	1	0.0620, 57.4, $+\infty$	0.0622, 55.0, $+\infty$	0.0624, 54.5, 66.2	0.0627, 54.5, 63.9
	1000	2	0.1131, 50.4, $+\infty$	0.1136, 48.6, $+\infty$	0.1141, 47.7, $+\infty$	0.1146, 47.5, 68.6
	800	3	0.1860, 44.5, $+\infty$	0.1866, 42.5, $+\infty$	0.1872, 41.4, $+\infty$	0.1878, 41.2, $+\infty$
Lena	1600	1	0.0741, 56.6, $+\infty$	0.0744, 55.4, $+\infty$	0.0747, 54.8, $+\infty$	0.0750, 54.3, $+\infty$
	1000	2	0.1394, 50.4, $+\infty$	0.1399, 48.6, $+\infty$	0.1404, 47.8, $+\infty$	0.1409, 47.3, $+\infty$
	800	3	0.2307, 43.6, $+\infty$	0.2313, 42.2, $+\infty$	0.2319, 41.5, $+\infty$	0.2325, 41.2, $+\infty$
Crowd	1600	1	0.1148, 57.7, $+\infty$	0.1151, 55.5, $+\infty$	0.1154, 54.7, $+\infty$	0.1157, 54.4, $+\infty$
	1000	2	0.1716, 49.8, $+\infty$	0.1721, 48.4, $+\infty$	0.1726, 47.8, $+\infty$	0.1731, 47.5, $+\infty$
	800	3	0.2377, 43.9, $+\infty$	0.2383, 41.9, $+\infty$	0.2389, 41.5, $+\infty$	0.2396, 41.2, $+\infty$

