Separable Reversible Data Hiding in Encrypted Images via Adaptive Embedding Strategy with Block Selection

Chuan Qin , Wei Zhang , Fang Cao , Xinpeng Zhang , Chin-Chen Chang

PII: S0165-1684(18)30230-5
DOI: 10.1016/j.sigpro.2018.07.008
Reference: SIGPRO 6872

To appear in: *Signal Processing*

Received date: 11 March 2018
Revised date: 8 June 2018
Accepted date: 6 July 2018

Please cite this article as: Chuan Qin , Wei Zhang , Fang Cao , Xinpeng Zhang , Chin-Chen Chang , Separable Reversible Data Hiding in Encrypted Images via Adaptive Embedding Strategy with Block Selection, *Signal Processing* (2018), doi: 10.1016/j.sigpro.2018.07.008

Highlights

- An adaptive, separable reversible data hiding scheme in encrypted image is proposed.
- Analogues stream-cipher and block permutation are used to encrypt original image.
- Classification and selection for encrypted blocks are conducted during embedding.
- An accurate prediction strategy was employed to achieve perfect image recovery.
- Our scheme has better rate-distortion performance than some state-of-the-art schemes.

Separable Reversible Data Hiding in Encrypted Images via Adaptive Embedding Strategy with Block Selection

Chuan Qin[1], Wei Zhang[1], Fang Cao[2], Xinpeng Zhang[3], and Chin-Chen Chang[4]

[1]School of Optical-Electrical and Computer Engineering,
University of Shanghai for Science and Technology, Shanghai 200093, China
E-mail: qin@usst.edu.cn, wellzhang@yeah.net

[2]College of Information Engineering,
Shanghai Maritime University, Shanghai 200135, China
E-mail: fangcao@shmtu.edu.cn

[3]School of Computer Science,
Fudan University, Shanghai 200433, China
E-mail: zhangxinpeng@fudan.edu.cn

[4]Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan
E-mail: alan3c@gmail.com

Correspondence Address:

Prof. Chin-Chen Chang
Department of Information Engineering and Computer Science,
Feng Chia University,
100 Wenhwa Road, Taichung 40724, Taiwan.
E-mail: alan3c@gmail.com
TEL: 886-4-24517250 ext. 3790
FAX: 886-4-27066495

Separable Reversible Data Hiding in Encrypted Images via Adaptive Embedding Strategy with Block Selection

Chuan Qin, Wei Zhang, Fang Cao, Xinpeng Zhang, and Chin-Chen Chang

Abstract: In this paper, an adaptive reversible data hiding scheme for encrypted images is proposed. Content owner uses an analogues stream-cipher and block permutation to encrypt non-overlapping blocks of original image through encryption key. Then, data hider classifies encrypted blocks into two sets corresponding to smooth and complex regions in original image. With data-hiding key, spare space is vacated to accommodate additional bits by compressing LSBs of the block set corresponding to smooth region. Separable operations of data extraction, direct decryption and image recovery are conducted by receiver according to the availability of encryption key and data-hiding key. Through an accurate prediction strategy, perfect image recovery is achieved. Since only a portion of blocks are modified during embedding, the directly-decrypted image quality is satisfactory. Also, more bits can be embedded into the blocks belonging to smooth set, hence, embedding rate is acceptable. Experimental results demonstrate the effectiveness of our scheme.
Keywords: data hiding, reversibility, encrypted image, image decryption, image recovery

1. Introduction

Information hiding technique, also called as data hiding, has been widely studied in both academia and industry in recent years, which can embed additional data into cover data, including text, audio, image, and video, in an imperceptible way [1, 2]. There are two main research directions for data hiding: 1) achieving various protecting functionalities (copyright identification, tampering recovery, retrieval and etc) for cover data through embedding the data, i.e., watermark, in different manners; 2) realizing covert communication (steganography) for large hiding capacity of secret additional data while maintaining acceptable fidelity of cover data with well-designed encoding strategies [3-5].

The data embedding process inevitably introduces distortions on the cover image, therefore, many investigations have been carried out to study the problem of complete recovery for cover image after embedded data are extracted, which is known as reversible data hiding (RDH) [6, 7]. The embedding strategies of reported schemes for RDH can be categorized into three main types: lossless compression [8], difference expansion (DE) [9-11] and histogram shifting (HS) [12-15]. With the aim of enhancing embedding rate and stego-image quality, many studies have investigated introducing the prediction strategy into RDH [11, 13-15]. Rather than directly

applying the original image as the cover data, the relative data of the original image, i.e., prediction error (PE), was constructed as the cover data for hiding.

Due to current success of cloud storing and computing, a vast number of personal data, such as images and videos, can be stored and processed on the Internet to alleviate computing burdens on user clients. However, in order to preserve privacy, it is better to encrypt user data before uploading it onto Internet. Thus, for the convenience of data management and retrieval, how to realize RDH in encrypted images (RDHEI) attracts considerable interest in the multimedia security community [16]. Compared with RDH in plaintext image, since the entropy of encrypted image is maximized, few redundancy can be directly exploited for data embedding. Hence, conventional RDH schemes for plaintext images are not suitable to encrypted images.

Recently, many works about RDHEI have been reported [17-37]. In most reported RDHEI schemes, there are three main entities: content owner, data hider and receiver. Specifically, the content owner encrypts the original image with an encryption key and sends the encrypted image to the data hider; the data hider then inserts the additional data into the encrypted image with a data-hiding key and transmits the marked, encrypted image to the receiver; the receiver can conduct direct decryption (to obtain an image approximating to original image), data extraction, or image recovery (to recover original image reversibly) based on the availability of the encryption key and data-hiding key.

Depending on if the operations of direct decryption, data extraction and image recovery can be conducted separately on the receiver side, the RDHEI schemes can be categorized into non-separable (joint) schemes [17-22] and separable schemes [23-26]. In the scheme [17], the stream cipher was first used to encrypt all bits of original image, and then, through flipping three LSBs for half of the pixels in each block, an additional bit was embedded. After direct decryption, data extraction and image recovery can be jointly achieved based on the spatial correlation. In order to reduce the error rate of extracted bits for smaller block sizes, during the procedure of data extraction and image recovery, Hong *et al*. improved the strategy of smoothness evaluation on each block and also considered pixel correlations at the border of adjacent blocks based on the side-match between both the recovered and unrecovered blocks [18]. By selecting the partial pixels to be flipped, the scheme [20] introduced less modification to each block during data hiding, which led to a remarkable improvement in the visual quality of

directly-decrypted images. Also, an adaptive evaluation function of smoothness along the isophote direction was applied to extract data and recover the image. Zhou *et al*. adopted a public-key modulation mechanism to realize data embedding in each encrypted block, and a two-class SVM classifier was used to differentiate encrypted and non-encrypted blocks for joint decoding of embedded data and original image [22]. However, direct decryption (i.e., image decryption without data extraction) cannot be implemented in this scheme. In order to achieve the separability of RDHEI, Zhang compressed LSB layers of an encrypted image to create a spare space in which to embed secret data [23]. Thus, the receiver can achieve independent operations of correct data extraction, image decryption and image recovery individually. Qian *et al*. proposed a separable RDHEI method based on progressive recovery [24], in which the data hider segmented the encrypted image into three portions and embedded different numbers of additional bits into these three portions. On the receiver side, original image can be recovered using a progressive mechanism. In [27], Wu and Sun presented two RDHEI schemes, i.e., one non-separable scheme and one separable scheme. In their non-separable scheme, some encrypted pixels were pseudo-randomly selected for embedding, and their four neighboring pixels were guaranteed not to be modified. Each bit was embedded into one pixel group by flipping the LSBs of the pixel group. Then, with the help of estimated values based on four neighborhoods, data extraction and image recovery were jointly achieved. In their separable scheme, each additional bit was hidden into each selected encrypted pixel by MSB replacement, thus, separable data extraction and image recovery can be achieved. However, the quality of directly-decrypted image in this scheme was not ideal.

Different from the above-mentioned RDHEI schemes [17-27] that vacated room after encryption (VRAE), some studies attempted to use a pre-processing method, i.e., reserving room before encryption (RRBE) [28-31]. In Ma *et al*.'s scheme [28], before image encryption, the original image was segmented into two parts, and the LSBs of one part were hidden in the other part with one conventional plaintext-image RDH method. Then, the LSBs of the first part were vacated to embed additional data through LSB replacement in the encrypted domain. The schemes [29, 30] first predicted the non-sampled pixels based on the sampled pixels before encryption, and then the prediction errors after adjustment and encryption, which had a concentrated histogram distribution, can be used for embedding additional data. Combining the

encrypted, sampled pixels with the marked, encrypted prediction errors, the final marked, encrypted image can be produced. Cao *et al*. compressed image patches using sparse representation [31], and the residual errors were encoded and self-embedded before image encryption to create a large hiding room for encrypted image. Besides the schemes of VRAE and RRBE, some RDHEI schemes preserved some redundancy and correlation in the encrypted image during image encryption, which can be effectively utilized to embed additional data. In the scheme [32], the pixels in the same sub-block were encrypted with the same key stream byte, thus, the correlation between neighboring pixels in each sub-block can be well maintained in the encrypted domain. Then, the conventional RDH schemes for plaintext images can be directly applied to the encrypted image for data embedding through exploiting the redundancy and correlation in the encrypted sub-blocks. Hence, this scheme can be classified as the category of vacating room within encryption (VRIE).

Some other different techniques were also involved to achieve RDHEI, such as low-density parity-check (LDPC) codes [33, 34] and homomorphic encryption [35-37]. In schemes [33, 34], the content owner both utilized the stream cipher to encrypt original image, and the data hider compressed a part of encrypted data using LDPC codes based on Slepian-Wolf model. Thus, the additional data as well as compressed data can be embedded into encrypted image reversibly. Wu *et al*. divided each pixel of original image into three components by energy transfer equation, and each component was encrypted with Paillier homomorphic encryption [36]. Based on the properties of homomorphism, additional bits can be embedded by manipulating the encrypted signals. Xiao *et al*. adopted pixel value ordering (PVO) strategy to hide data in each block after the homomorphic encryption [37], and the additive homomorphism guaranteed the performance of PVO in encrypted domain was close to that in plaintext domain.

In this paper, in order to achieve better rate-distortion performance, we propose a novel RDHEI scheme with separable capability and high quality of directly-decrypted image. In our scheme, the content owner divides original image into non-overlapping blocks and encrypts all blocks by specific stream cipher and block permutation with the encryption key. Because image recovery of most RDHEI schemes is based on the characteristic of spatial correlation, therefore, in order to decrease the errors of image recovery caused by complex image distribution and keep acceptable embedding rate, the data hider in our scheme classifies all encrypted blocks into two

sets corresponding to smooth and complex regions within original image. With data-hiding key, spare space can then be vacated to accommodate additional bits by compressing LSB layers of the block set corresponding to smooth region. On the receiver side, separable operations of data extraction, direct decryption and image recovery can be carried out according to the availability of encryption key and data-hiding key. Through an accurate prediction strategy, successful image recovery can be achieved. Since only a portion of image blocks are modified during data embedding, the quality of directly-decrypted image is satisfactory. More additional bits can be embedded into the blocks belonging to smooth set, hence, embedding rate is also acceptable.

