Optics and Lasers in Engineering 124 (2020) 105837

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

An efficient visually meaningful image compression and encryption scheme based on compressive sensing and dynamic LSB embedding

Xiuli Chai[a,*], Haiyang Wu[a], Zhihua Gan[b], Yushu Zhang[c,d,e], Yiran Chen[f], Kent W. Nixon[f]

[a] School of Computer and Information Engineering, Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China
[b] School of Software, Henan University, Kaifeng 475004, China
[c] School of Information Technology, Deakin University, Victoria 3125, Australia
[d] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[e] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China
[f] Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, United States

ARTICLE INFO

Key words:
Image compression and encryption
Compressive sensing
3-D Cat map

ABSTRACT

In this paper, an efficient visually meaningful image compression and encryption (VMICE) scheme is proposed by combining compressive sensing (CS) and Least Significant Bit (LSB) embedding. First, the original image (I_{orig}) is compressed and encrypted into a secret image (I_{sec}) by CS and Zigzag confusion. Next, dynamic LSB embedding is utilized to randomly embed I_{sec} into a separate carrier image (I_{car}) to create the final visually meaningful (VM) cipher image (I_{ciph}), which is the same size as I_{orig}. To generate the measurement matrix for CS and determine the embedding position of I_{sec} in I_{car}, a 3-D Cat map is employed. The utilization of this less-chaotic system makes our algorithm easy to implement. Moreover, the initial values of Cat map are computed using I_{orig}, making our image compression and encryption (ICE) algorithm robust to known-plaintext and chosen-plaintext attacks. Additionally, the adoption of LSB embedding allows for I_{car} to be selected independently of I_{orig}. Simulation results and performance analyses are presented to illustrate the effectiveness and efficiency of the proposed cryptosystem.

1. Introduction

With the rapid developments of cloud computing and big data technology, digital images are generated, transmitted, manipulated, and stored at ever increasing scales. We have attached great importance to securing the contents of these images, esp. w.r.t. malicious tampering and illegal distribution of intellectual property. While encryption is accepted as an effective technology to protect sensitive data [1,2], the nature of digital images (2-dimensional data which has features such as high redundancy, large size, and high correlation between adjacent pixels) make traditional AES and DES encryption methods inefficient [3]. Thus, dedicated image encryption (IE) schemes have been proposed with utilizing altogether different techniques, such as optical transformation [4,5], DNA sequence operations [6–8], cellular automata [9,10], quantum computation [11], and chaotic systems [12,13]. Of these, chaotic-system-based IE methods have attracted the most attention from scholars due to their computationally-efficient nature [14,15].

Chaotic-system-based IE algorithms may encrypt an original image (I_{orig}) into an encoded image (I_{enc}). A visually-similar representation of I_{orig} (I'_{orig}) can be extracted from I_{enc} with a set of decryption keys [16–18]. Unfortunately, IE algorithms require the resolution of decrypted image I'_{orig} and I_{enc} to remain equal to that of I_{orig}. This is undesirable because as the resolution of digital images continues to increase, it becomes necessary to both encrypt and compress images in order to reduce transmission bandwidth and storage space. To allow for this, compressive sensing (CS) has recently been proposed for use in newer image compression and encryption (ICE) schemes, allowing for simultaneous sampling, compression, and encryption of images [19–22]. An example of such an ICE algorithm, based on a chaotic system and CS, is presented in [23]. In that work, the authors utilize elementary cellular automata (ECA) to shuffle the sparse coefficient matrix of I_{orig} in order to improve reconstruction effect. Hu et al. [24] proposed a novel ICE scheme under a parallel CS framework, demonstrating that the efficiency of this algorithm is ensured. More recently, Chen et al. [25] presented an ICE scheme which was composed of two stages: 1) the CS phase and 2) the permutation-diffusion stage. However, this scheme results in a I_{ciph} comprised of noise-like (Fig. 1(a)) or texture-like (Fig. 1(b)) patterns. This immediately distinguishes a I_{ciph} from other natural images, revealing their nature as encryption tools and making them a prime target for attackers. Therefore, in order to better disguise the important nature of I_{ciph}, an ICE algorithm should be developed s.t. the resultant I_{ciph} are visually similar to natural images.