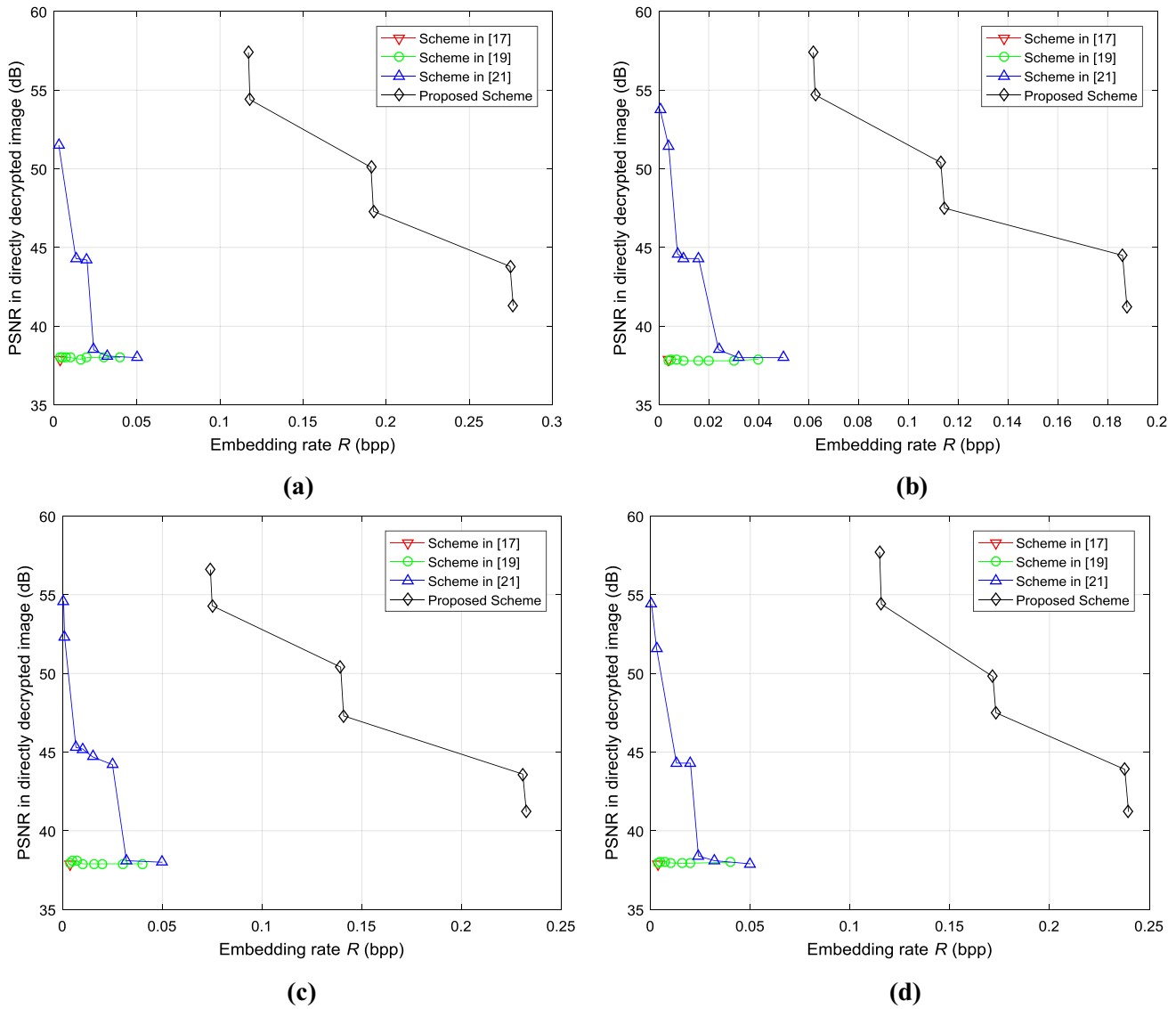


Fig. 7 Rate-PSNR comparisons between the proposed scheme and [17, 19, 21]. **a** *Airplane*, **b** *Lake*, **c** *Lena*, **d** *Crowd*

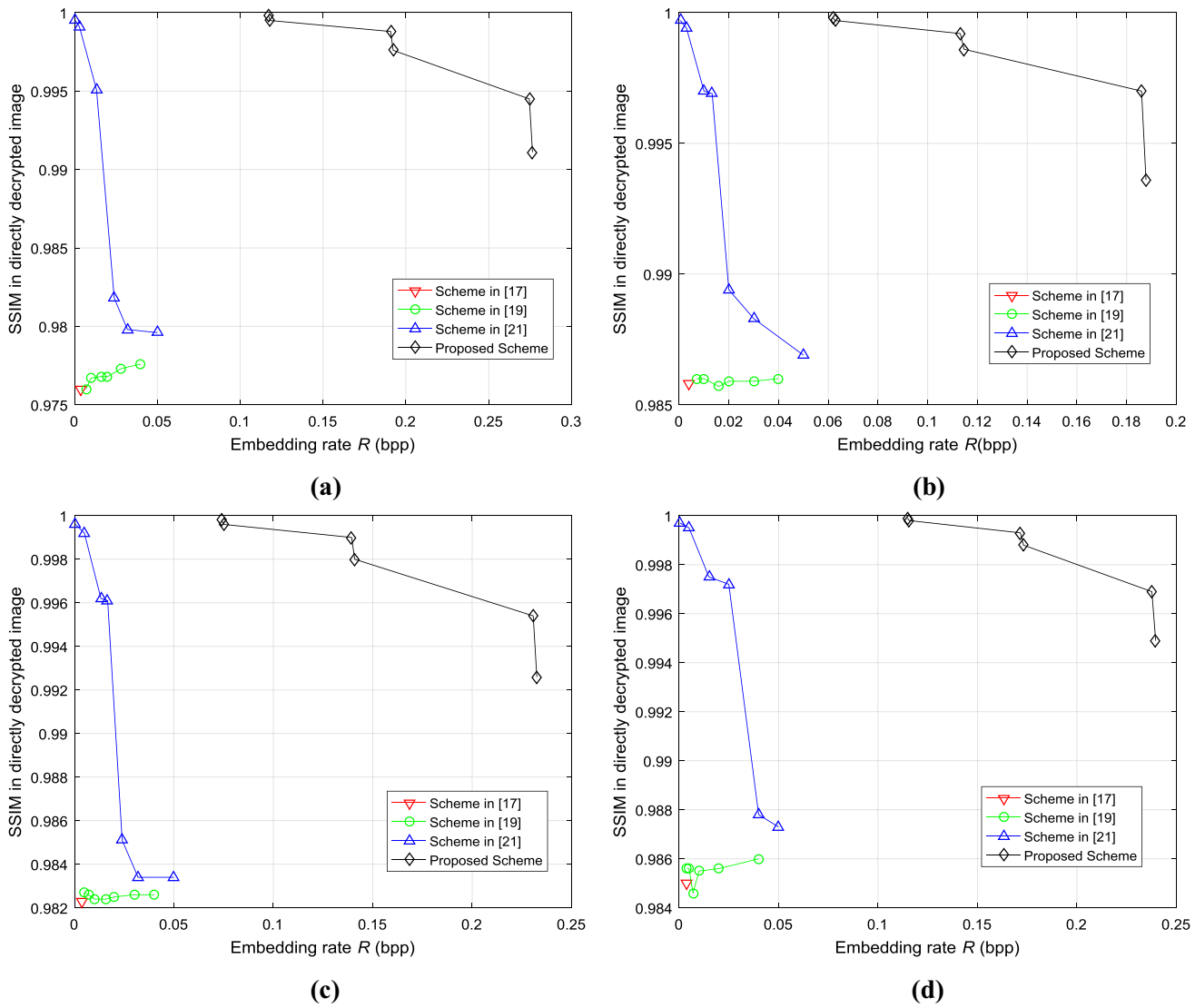


Fig. 8 Rate-SSIM comparisons between the proposed scheme and [17, 19, 21]. **a** *Airplane*, **b** *Lake*, **c** *Lena*, **d** *Crowd*