The remaining parts of the paper are arranged as follows. Section 2 describes the proposed scheme detailedly, including the procedures of image encryption, data embedding, and data extraction and image recovery. Experimental results and analysis are given to demonstrate the effectiveness and superiority of our scheme in Section 3. Section 4 concludes this paper.

2. Proposed Scheme

In this section, an effective RDHEI scheme is presented, which consists of image encryption, data embedding, data extraction and image recovery. Figure 1 illustrates the framework of the proposed scheme. The content owner divides original image into a series of non-overlapping blocks and encrypts the image according to the encryption key. Then, the data hider, such as a cloud administrator, classifies the encrypted blocks into two sets according to a pre-determined threshold, and the additional data are embedded into one set of encrypted blocks, corresponding to smooth region in the original image, through compressing the least significant bits (LSB) to create a spare space. Then, the marked, encrypted image is sent to the receiver side. If the receiver only has the data-hiding key, he/she can also distinguish the two sets in received image and extract the embedded data successfully. A directly-decrypted image with good quality approximating to original image can be obtained only with the encryption key. When both the encryption key and data-hiding key are available, the receiver can extract embedded data and recover original image perfectly through exploiting spatial correlation of natural images. Detailed procedures of our scheme are presented below, and the main symbols used for the description of our scheme and their definitions are listed in Table 1.

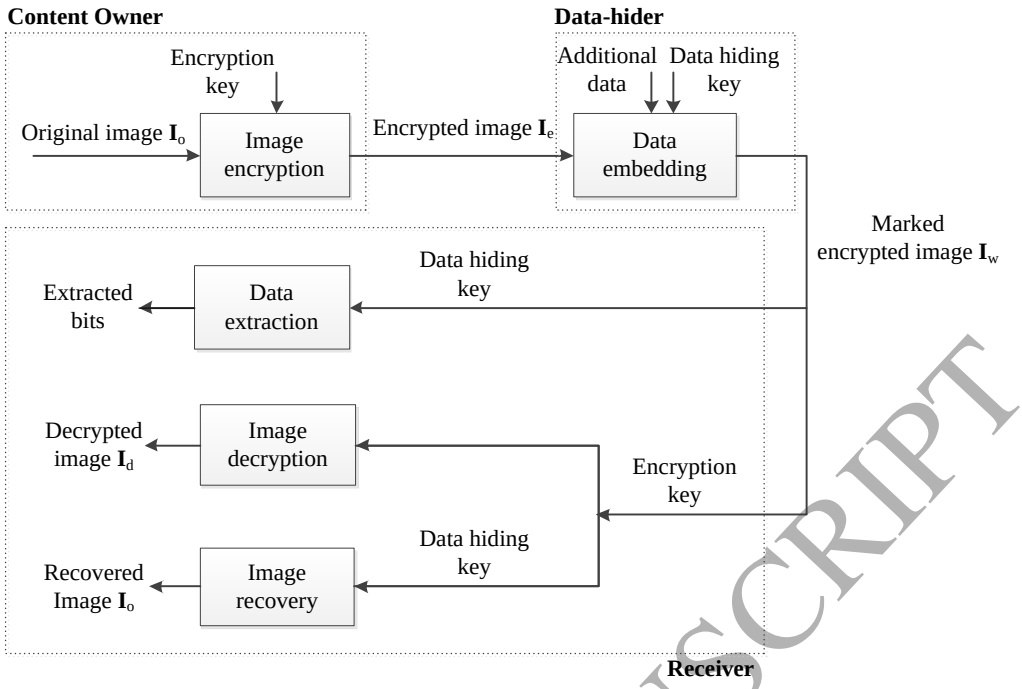


Figure 1 Framework of the proposed scheme

Table 1 Main symbols used in our scheme and their definitions

Symbols	Definitions
$\mathbf{I}_o$	The original image
$\mathbf{I}_e$	The final encrypted image
$\mathbf{I}_w$	The marked, encrypted image
$\mathbf{I}_d$	The directly-decrypted image
M, N	The height and the width of image
$\mathbf{B}_{i,j}$, $\mathbf{C}_{i,j}$	Non-overlapping block sized 2 × 2 in $\mathbf{I}_o$ or $\mathbf{I}_e$
$B_{i,j}^{(x,y)}$, $C_{i,j}^{(x,y)}$	The pixel with the coordinate (x, y) in $\mathbf{B}_{i,j}$ or $\mathbf{C}_{i,j}$
u	The number of LSB layers used for data embedding
$\delta_{i,j}$	Complexity degree of $\mathbf{C}_{i,j}$
T	The complexity threshold
$\mathbf{\Omega}_1$, $\mathbf{\Omega}_2$	Smooth set and complex set
γ	The number of blocks belong to $\mathbf{\Omega}_1$
$\mathfrak{R}_k$	The pixel group of the smooth set $\mathbf{\Omega}_1$
p	The number of pixels in each $\mathfrak{R}_k$
α	The number of bits that can be embedded into each $\mathfrak{R}_k$
τ	The embedding rate

A. Image Encryption

During this stage, suppose that the content owner has an original image $\mathbf{I}_o$ sized $M \times N$, and the gray value of each pixel falls into [0, 255]. First, the original image $\mathbf{I}_o$ is divided into a number of non-overlapping 2×2-sized blocks in raster-scanning order, and the number of blocks is equal to $M \times N/4$. Denote the four pixels of one block $\mathbf{B}_{i,j}$ as two triangle pixels $B_{i,j}^{(0,0)}$ and $B_{i,j}^{(1,1)}$, a circle pixel $B_{i,j}^{(0,1)}$, and a square pixel $B_{i,j}^{(1,0)}$, as illustrated in Figure 2, where i and j are the indices of the block $\mathbf{B}_{i,j}$ ($1 \le i \le M/2$ and $1 \le j \le N/2$). Each of these four pixels $B_{i,j}^{(x,y)}$ can be represented by 8 bits: $b_{i,j}^{(x,y,0)}$, $b_{i,j}^{(x,y,1)}$, …, $b_{i,j}^{(x,y,7)}$, where $(x, y) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$, see Eqs. (1-2).

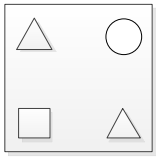


Figure 2 Four pixels in one block $\mathbf{B}_{i,j}$

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

$$B_{i,j}^{(x,y)} = \sum_{s=0}^{7} [2^s \times b_{i,j}^{(x,y,s)}]. \tag{2}$$

In order to protect the confidentiality of image contents, the content owner conducts the encryption operation on each block $\mathbf{B}_{i,j}$ ($1 \le i \le M/2$ and $1 \le j \le N/2$). The operation of image encryption consists of two stages, i.e., analogous stream-cipher encryption and block permutation. First, for the bits $b_{i,j}^{(x,y,s)}$ of the block $\mathbf{B}_{i,j}$, when (x, y, s) belongs to $\mathbf{\Phi}_1 \cup \mathbf{\Phi}_2$, a pseudo-random binary sequence $r_{i,j}^{(x,y,s)}$ generated by the encryption key is utilized to encrypt $b_{i,j}^{(x,y,s)}$ through exclusive-or operation, see Eqs. (3-5).

$$b'^{(x,y,s)}_{i,j} = b_{i,j}^{(x,y,s)} \oplus r_{i,j}^{(x,y,s)}, \quad (x, y, s) \in \mathbf{\Phi}_1 \bigcup \mathbf{\Phi}_2, \tag{3}$$

$$\mathbf{\Phi}_1 = \{(x, y, s) \mid (x, y) \in \{(1,0), (0,1)\} \text{ and } s = 0, 1, \ldots, 7\}, \tag{4}$$

$$\mathbf{\Phi}_2 = \{(x, y, s) \mid (x, y) \in \{(0,0), (1,1)\} \text{ and } s = 0, 1, \ldots, u-1\}, \tag{5}$$

where u denotes the number of LSB layers used for future data embedding, and in our scheme, u is set no greater than 3 for acceptable quality of directly-decrypted image. Then, when (x, y, s) belongs to $\mathbf{\Phi}_3$, another different pseudo-random binary sequence $\rho_{i,j}$ ($1 \le i \le M/2$ and $1 \le j \le N/2$) is also generated by the encryption key to encrypt $b_{i,j}^{(x,y,s)}$:

$$\begin{cases} b'^{(x,y,s)}_{i,j} = \overline{b^{(x,y,s)}_{i,j}}, & \text{if } \rho_{i,j} = 1, \\ b'^{(x,y,s)}_{i,j} = b^{(x,y,s)}_{i,j}, & \text{if } \rho_{i,j} = 0, \end{cases} \quad (x, y, s) \in \mathbf{\Phi}_3, \tag{6}$$

$$\mathbf{\Phi}_3 = \{(x, y, s) \mid (x, y) \in \{(0,0),(1,1)\} \text{ and } s = u, u+1, \ldots, 7\}, \tag{7}$$

where $b'^{(x,y,s)}_{i,j}$ denotes the encrypted result for $b_{i,j}^{(x,y,s)}$ with the above analogous stream-cipher way. Then, all $b'^{(x,y,s)}_{i,j}$ are collected to produce a preliminary encrypted block $\mathbf{B}'_{i,j}$:

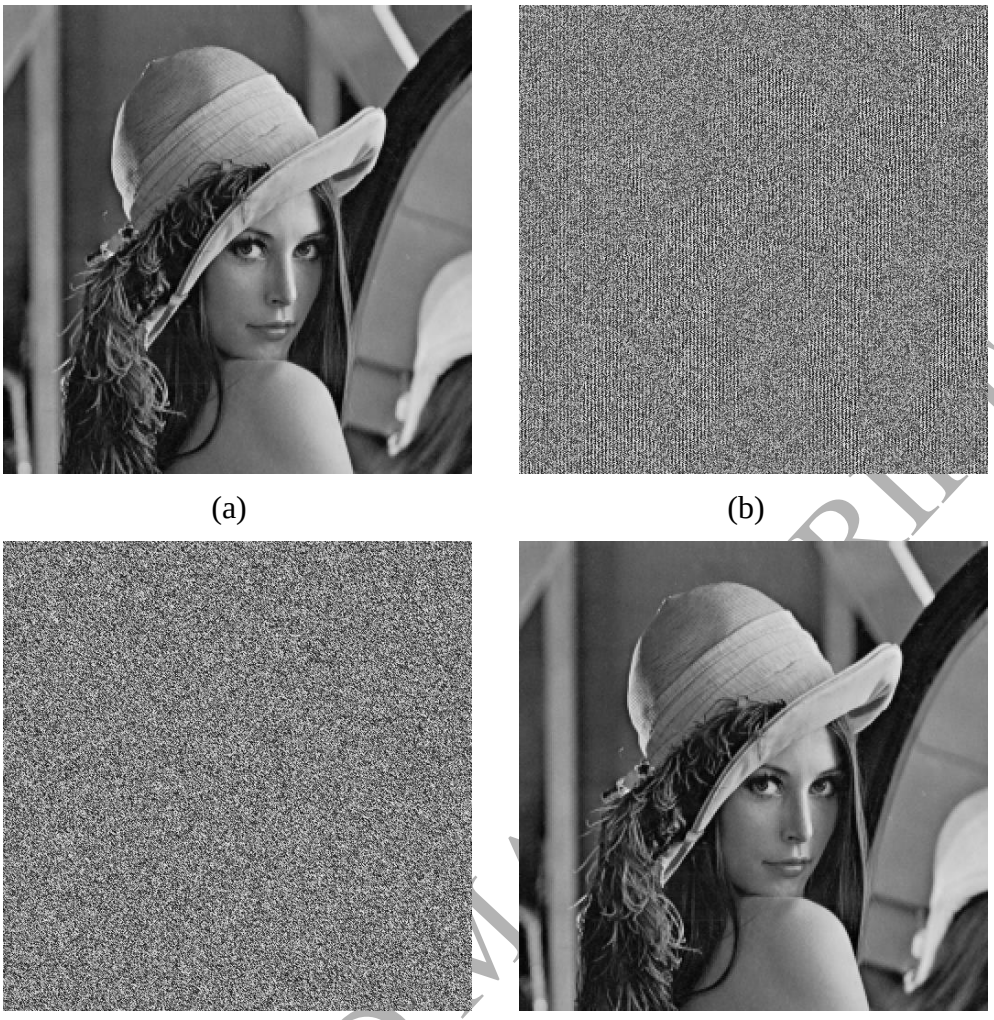
$$B'^{(x,y)}_{i,j} = \sum_{s=0}^{7} [2^s \times b'^{(x,y,s)}_{i,j}], \tag{8}$$

where $B'^{(x,y)}_{i,j}$ denotes one of the four pixels in $\mathbf{B}'_{i,j}$, $(x, y) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$.