them directly by the encryption key. If the receiver only has the data hiding key, he can classifies encrypted blocks into smooth region and complex region according to the threshold T, and then the embedded data can be extracted by the data hiding key. If the receiver has both encryption key and data hiding key, he can extract the embedded data directly and recover the original image perfectly. Our scheme has good quality of decrypted image and high embedding rate due to the MSB replacement in smooth region. Experimental results show the better rate-distortion performance of our scheme than some of state-of-the-art schemes and also the high computational efficiency for the requirements of real-time applications.

Acknowledgements This work was supported by the National Natural Science Foundation of China (61672354, 61702332), Shanghai Engineering Center Project of Massive Internet of Things Technology for Smart Home (GCZX14014), and Hujiang Foundation of China (C14001, C14002). The authors would like to thank the anonymous reviewers for their valuable comments.

References

1. Zhang, Y., Qin, C., Zhang, W.M., Liu, F.L., Luo, X.Y.: On the fault-tolerant performance for a class of robust image steganography. Signal Process. **146**, 99–111 (2018)
2. Qin, C., Chang, C.C., Chiu, Y.P.: A novel joint data-hiding and compression scheme based on SMVQ and image inpainting. IEEE Trans. Image Process. **23**(3), 969–978 (2014)
3. Ma, Y.Y., Luo, X.Y., Li, X.L., Bao, Z.K., Zhang, Y.: Selection of rich model steganalysis features based on decision rough set α-positive region reduction. IEEE Trans. Circuits Syst. Video Technol. (2018). https://doi.org/10.1109/TCSVT.2018.2799243

4. Luo, X.Y., Song, X.F., Li, X.L., Zhang, W.M., Lu, J.C., Yang, C.F., Liu, F.L.: Steganalysis of HUGO steganography based on parameter recognition of syndrome-trellis-codes. Multimed. Tools Appl. **75**(21), 13557–13583 (2016)
5. Qin, C., Ji, P., Zhang, X.P., Dong, J., Wang, J.W.: Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. Signal Process. **138**, 280–293 (2017)
6. Qin, C., Ji, P., Chang, C.-C., Dong, J., Sun, X.M.: Non-uniform watermark sharing based on optimal iterative BTC for image tampering recovery. IEEE Multimed. (2018). https://doi.org/10.1109/MMUL.2018.112142509
7. Qin, C., Chen, X.Q., Ye, D.P., Wang, J.W., Sun, X.M.: A novel image hashing scheme with perceptual robustness using block truncation coding. Inf. Sci. **361–362**, 84–99 (2016)
8. Qin, C., Chen, X.Q., Luo, X.Y., Zhang, X.P., Sun, X.M.: Perceptual image hashing via dual-cross pattern encoding and salient structure detection. Inf. Sci. **423**, 284–302 (2018)
9. Yao, H., Qin, C., Tang, Z.J., Tian, Y.: Improved dual-image reversible data hiding method using the selection strategy of shiftable pixels' coordinates with minimum distortion. Signal Process. **135**, 26–35 (2017)
10. Yao, H., Qin, C., Tang, Z.J., Tian, Y.: Guided filtering based color image reversible data hiding. J. Vis. Commun. Image Represent. **43**, 152–163 (2017)
11. Li, X.L., Zhang, W.M., Gui, X.L., Yang, B.: Efficient reversible data hiding based on multiple histograms modification. IEEE Trans. Inf. Forensics Secur. **10**(9), 2016–2027 (2015)
12. Ou, B., Li, X.L., Zhao, Y., Ni, R.R., Shi, Y.Q.: Pairwise prediction-error expansion for efficient reversible data hiding. IEEE Trans. Image Process. **22**(12), 5010–5021 (2013)
13. Li, X.L., Li, B., Yang, B., Zeng, T.Y.: General framework to histogram-shifting-based reversible data hiding. IEEE Trans. Image Process. **22**(6), 2181–2191 (2013)
14. Qin, C., Chang, C.C., Huang, Y.H., Liao, L.T.: An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism. IEEE Trans. Circuits Syst. Video Technol. **23**(7), 1109–1118 (2013)
15. Qin, C., Chang, C.C., Hsu, T.J.: Reversible data hiding scheme based on exploiting modification direction with two steganographic images. Multimed. Tools Appl. **74**(15), 5861–5872 (2015)
16. Hu, R., Li, X., Yang, B.: A new lossy compression scheme for encrypted gray-scale images. In: IEEE International Conference on Acoustic, Speech and Signal Processing, pp. 7387–7390 (2014)
17. Zhang, X.: Reversible data hiding in encrypted image. IEEE Signal Process. Lett. **18**(4), 255–258 (2011)
18. Hong, W., Chen, T., Wu, H.: An improved reversible data hiding in encrypted images using side match. IEEE Signal Process. Lett. **19**(4), 199–202 (2012)
19. Liao, X., Shu, C.: Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. J. Vis. Commun. Image Represent. **28**, 21–27 (2015)
20. Liu, W., Zeng, W., Dong, L., Yao, Q.: Efficient compression of encrypted grayscale images. IEEE Trans. Image Process. **19**(4), 1097–1102 (2010)
21. Zhang, X.: Separable reversible data hiding in encrypted image. IEEE Trans. Inf. Forensics Secur. **7**(2), 826–832 (2012)
22. Qian, Z., Zhang, X., Feng, G.: Reversible data hiding in encrypted images based on progressive recovery. IEEE Signal Process. Lett. **23**(11), 1672–1676 (2016)
23. Bianchi, T., Piva, A., Barni, M.: On the implementation of the discrete Fourier transform in the encrypted domain. IEEE Trans. Inf. Forensics Secur. **4**(1), 86–97 (2009)
24. Puteaux, P., Trinel, D., Puech, W.: High-capacity data hiding in encrypted images using MSB prediction. In: 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA), Oulu, Finland, pp. 1–6 (2016)
25. Zhang, X., Qian, Z., Feng, G., Ren, Y.: Efficient reversible data hiding in encrypted images. J. Vis. Commun. Image Represent. **25**(2), 322–328 (2014)
26. Qian, Z., Zhang, X.: Reversible data hiding in encrypted image by distributed encoding. IEEE Trans. Circuits Syst. Video Technol. **26**(4), 636–646 (2016)
27. Qin, C., Zhang, X.P.: Effective reversible data hiding in encrypted image with privacy protection for image content. J. Vis. Commun. Image Represent. **31**, 154–164 (2015)
28. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)