In order to further increase the security of image encryption, we permute the locations of all the $M \times N/4$ preliminary encrypted blocks $\mathbf{B}'_{i,j}$ ($1 \le i \le M/2$ and $1 \le j \le N/2$) according to the encryption key to generate the final encrypted image $\mathbf{I}_e$. Note that, as for block permutation, there are at most $(MN/4)!$ different permuted patterns that can be produced. Generally speaking, $(MN/4)!$ is a very large number, and there is almost no one-to-one mapping that can be directly created between the original order and the randomly permuted pattern. Obviously, by the encryption key, the encrypted image can be easily and fully decrypted to the original version through inverse block permutation and the re-generated pseudo-random binary sequences $r_{i,j}^{(x,y,s)}$ and $\rho_{i,j}$. Although other larger block sizes can also be selected, the security performance of the scheme will be affected; on the other hand, larger block sizes will influence the accuracy of further block classification for encrypted blocks by the data-hider. Hence, in our scheme, the block size of 2 × 2 is adopted.

Figure 3 gives an example of the image encryption for *Lena* sized 512 × 512. Figure 3(a) shows original image *Lena*. Figures 3(b-c) are the preliminary encrypted result with analogous stream-cipher encryption and the final encrypted result with block permutation, respectively. Figure 3(d) is the decrypted image that is exactly the same with original image. Figures 4(a-b)

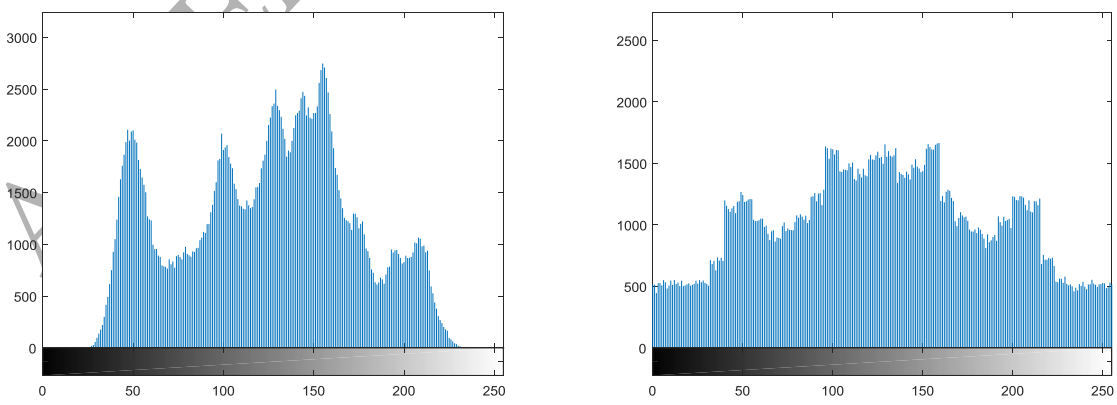
show the histograms of original image *Lena* and its final encrypted version (the two histograms for the images in Figures 3(b) and (c) are the same). Figures 4(c-d) illustrate the distributions of pixel values in original image *Lena* and its final encrypted version, in which the X-axis, Y-axis and Z-axis denote the row index, column index and the pixel values of the corresponding image, respectively. Remarkably, the pixel values of original image are continuously distributed as shown in Figure 4(c), and the distribution of pixel values in the encrypted image is uniform, see Figures 4(d). It can be found from Figures 3 and 4 that, not only the appearances of original image and its encrypted version are visually distinct, but also the encrypted image has quite different histogram and distribution with those of the original image. Besides the statistical histogram, we also evaluate the security of our image encryption method with respect to pixel correlation and information entropy. The correlation values of all adjacent pixels in the horizontal, vertical and diagonal directions for the original image and the encrypted image are calculated, respectively. Obviously, the adjacent pixels in the original image are usually highly correlated and the corresponding correlation values are close to 1. After image encryption, the adjacent pixels are decorrelated, and the corresponding correlation values are declining rapidly (close to 0), see Table 2. In order to show the results intuitively, the correlation distributions of adjacent pixels are also illustrated. The first row, i.e., (a-c) and the second row, i.e., (d-f), of Figures 5 show the distributions of horizontal, vertical and diagonal correlation for original image and encrypted image, respectively. It can be observed that, compared with the adjacent pixels in the original image, the adjacent pixels in the encrypted image are significantly scattered. Information entropy is one of the most important features for randomness evaluation. Obviously, the greater information entropy is, the more randomness and the uncertainty the system have. The information entropies of the original image and the encrypted image are also shown in Table 2. It can be found from the results that, the encrypted image has greater entropy than the original image. Therefore, although the security of our image encryption method is indeed somewhat weaker than the standard stream cipher, however, based on the above analysis of statistical histogram, pixel correlation and information entropy, the security of our image encryption method is generally acceptable, which can effectively satisfy the protection requirement for the image contents.



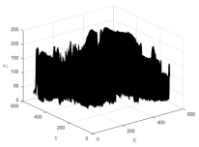
(a) (b)

(c) (d)

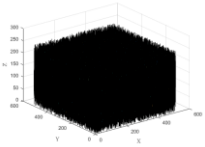
Figure 3 An example of the image encryption for *Lena*. (a) Original image, (b) Preliminary encrypted result with analogous stream-cipher encryption, (c) Final encrypted result with block permutation, (d) Decrypted image.



(a) (b)



(c)

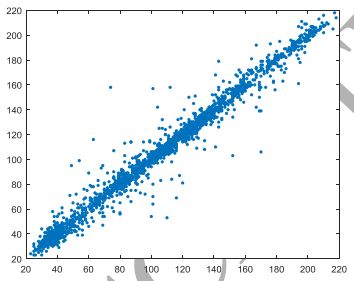


(d)

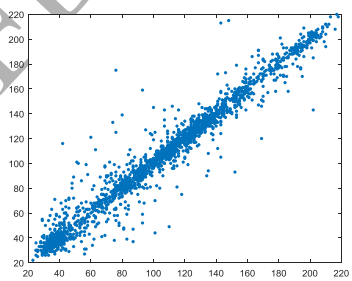
Figure 4 Histograms and distributions of original image *Lena* and its final encrypted version. (a) Histogram of original image, (b) Histogram of encrypted image, (c) Distribution of pixel values in original image, (d) Distribution of pixel values in encrypted image.

Table 2 Pixel correlation and information entropy of original image and encrypted image

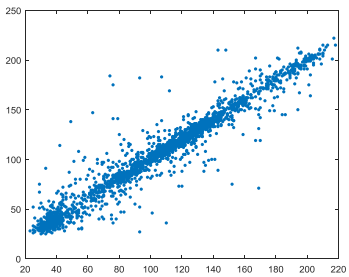
Images	Pixel correlation			Information entropy
	Horizontal	Vertical	Diagonal	
Original image	0.9862	0.9755	0.9629	7.3871
Encrypted image	–0.0011	–0.0017	0.1445	7.8989



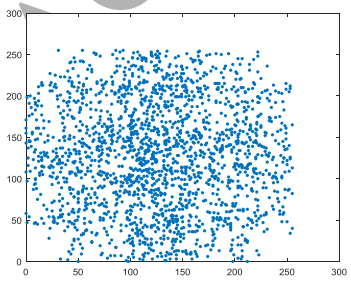
(a)



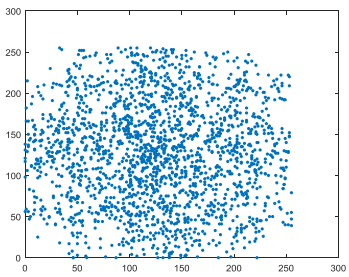
(b)



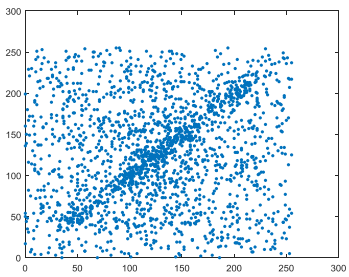
(c)



(d)



(e)



(f)

Figure 5 Distributions of pixel correlation of original image *Lena* and its final encrypted version.

(a) Horizontal direction of original image, (b) Vertical direction of original image, (c) Diagonal direction of original image, (d) Horizontal direction of encrypted image, (e) Vertical direction of encrypted image, (f) Diagonal direction of encrypted image.

B. Data Embedding

When the encrypted image $\mathbf{I}_e$ is received from the content owner, the data hider can embed the additional data into $\mathbf{I}_e$. First, the data hider also divides the received, encrypted image $\mathbf{I}_e$ into $M \times N/4$ non-overlapping blocks $\mathbf{C}_{i,j}$ sized 2×2 ($1 \le i \le M/2$ and $1 \le j \le N/2$), and the four pixels in each encrypted block $\mathbf{C}_{i,j}$ from left to right, then from top to bottom are $C_{i,j}^{(0,0)}$, $C_{i,j}^{(0,1)}$, $C_{i,j}^{(1,0)}$, $C_{i,j}^{(1,1)}$, respectively. Then, the data hider calculates the absolute value $\delta_{i,j}$ between the top-left pixel and the bottom-right pixel in the each encrypted block $\mathbf{C}_{i,j}$:

$$\delta_{i,j} = \left| 2^u \cdot \left\lfloor C_{i,j}^{(0,0)} / 2^u \right\rfloor - 2^u \cdot \left\lfloor C_{i,j}^{(1,1)} / 2^u \right\rfloor \right|. \tag{9}$$

It can be observed from Eq. (9) that, the value of $\delta_{i,j}$ is only dependent on the $(8 - u)$ MSBs of the block $\mathbf{C}_{i,j}$, which is not changed before and after encryption and can be used to reflect the complexity of the decrypted version for $\mathbf{C}_{i,j}$. A greater value of $\delta_{i,j}$ indicates that the decrypted result of the block $\mathbf{C}_{i,j}$ has a relatively complex distribution of image contents. Then, even though without knowing the contents of original image, the data hider can utilize a threshold T to classify all encrypted blocks into two sets, i.e., smooth set $\mathbf{\Omega}_1$ and complex set $\mathbf{\Omega}_2$:

$$\begin{cases} \mathbf{C}_{i,j} \in \mathbf{\Omega}_1, & \text{if } \delta_{i,j} \le T, \\ \mathbf{C}_{i,j} \in \mathbf{\Omega}_2, & \text{if } \delta_{i,j} > T. \end{cases} \tag{10}$$

Since only the blocks belonging to $\mathbf{\Omega}_1$ are used to hide the additional data in our scheme, thus, if more additional bits are required to embed into the encrypted image, greater value of the threshold T should be applied. On the other hand, smaller value of T will lead to lower hiding capacity of additional data and better visual quality of the directly-decrypted image.