Wei Zhang received B.S. degree in automation from Nanhang Jincheng College, Nanjing, Jiangsu, China, in 2016. He is currently pursuing the M.S. degree in instrument and meter engineering from University of Shanghai for Science and Technology, China. His research interests include data hiding in encrypted image, image watermarking and image authentication.

Ping Kong received the B.S. degree in mathematics and applied mathematics from Qufu Normal University, Shandong, China, in 2001, the M.S. degree in system theory from Kunming University of Science and Technology, Yunnan, China, in 2004, and the Ph.D. degree in management science and engineering from University of Shanghai for Science and Technology, Shanghai, China, in 2011. Since 2011, she has been with the faculty of Shanghai Key Laboratory for Molecular Imaging, Shanghai University of Medicine and Health Sciences, where she is currently an Associate Professor. Her research interest is medical image processing.

Heng Yao received the B. Sc. degree from Hefei University of Technology, China, in 2004, the M. Eng. degree from Shanghai Normal University, China, in 2008, and the Ph.D. degree in signal and information processing from Shanghai University, China, in 2012. Currently, he is with School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. His

research interests include multimedia security, image processing, and pattern recognition.

Yu-Chen Hu received his Ph.D. degree in computer science and information engineering from the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan in 1999. Currently, Dr. Hu is a professor in the Department of Computer Science and Information Management, Providence University, Sha-Lu, Taiwan. He is a member of ACM and IEEE. He is also a member of Computer Vision, Graphics, and Image Processing (CVGIP), Chinese Cryptology and Information Security Association. Dr Hu Servers as the Editor-in-Chief of International Journal of Image Processing since 2009. In addition, he is the managing editor of Journal of Information Assurance and Security. His research interests include image and signal processing, data compression, information hiding, and data engineering.

Fang Cao received the B.S. degree in applied electronics from Shanghai Normal University, Shanghai, China, in 2002, the M.S. degree in signal and information processing from Shanghai Maritime University, Shanghai, China, in 2004, and the Ph.D. degree in communication and information system from Shanghai University, Shanghai, China, in 2013. Since 2005, she has been with the faculty of the College of Information Engineering, Shanghai Maritime University, where she is currently a Lecturer. Her research interests include image processing, computer vision and multimedia security.