Figure 6 shows block classification results for the encrypted image *Lena* and corresponding results after inverse block permutation, in which the white and the black regions represent the blocks belonging to smooth set $\mathbf{\Omega}_1$ and complex set $\mathbf{\Omega}_2$, respectively. Figures 6(a) and (c) are the classification results on encrypted image with $T = 20$ and $T = 10$, respectively. Figures 6(b) and (d) are the results after conducting inverse block permutation on (a) and (c) with the encryption

key, respectively, which demonstrates the effectiveness of block classification.

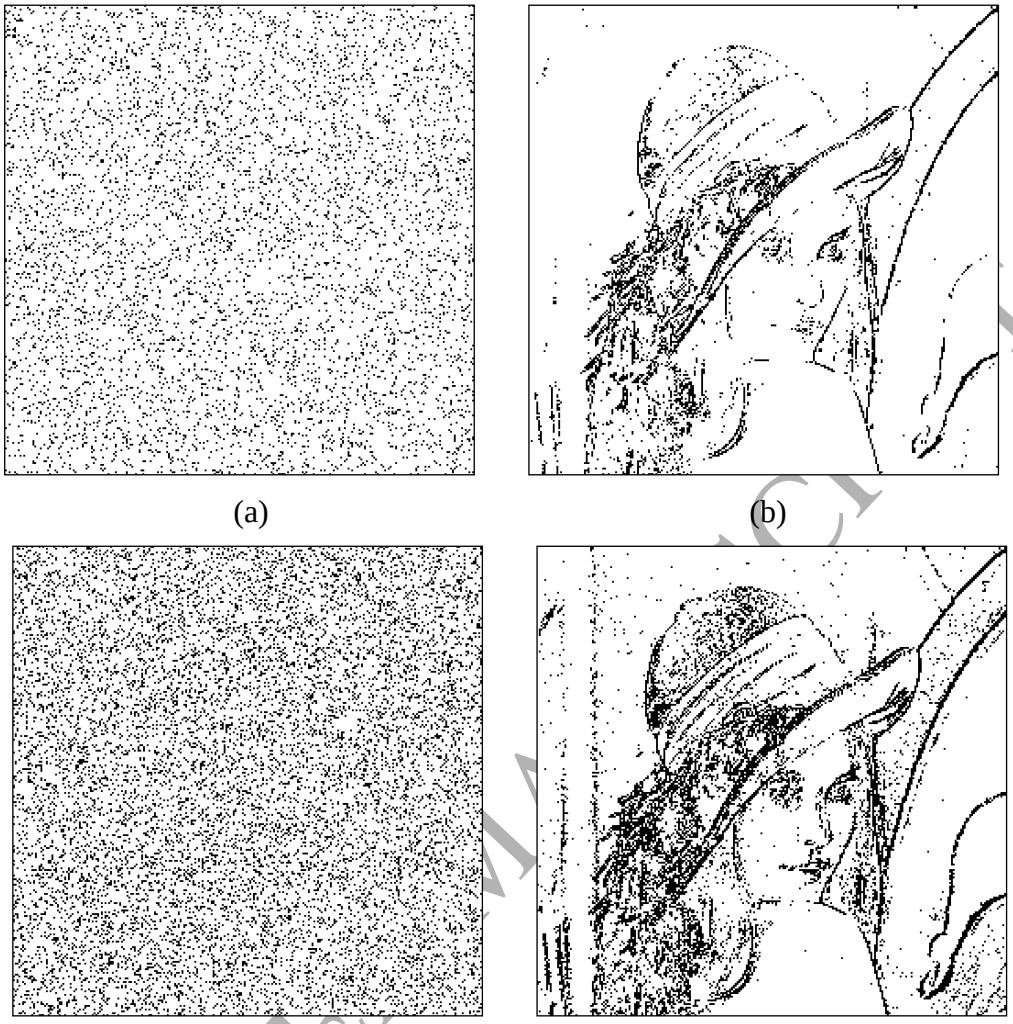


Figure 6 Block classification results for the encrypted image *Lena*, in which the white and the black regions represent the blocks belonging to smooth set $\mathbf{\Omega}_1$ and complex set $\mathbf{\Omega}_2$, respectively. (a-b) Block classification result when T = 20, and corresponding result after inverse block permutation, (c-d) Block classification result when T = 10, and corresponding result after inverse block permutation.

Denote the number of blocks belong to $\mathbf{\Omega}_1$ as γ ($\leq M \times N/4$). By the data-hiding key, the data hider randomly segments the pixels of the blocks belonging to the smooth set $\mathbf{\Omega}_1$ into a series of groups, and each group contains p pixels. Note that, the four pixels in one block belonging to $\mathbf{\Omega}_1$ are arranged in the same group. Thus, p is a multiple of four, and the group number is equal to $4\gamma / p$. For each pixel group $\mathfrak{R}_k$ (k = 1, 2, …, $4\gamma / p$) from the smooth set $\mathbf{\Omega}_1$, collect the binary bits in the u LSB layers of its p pixels and denote them as $\mathbf{V}_k = \{v(k, 1), v(k, 2), \ldots, v(k, l)\}$, where $l = u \cdot p$. A binary matrix $\mathbf{G}$ is then generated by the data hider with the size of $(l - \alpha) \times l$,

which is comprised of two parts:

$$\mathbf{G} = [\mathbf{E}_{l-\alpha}, \mathbf{Q}], \tag{11}$$

where $\mathbf{E}$ is the identity matrix sized $(l - \alpha) \times (l - \alpha)$, and $\mathbf{Q}$ is a pseudo-random binary matrix sized $(l - \alpha) \times \alpha$ controlled by the data-hiding key. Here, α is the embedding parameter that is a positive integer much smaller than l. The row sparsity of the matrix $\mathbf{G}$ leads to the low complexity of the calculation and the randomness of $\mathbf{Q}$ in $\mathbf{G}$ enhances the security of the data embedding process. Then, the data hider transforms each group $\mathbf{V}_k$ with l bits into $\mathbf{Z}_k$ with $(l - \alpha)$ bits, see Eq. (12).

$$\begin{bmatrix} z(k,1) \\ z(k,2) \\ \vdots \\ z(k,l-\alpha) \end{bmatrix} = \mathbf{G} \cdot \begin{bmatrix} v(k,1) \\ v(k,2) \\ \vdots \\ v(k,l) \end{bmatrix}, \tag{12}$$

where the arithmetic in Eq. (12) is modulo-2, and $\mathbf{Z}_k = \{z(k, 1), z(k, 2), \ldots, z(k, l-\alpha)\}$. In this way, a spare room of α bits corresponding to each group $\mathbf{V}_k$ is vacated for embedding additional data. Then, replace the $(l - \alpha)$ bits, i.e., $v(k, 1)$, $v(k, 2)$, …, $v(k, l-\alpha)$, and the α bits, i.e., $v(k, l-\alpha+1)$, $v(k, l-\alpha+2)$, …, $v(k, l)$, in $\mathbf{V}_k$ with $z(k, 1)$, $z(k, 2)$,…, $z(k, l-\alpha)$ of $\mathbf{Z}_k$ and the α-bits secret data to be embedded, respectively, and then arrange the newly-replaced l bits back to the corresponding u LSB layers of the p pixels in $\mathfrak{R}_k$. After all pixel groups $\mathfrak{R}_k$ (k = 1, 2, …, $4\gamma / p$) from the blocks belonging to the set $\mathbf{\Omega}_1$ are employed with above procedure, the total $4\alpha \cdot \gamma / p$ bits are embedded, and the marked, encrypted image $\mathbf{I}_w$ can be generated.

Since the values of encrypted pixels in $\mathbf{\Omega}_2$ and the $(8 - u)$ MSBs of encrypted pixels in $\mathbf{\Omega}_1$ are kept unchanged during data embedding, the visual quality of directly-decrypted image on the receiver side can be excellent. In addition, the security of the encrypted image is also effectively guaranteed in the encryption stage.

In our scheme, the embedding parameters u, T, p and α are designed as the integers belonging to [1, 3], [0, 255], [1, 4800] and [1, 16], respectively. Thus, the binary representations for u, T, p and α occupy 2, 8, 13 and 4 bits, respectively, and the total bit number of binary representations for these four parameter requires 27 bits. There are two ways to transmit these

four parameters from the data-hider to the receiver: 1) The data hider encrypts and transmits these 27 bits of parameters as auxiliary information together with the marked, encrypted image $\mathbf{I}_w$ to the receiver. 2) The data hider chooses the first 27 pixels of the encrypted image $\mathbf{I}_e$ and replaces their LSB with the 27 bits of parameters. The original 27 bits in the replaced LSB are concatenated at the front of the additional bits and then embedded into the remaining pixels of the encrypted image. Thus, after receiving the marked, encrypted image $\mathbf{I}_w$, the parameters u, T, p and α can be correctly retrieved from the first 27 pixels and the operations of data extraction, image decryption and image recovery can then be performed on the receiver side.

C. Data Extraction and Image Recovery

In this procedure, three cases are considered for data extraction and image recovery, i.e., 1) the receiver only has the data-hiding key, 2) the receiver only has the encryption key, and 3) the receiver has both the data-hiding key and the encryption key.

1) If the receiver only has the data-hiding key, he/she can correctly extract all embedded bits from the marked, encrypted image $\mathbf{I}_w$. Detailedly, the receiver first divides $\mathbf{I}_w$ into $M \times N/4$ non-overlapping blocks sized 2×2. Then, since the $(8 - u)$ MSB layers of $\mathbf{I}_e$ and $\mathbf{I}_w$ are the same, through the same way of the data hider, the receiver can classify the $M \times N/4$ blocks of $\mathbf{I}_w$ into the smooth set $\mathbf{\Omega}_1$ and the complex set $\mathbf{\Omega}_2$ with the assist of the threshold T. Next, with the data-hiding key, the 4γ pixels in the γ blocks belonging to $\mathbf{\Omega}_1$ are segmented into $4\gamma / p$ groups, i.e., $\mathfrak{R}_k$ ($k = 1, 2, \ldots, 4\gamma / p$). By collecting the bits in the u LSB layers of each pixel group $\mathfrak{R}_k$, the corresponding α embedded data bits can be retrieved. After all pixel groups finish the above procedure, totally $4\alpha \cdot \gamma / p$ embedded bits can be extracted.

2) If the receiver only has the encryption key, he/she can decrypt the marked, encrypted image $\mathbf{I}_w$ to obtain a directly-decrypted image $\mathbf{I}_d$ that is visually similar to $\mathbf{I}_o$. Detailedly, the receiver divides $\mathbf{I}_w$ into $M \times N/4$ non-overlapping blocks sized 2×2 and classifies them into the smooth set $\mathbf{\Omega}_1$ and the complex set $\mathbf{\Omega}_2$ with the threshold T. With the image encryption method described in Section 2-A, the pseudo-random binary sequences $r_{i,j}^{(x, y, s)}$ and $\rho_{i,j}$ can be re-generated by the encryption key. Then, the eight bit-layers of the pixels belonging to $\mathbf{\Omega}_2$ and the $(8 - u)$ MSBs of the pixels belonging to $\mathbf{\Omega}_1$ are first decrypted, and then all $M \times N/4$ blocks are inversely permuted back to their original locations in the image. Thus, the directly-decrypted

image $\mathbf{I}_{\mathrm{d}}$ can be obtained. Since $(8-u)$ MSBs of the pixels belonging to $\mathbf{\Omega}_1$ and the values of the pixels belonging to $\mathbf{\Omega}_2$ are the same as those of the original image $\mathbf{I}_{\mathrm{o}}$, the directly-decrypted image $\mathbf{I}_{\mathrm{d}}$ can achieve the satisfactory image quality.

3) When the data-hiding key and encryption key are simultaneously available, the receiver can not only extract the additional data, but also recover the original image. As described above, with the data-hiding key, the 4γ pixels in the γ blocks belonging to $\mathbf{\Omega}_1$ from $\mathbf{I}_{\mathrm{w}}$ are segmented into $4\gamma / p$ groups, i.e., $\mathfrak{R}_k$ (k = 1, 2, …, $4\gamma / p$), and the $(l-\alpha)$ transformed bits, i.e., $\mathbf{Z}_k = \{z(k, 1), z(k, 2), \ldots, z(k, l-\alpha)\}$, in the u LSB layers of each pixel group $\mathfrak{R}_k$ can be collected, see Eq. (12). In the following, the main task is to find the encrypted l bits of the u LSB layers for each pixel group $\mathfrak{R}_k$ before data embedding, i.e., $\mathbf{V}_k = \{v(k, 1), v(k, 2), \ldots, v(k, l)\}$, and decrypt them to recover the original image $\mathbf{I}_{\mathrm{o}}$.

The receiver first generates the matrix $\mathbf{H}$ through the data-hiding key:

$$\mathbf{H} = [\mathbf{Q}', \mathbf{E}_\alpha], \tag{13}$$

where $\mathbf{H}$ is a binary matrix sized $\alpha \times l$, consisting of the transpose of $\mathbf{Q}$ and an $\alpha \times \alpha$ identity matrix $\mathbf{E}_\alpha$. Here, we define a set $\mathbf{\Theta}_k$ of 2^α vectors sized $1 \times l$:

$$\mathbf{\Theta}_k = \{\mathbf{\Lambda}_k^{(1)}, \mathbf{\Lambda}_k^{(2)}, \ldots, \mathbf{\Lambda}_k^{(2^\alpha)}\}, \tag{14}$$

$$\mathbf{\Lambda}_k^{(t)} = \left[z(k,1), z(k,2), \ldots, z(k,l-\alpha), 0, 0, \ldots, 0\right] + \mathbf{\Gamma}_t \cdot \mathbf{H}, \quad t = 1, 2, \ldots, 2^\alpha, \tag{15}$$

where $\mathbf{\Gamma}_t$ is an arbitrary $1 \times \alpha$ binary vector. Thus, according to Eq. (12), the vector $\mathbf{V}_k = [v(k, 1), v(k, 2), \ldots, v(k, l)]$ must be one of the 2^α vectors in set $\mathbf{\Theta}_k$.

For each pixel group $\mathfrak{R}_k$ (k = 1, 2, …, $4\gamma / p$) in $\mathbf{I}_{\mathrm{w}}$, the receiver replaces its u LSB layers of the p pixels with the l bits of each vector $\mathbf{\Lambda}_k^{(t)}$ in $\mathbf{\Theta}_k$, and decrypts the image with the encryption key to produce the temporary image $\mathbf{I}_{\mathrm{d}}^{(k, t)}$, t =1, 2, …, 2^α. Obviously, the $(8-u)$ MSB layers of $\mathbf{I}_{\mathrm{d}}^{(k, t)}$ are the same as the original image $\mathbf{I}_{\mathrm{o}}$. Then, in the image $\mathbf{I}_{\mathrm{d}}^{(k, t)}$, the accumulated difference $D_k^{(t)}$ between the pixels in $\mathfrak{R}_k$ and their estimated values based on the $(8-u)$ MSB layers of the neighboring pixels are calculated:

$$D_k^{(t)} = \sum_{(x, y) \in \mathfrak{R}_k} \left| \beta_{x,y}^{(t)} - \hat{\beta}_{x,y} \right|, \quad t = 1, 2, \ldots, 2^\alpha, \tag{16}$$

$$\hat{\beta}_{x,y} = 2^u \times (\lambda_1 \cdot \widehat{\beta}_{x,y} + \lambda_2 \cdot \breve{\beta}_{x,y}) + 2^{u-1}, \tag{17}$$

$$\widehat{\beta}_{x,y} = \frac{\lfloor \beta^{(t)}_{x-1,y} / 2^u \rfloor + \lfloor \beta^{(t)}_{x+1,y} / 2^u \rfloor + \lfloor \beta^{(t)}_{x,y-1} / 2^u \rfloor + \lfloor \beta^{(t)}_{x,y+1} / 2^u \rfloor}{4}, \tag{18}$$

$$\breve{\beta}_{x,y} = \frac{\lfloor \beta^{(t)}_{x-1,y-1} / 2^u \rfloor + \lfloor \beta^{(t)}_{x+1,y-1} / 2^u \rfloor + \lfloor \beta^{(t)}_{x-1,y+1} / 2^u \rfloor + \lfloor \beta^{(t)}_{x+1,y+1} / 2^u \rfloor}{4}, \tag{19}$$

where $\beta_{x,y}^{(t)}$ denotes the pixel in the temporary image $\mathbf{I}_{\mathrm{d}}^{(k,\,t)}$ with the coordinate (x, y), and λ_1 and λ_2 are the two weights for pixel-value estimation ($\lambda_1 \geq \lambda_2$ and $\lambda_1 + \lambda_2 \equiv 1$). Note that, for the image border pixels, their estimated values are equal to the average values of their two or three neighboring pixels in the horizontal and vertical directions. Due to the characteristic of smoothness for natural images, the vector $\mathbf{\Lambda}_k^{(t)}$ that leads to the smallest accumulated difference $D_k^{(t)}$, i.e., $\mathbf{\Lambda}_k^{(t^*)}$, is regarded as the correct result for the vector $\mathbf{V}_k$ that is the encrypted u LSB layers of the pixel group $\mathfrak{R}_k$ before data embedding.

$$t^* = \arg\min_t D_k^{(t)}, \quad t = 1, 2, \ldots, 2^{\alpha}, \tag{20}$$

After all pixel groups $\mathfrak{R}_k$ (k = 1, 2, …, $4\gamma / p$) in $\mathbf{I}_{\mathrm{w}}$ are conducted with above procedure, their encrypted u LSB layers before embedding can be recovered, which can then be decrypted correctly by the encryption key. Finally, through combining the decrypted u LSB layers of the pixels belonging to $\mathbf{\Omega}_1$ with the decrypted $(8 - u)$ MSBs of the pixels belonging to $\mathbf{\Omega}_1$ and the decrypted pixels belonging to $\mathbf{\Omega}_2$, the original image $\mathbf{I}_{\mathrm{o}}$ can successfully recovered.

In our scheme, the content owner encrypts the original image through a simply modified stream cipher (rather than the standard stream cipher of VRAE schemes) and block permutation, and after image encryption, only the encrypted pixel groups in $\mathbf{\Omega}_1$ corresponding to the smooth region of original image are chosen by the data-hider to conduct the matrix-compression based embedding (rather than reserving room before encryption of RRBE schemes). Therefore, our scheme can be considered as vacating room within and after encryption, which is between VRAE and RRBE. Compared with the schemes of VRAE, such as [17, 19, 23], our scheme can not only achieve better visual quality of directly-decrypted image and embedding rate, but also superior performance of recovery accuracy. Compared with the schemes of RRBE, such as [28,

29, 31], the performance of our scheme with respect to embedding rate and directly-decrypted image quality are indeed weaker. But, the computation burden on the content owner side of our scheme is much lighter than that of the RRBE schemes, which is more suitable for the user client with lower computation capability and energy.

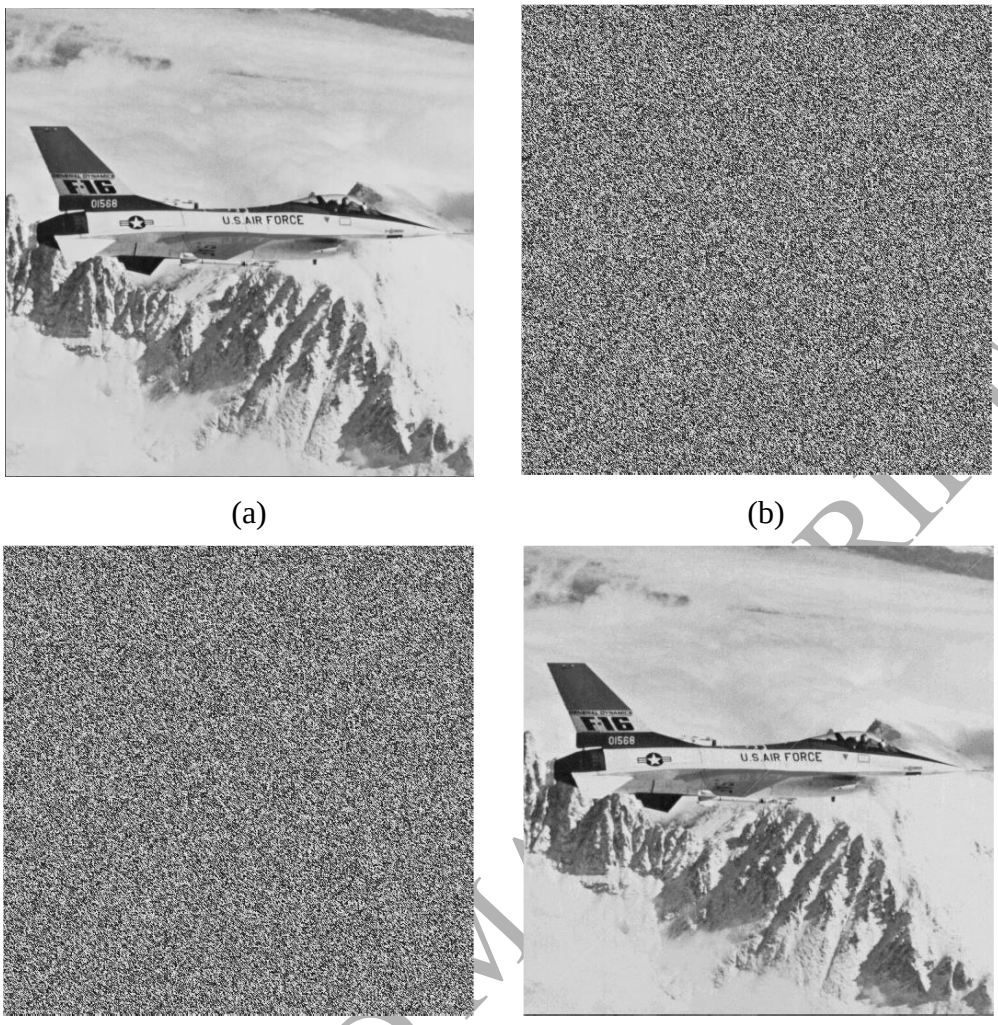
3. Experimental Results and Analysis

In order to demonstrate the effectiveness and superiority of our scheme, experiments were carried out on a large number of standard images, and the environment of our experiments was based on a personal computer with a 3.30 GHz Intel i3 processor, 4.00 GB memory, Windows 7 operating system, and Matlab 7.

Figures 7-8 show the results of our scheme, under the parameters $T = 40$, $u = 3$, $p = 240$, $\alpha = 5$, $\lambda_1 = 0.9$ and $\lambda_2 = 0.1$ for the two test images *Airplane* and *Man* both sized 512 × 512. Subfigures (a-d) in Figures 7-8 are the original images, the encrypted images, the marked, encrypted images with the embedding rate τ of 0.020 bit per pixel (bpp), and the directly-decrypted images with peak signal-to-noise ratios (PSNR) of 41.4 dB and 41.2 dB for *Airplane* and *Man*, respectively. Note that, the original images can be exactly recovered from the marked, encrypted images with the data-hiding key and encryption key, and embedding rate τ can be calculated with Eq. (21).

$$\tau = \frac{4 \cdot \alpha \cdot \gamma}{p \cdot M \cdot N}. \tag{21}$$

We also conducted the experiments on other different test images, such as *Lena*, *Baboon* and the 1338 various images of the uncompressed color image database (UCID) [38]. For color images, the luminance components were used for testing. Figure 9 gives the curves of PSNR values for directly-decrypted images ($PSNR_d$) with respect to different embedding rates τ, and the maximum value of PSNR was 57.8 dB, and the minimum was slightly greater than 41 dB.



(a) (b)

(c) (d)

Figure 7 Results of the proposed scheme for *Airplane*. (a) Original image, (b) Encrypted image, (c) Marked, encrypted image (τ = 0.020 bpp), (d) Directly decrypted image with PSNR of 41.4 dB.

(a) (b)

(c) (d)

Figure 8 Results of the proposed scheme for *Man*. (a) Original image, (b) Encrypted image, (c) Marked, encrypted image (τ = 0.020 bpp), (d) Directly-decrypted image with PSNR of 41.2 dB.

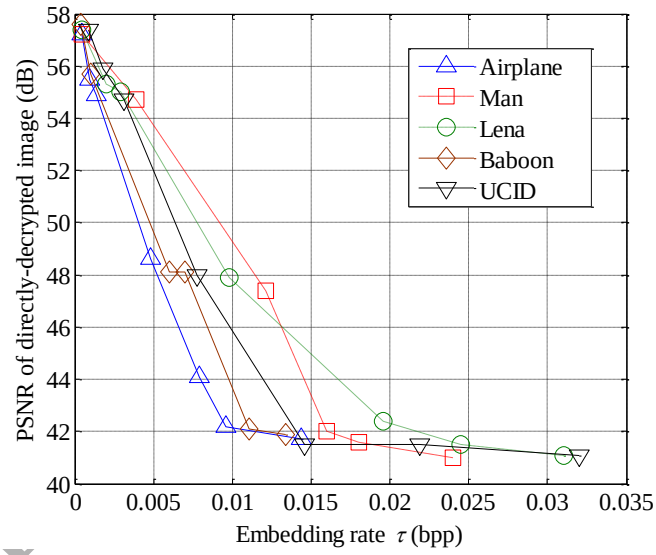


Figure 9 PSNR of directly-decrypted image with respect to different embedding rates

We also analyzed the influences of different parameters on the performance of our scheme. Table 3 lists the experimental results under the different thresholds T (which is used to classify the encrypted image blocks into two sets $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ during data embedding) for *Lena* and *Baboon* both sized 512 × 512, including the percentage of blocks belonging to $\mathbf{\Omega}_1$ among all the blocks (i.e., $4\gamma/MN$), embedding rate τ, and PSNR of the directly-decrypted images (PSNR_d). Here, the values of u, p, and α are 3, 240, and 5, respectively. It can be found from Table 3 that, when the threshold T becomes smaller, the percentage of the blocks belonging to $\mathbf{\Omega}_1$ (actually used for data embedding) and the embedding rate would decrease correspondingly. In addition,

with the same value of T, the original image with relatively smooth distribution (e.g. *Lena*) achieves greater embedding rate τ than the complex image (e.g. *Baboon*). Table 4 gives the performance of our scheme under different parameters of u (the number of LSB layers used for data embedding), α (the bit number embedded into each pixel group $\mathfrak{R}_k$) and p (the pixel number in each $\mathfrak{R}_k$), in which embedding rate τ, PSNR of directly decrypted images ($PSNR_d$) and PSNR of recovered images ($PSNR_r$) are listed. It can be observed from Table 4 that, greater value of α leads to a higher embedding rate τ. Also, smaller values of α and u lead to better quality of directly-decrypted images because more bits in the encrypted image are not modified during data embedding. The symbol "+∞" in Table 4 indicates that $PSNR_r$ is positive infinity and the original images can be recovered without any error. Larger value of u and smaller value of α can be helpful to the perfect image recovery since more useful data in $\mathbf{Z}_k$ and less candidate vectors $\mathbf{\Gamma}_t$ are involved in the recovery procedure, see Eqs. (12) and (15).

Table 3 Percentage of the blocks in $\mathbf{\Omega}_1$ ($4\gamma/MN$), embedding rate (τ), and PSNR of directly-decrypted images ($PSNR_d$) under different thresholds T

Threshold	*Lena*			*Baboon*		
	$4\gamma/MN$	τ	$PSNR_d$	$4\gamma/MN$	τ	$PSNR_d$
T = 5	46.01%	0.0096	44.5	21.42%	0.0044	47.8
T = 10	82.74%	0.0172	41.9	52.07%	0.0108	43.9
T = 20	91.18%	0.0190	41.5	68.14%	0.0140	42.8
T = 40	97.93%	0.0204	41.2	89.11%	0.0186	41.6
T = 60	99.11%	0.0206	41.1	94.79%	0.0197	41.3
T = 80	99.71%	0.0207	41.1	98.50%	0.0205	41.1
T = 100	99.86%	0.0208	41.1	99.42%	0.0207	41.2

Table 4 Embedding rate (τ), PSNR of directly-decrypted images ($PSNR_d$) and PSNR of recovered images ($PSNR_r$) under different parameters u, p and α

	u	p	α = 1	α = 2	α = 3	α = 4
Airplane	1	1200	0.0008, 57.5, +∞	0.0016, 55.8, +∞	0.0024, 54.9, +∞	0.0032, 54.7, +∞
	2	400	0.0024, 50.4, +∞	0.0048, 48.3, +∞	0.0072, 47.8, +∞	0.0096, 47.5, +∞

	3	160	0.0060, 44.2, +∞	0.0120, 42.6, +∞	0.0180, 41.9, +∞	0.0240, 41.6, +∞
Man	1	1200	0.0008, 57.7, +∞	0.0016, 55.7, +∞	0.0024, 54.7, +∞	0.0032, 54.4, 74.6
	2	400	0.0024, 50.0, +∞	0.0049, 48.2, +∞	0.0073, 47.7, +∞	0.0097, 47.5, 70.6
	3	160	0.0061, 44.1, +∞	0.0121, 42.2, +∞	0.0182, 41.6, 74.2	0.0243, 41.4, 69.1
Lena	1	1200	0.0008, 57.4, +∞	0.0016, 55.4, +∞	0.0024, 54.9, +∞	0.0033, 54.5, 77.8
	2	400	0.0024, 50.2, +∞	0.0049, 48.5, +∞	0.0073, 47.8, +∞	0.0098, 47.4, +∞
	3	120	0.0082, 44.1, +∞	0.0163, 41.9, +∞	0.0245, 41.7, +∞	0.0326, 41.3, +∞
Baboon	1	4800	0.0002, 57.0, +∞	0.0004, 56.1, +∞	0.0005, 55.0, 71.5	0.0007, 54.9, 68.8
	2	1400	0.0006, 50.8, +∞	0.0013, 49.2, +∞	0.0019, 48.3, +∞	0.0025, 48.1, 66.8
	3	400	0.0022, 44.5, +∞	0.0045, 42.6, +∞	0.0067, 41.8, 69.5	0.0089, 41.8, 66.3
UCID	1	1000	0.0009, 57.1, +∞	0.0018, 55.9, +∞	0.0026, 55.1, +∞	0.0031, 54.7, +∞
	2	320	0.0027, 50.5, +∞	0.0063, 48.7, +∞	0.0078, 48.0, +∞	0.0110, 47.7, +∞
	3	240	0.0037, 44.5, +∞	0.0078, 42.4, +∞	0.0110, 41.7, +∞	0.0146, 41.5, +∞

In order to show the superiority of our scheme, comparisons were conducted between our scheme and four state-of-the-art schemes [17, 19, 23, 24]. Figures 10-11 show the rate-distortion curves for the directly-decrypted images *Airplane*, *Man*, *Lena* and *Baboon*. Besides the traditional PSNR, the typical index of structural similarity (SSIM) [39], which integratedly considers the information of luminance, contrast and structure, was also utilized to measure visual quality of directly-decrypted images. Calculation method for SSIM is given in Eqs. (22) and (23), and details can be found in [39].

$$\Phi(\mathbf{X}, \mathbf{Y}) = \frac{1}{K} \cdot \sum_{i=1}^{K} \varphi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}), \tag{22}$$

$$\varphi(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) = \frac{(2\mu_x^{(i)}\mu_y^{(i)} + C_1) \cdot (2\sigma_{xy}^{(i)} + C_2)}{\{[\mu_x^{(i)}]^2 + [\mu_y^{(i)}]^2 + C_1\} \cdot \{[\sigma_x^{(i)}]^2 + [\sigma_y^{(i)}]^2 + C_2\}}, \tag{23}$$

where $\mathbf{X}$ and $\mathbf{Y}$ denote the original image and the modified image for evaluation, $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ denote the i-th block of $\mathbf{X}$ and $\mathbf{Y}$, K is the block number, $\mu_x^{(i)}$ and $\mu_y^{(i)}$ are the mean values of $\mathbf{x}^{(i)}$

and $\mathbf{y}^{(i)}$, $\sigma_x^{(i)}$ and $\sigma_y^{(i)}$ are the standard deviations of $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$, $\sigma_{xy}^{(i)}$ is the covariance of $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$, and C_1 and C_2 are small constants near zero. The value $\Phi(\mathbf{X}, \mathbf{Y})$ of SSIM index belongs to [0, 1], and the greater the SSIM index is, the better visual quality of the evaluated image **Y** with respect to the original image **X** is. In subfigures (a-d) of Figures 10-11, the abscissa represents the embedding rate τ, and the ordinate denotes the values of PSNR or SSIM index. The points on the curves were derived from different parameter values under the condition that original images can be perfectly recovered with the encryption and data-hiding keys. It can be found in Figures 10-11 that, through the block classification strategy during data embedding, the proposed scheme can achieve significantly better rate-distortion performance for the directly-decrypted image than the schemes [17, 19, 23, 24] no matter for smooth images or complex images.

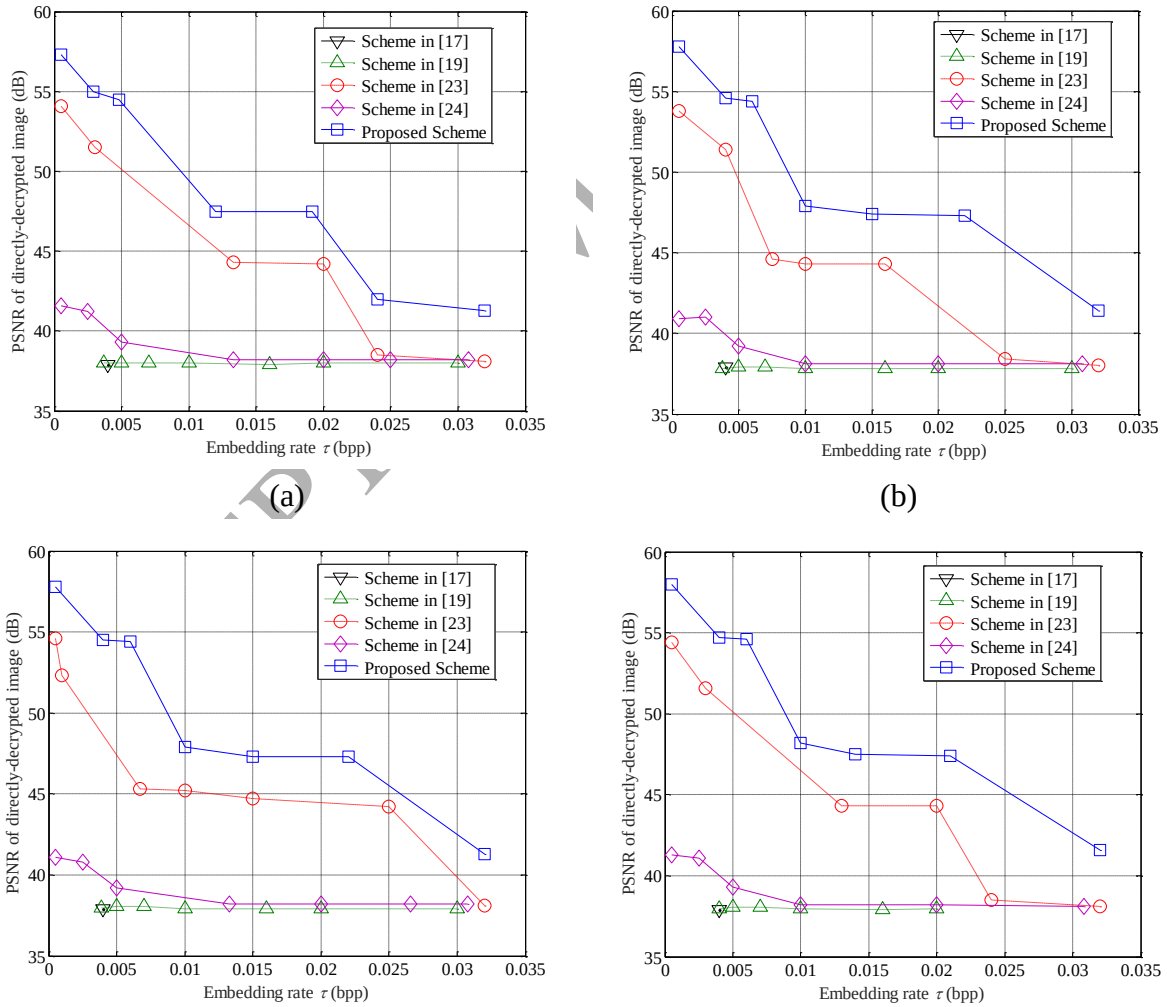


Figure 10 Rate-PSNR comparisons between the proposed scheme and [17, 19, 23, 24]. (a)

Airplane, (b) *Man*, (c) *Lena*, (d) *Baboon*.

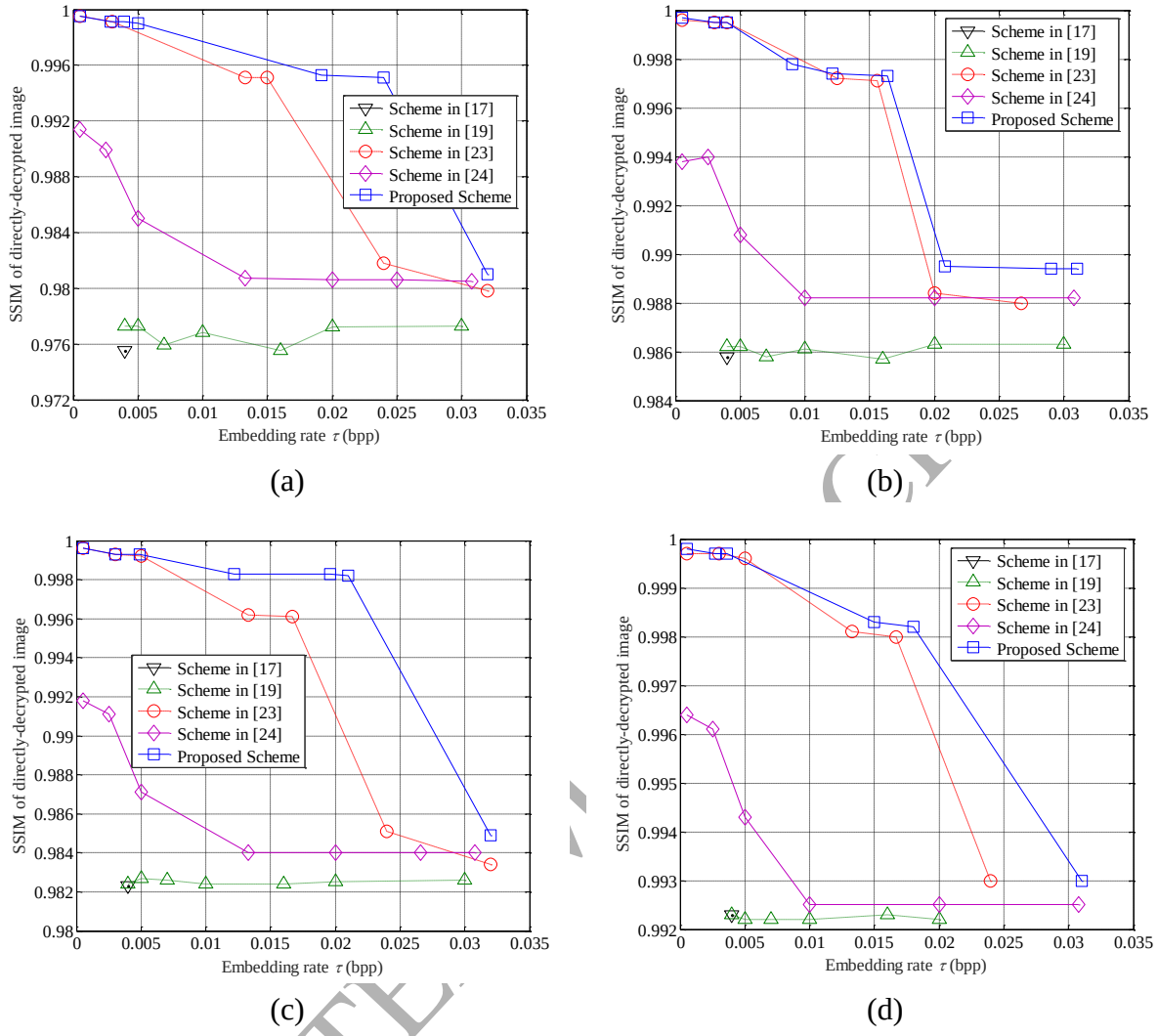
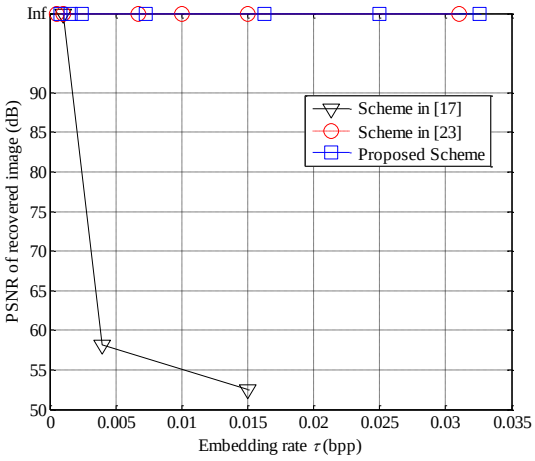
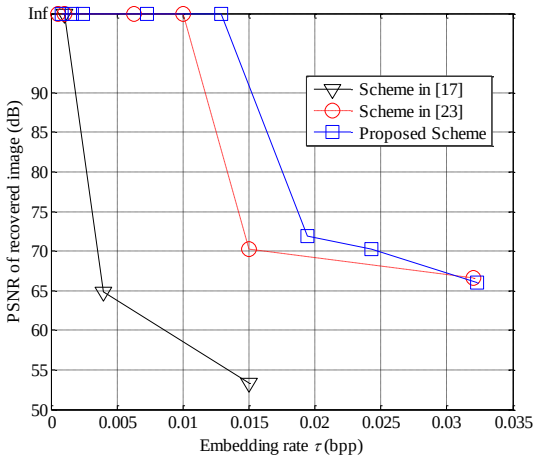


Figure 11 Rate-SSIM comparisons between the proposed scheme and [17, 19, 23, 24]. (a) *Airplane*, (b) *Man*, (c) *Lena*, (d) *Baboon*.

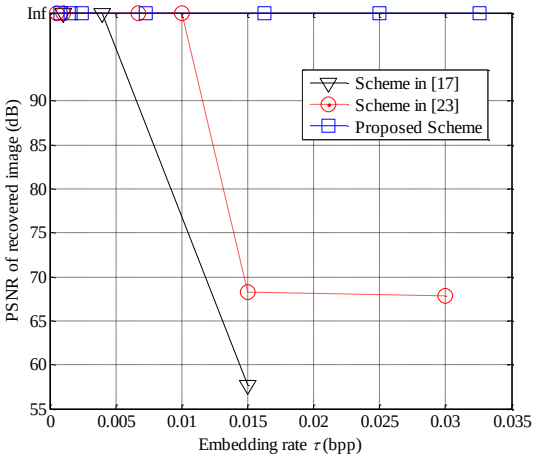
In addition, we also compared the recovery accuracy of our scheme with that of the VRAE based schemes [17, 23]. Figures 12-13 show the values of PSNR and SSIM for recovered images under different embedding rates, respectively. It can be observed from Figures 12-13 that, because the characteristic of spatial correlation for natural images is exploited during image recovery in the proposed scheme, the performance of recovery accuracy is better for the images with relatively smoother distribution, and due to the accurate prediction method, our scheme can generally achieve superior performance of recovery accuracy than the schemes [17, 23].



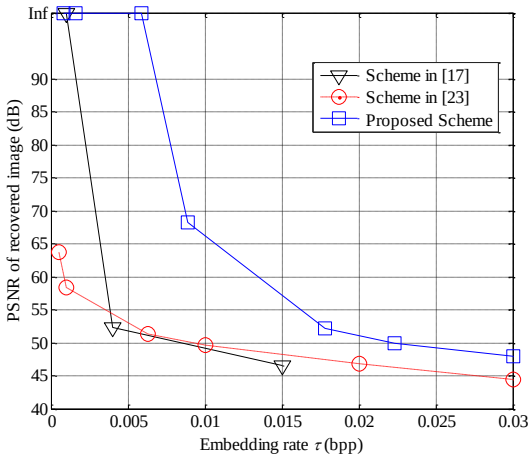
(a)



(b)

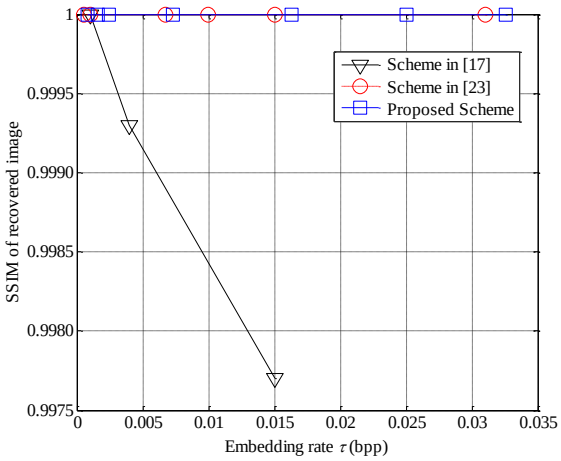


(c)

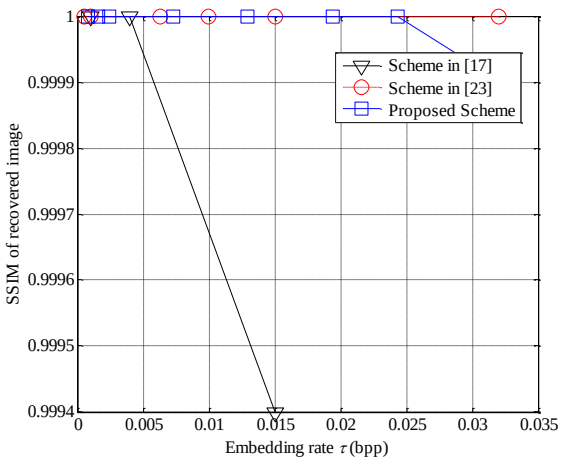


(d)

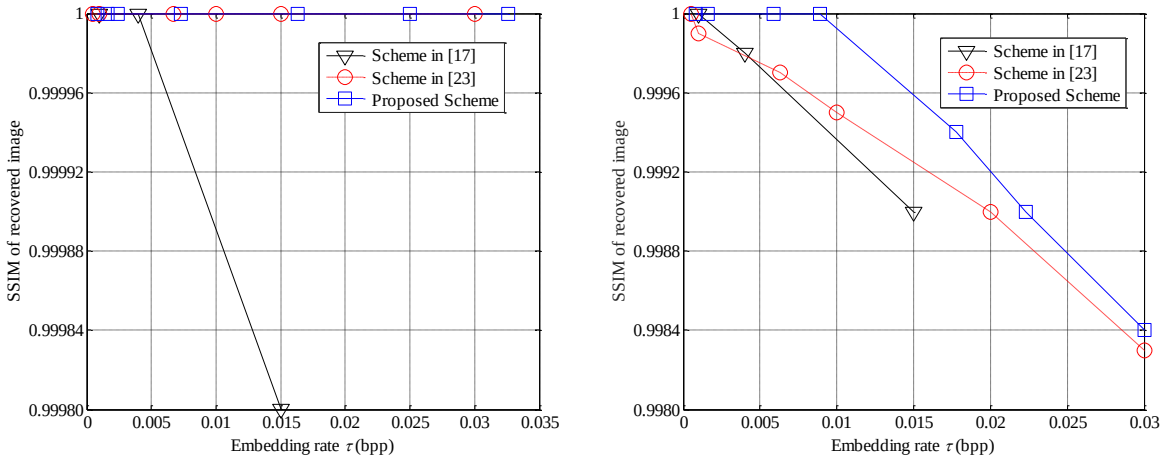
Figure 12 Comparisons of PSNR values for recovered image with respect to different embedding rate. (a) *Airplane*, (b) *Man*, (c) *Lena*, (d) *Baboon*.



(a)



(b)



(c) (d)

Figure 13 Comparisons of SSIM values for recovered image with respect to different embedding rate. (a) *Airplane*, (b) *Man*, (c) *Lena*, (d) *Baboon*.

4. Conclusions

In this work, a novel scheme for reversible data hiding in encrypted images is presented, which consists of image encryption, data embedding, data extraction and image recovery. In the stage of image encryption, all non-overlapping divided blocks in original image are encrypted by an analogous stream-cipher way and then conducted with a block permutation operation to protect the confidentiality of image contents. During the stage of data embedding, even though the data hider has no idea of the contents of original image, he/she can classify the blocks into two sets corresponding to the smooth and complex regions within original image. Through compressing the LSB layers of the block set corresponding to smooth region, spare space can be vacated to accommodate additional data to be embedded. The receiver can conduct separable operations of data extraction, direct decryption and image recovery according to the availability of encryption key and data-hiding key. Because the blocks corresponding to complex set are not modified during data embedding, thus, the visual quality of directly-decrypted image is significantly improved. On the other hand, more additional data can be embedded into the blocks belonging to smooth set, hence, the embedding rate is also acceptable. Through the elaborately-designed prediction strategy based on spatial correlation of natural images, perfect image recovery can be realized. Experimental results and comparisons demonstrate that the proposed scheme has better rate-distortion performance than some state-of-the-art schemes.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61672354), the Open Project Program of the National Laboratory of Pattern Recognition (201600003), the Open Project Program of Shenzhen Key Laboratory of Media Security, 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 suggestions.

References

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding — a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[2] C. D. Vleeschouwer, J. F. Delaigle, and B Macq, "Invisibility and application functionalities in perceptual watermarking: an overview," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 64–77, 2002.

[3] C. Qin, P. Ji, X. P. Zhang, J. Dong, and J. W. Wang, "Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy," *Signal Processing*, vol. 138, pp. 280–293, 2017.

[4] C. Qin, X. Q. Chen, D. P. Ye, and J. W. Wang, and X. M. Sun, "A novel image hashing scheme with perceptual robustness using block truncation coding," *Information Sciences*, vol. 361–362, pp. 84–99, 2016.

[5] C. Qin, C. C. Chang, and Y. P. Chiu, "A novel joint data-hiding and compression scheme based on SMVQ and image inpainting," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 969–978, 2014.

[6] Y. Q. Shi, X. L. Li, X. P. Zhang, H. T. Wu, and B. Ma, "Reversible data hiding: Advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.

[7] C. Qin, C. C. Chang, and T. J. Hsu, "Reversible data hiding scheme based on exploiting modification direction with two steganographic images," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5861–5872, 2015.

[8] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data

embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.

[9] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[10] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.

[11] B. Ou, X. L. Li, Y. Zhao, R. R. Ni, and Y. Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5010–5021, 2013.

[12] Z. C. Ni, Y. Q. Shi, N. Ansari and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[13] X. L. Li, B. Li, B. Yang, and T. Y. Zeng, "General framework to histogram-shifting-based reversible data hiding," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2181–2191, 2013.

[14] C. Qin, C. C. Chang, Y. H. Huang, and L. T. Liao, "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 7, pp. 1109–1118, 2013.

[15] X. L. Li, W. M. Zhang, X. L. Gui, and B. Yang, "Efficient reversible data hiding based on multiple histograms modification," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 2016–2027, 2015.

[16] W. Puech, M. Chaumont, and O. Strauss, "A Reversible Data Hiding Method for Encrypted Images," *Proceedings of SPIE 6819*, pp. 1–9, 2008.

[17] X. P. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[18] W. Hong, T. S. Chen, and H. Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[19] X. Liao and C. W. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication and Image Representation*, vol. 28, pp. 21–27, 2015.

[20] C. Qin and X. P. Zhang, "Effective reversible data hiding in encrypted image with

privacy protection for image content," *Journal of Visual Communication and Image Representation*, vol. 31, pp. 154–164, 2015.

[21] S. Agrawal and M. Kumar, "Mean value based reversible data hiding in encrypted images," *Optik - International Journal for Light and Electron Optics*, vol. 130, pp. 922–934, 2017.

[22] J. T. Zhou, W. W. Sun, L. Dong, X. M. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 441–452, 2016.

[23] X. P. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 526–532, 2012.

[24] Z. X. Qian, X. P. Zhang, and G. R. Feng, "Reversible data hiding in encrypted images based on progressive recovery," *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1672–1676, 2016.

[25] P. Puteaux, D. Trinel, and W. Puech, "High-capacity data hiding in encrypted images using MSB prediction," *Proceedings of the 2016 Sixth International Conference on Image Processing Theory, Tools and Applications*, Oulu, Finland, pp. 1–6, Dec. 12-15, 2016.

[26] P. Singh and B. Raman, "Reversible data hiding for rightful ownership assertion of images in encrypted domain over cloud," *AEU - International Journal of Electronics and Communications*, vol. 76, pp. 18–35, 2017.

[27] X. T. Wu and W. Sun, "High-capacity reversible data hiding in encrypted images by prediction error," *Signal Processing*, vol. 104, pp. 387–400, 2014.

[28] K. D. Ma, W. M. Zhang, X. F. Zhao, N. H. Yu, and F. H. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553–562, 2013.

[29] W. M. Zhang, K. D. Ma, and N. H. Yu, "Reversibility improved data hiding in encrypted images," *Signal Processing*, vol. 94, pp. 118–127, 2014.

[30] D. W. Xu and R. D. Wang, "Separable and error-free reversible data hiding in encrypted images," *Signal Processing*, vol. 123, pp. 9-21, 2016.

[31] X. C. Cao, L. Du, X. X. Wei, D. Meng, and X. J. Guo, "High capacity reversible data

hiding in encrypted images by patch-level sparse representation," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1132–1143, 2016.

[32] F. J. Huang, J. W. Huang, and Y. Q. Shi, "New framework for reversible data hiding in encrypted domain," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2777–2789, 2016.

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

[34] Z. X. Qian and X. P. Zhang, "Reversible data hiding in encrypted images with distributed source encoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 636–646, 2016.

[35] Y. C. Chen, C. W. Shiu and G. Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 1164–1170, 2014.

[36] X. T. Wu, B. Chen, and J. Weng, "Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 58–64, 2016.

[37] D. Xiao, Y. P. Xiang, H. Y. Zheng, and Y. Wang, "Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism," *Journal of Visual Communication and Image Representation*, vol. 45, pp. 1–10, 2017.

[38] G. Schaefer and M. Stich, "UCID – an uncompressed color image database," *Proceedings of SPIE in Storage and Retrieval Methods and Applications for Multimedia*, vol. 5307, pp. 472–480, 2004.

[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.