* Corresponding author.
E-mail address: chaixiuli@henu.edu.cn (X. Chai).

https://doi.org/10.1016/j.optlaseng.2019.105837
Received 23 January 2019; Received in revised form 14 August 2019; Accepted 15 August 2019
Available online xxx
0143-8166/

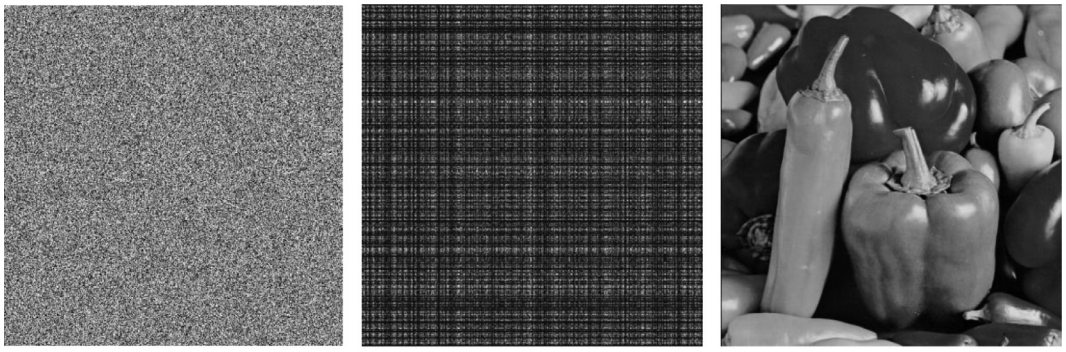

Fig. 1. Cipher images.

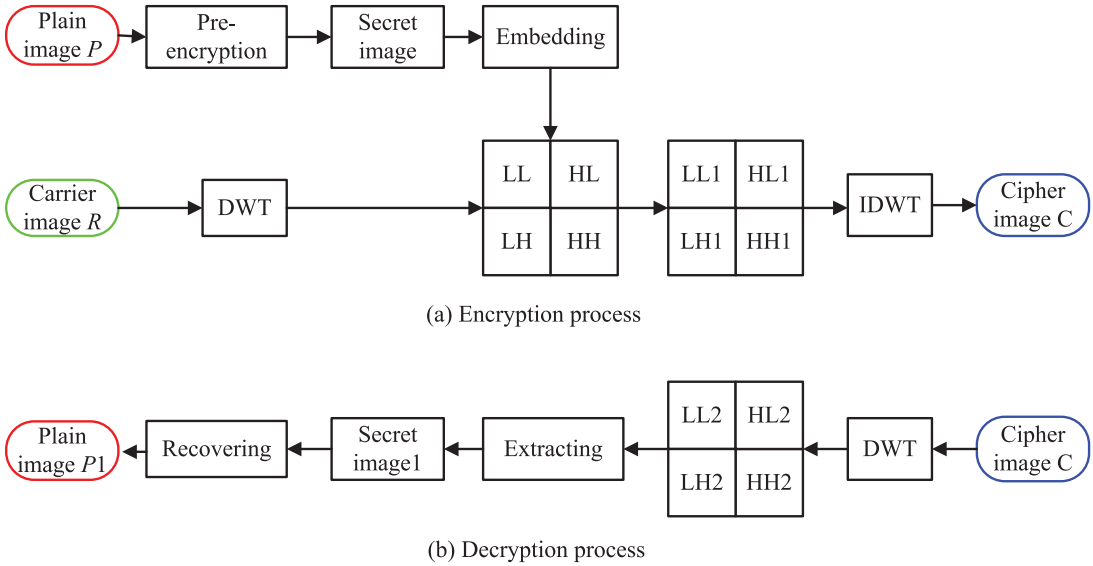

Fig. 2. The schematic of the encryption process and decryption process.

In 2015, Bao and Zhou [26] firstly proposed the concept of visually meaningful (VM) I_{ciph}, where a new IE method was also presented. This algorithm consisted of pre-encryption process and a discrete-wavelet-transform-based content transform (DWTCT). I_{orig} was encrypted into I_{sec} by an existing IE algorithm, and then DWTCT was performed on I_{sec} to get a VM I_{ciph} (such as the image in Fig. 1(c)). The obtained I_{ciph} was visually similar to a carrier image (I_{car}), making it easier to disguise as a natural image. In this algorithm, the size of I_{orig} is $M \times N$, while the I_{car}, I_{ciph} are all $2M \times 2N$- that is, the resolution is four times that of I_{orig}. From the perspective of computational complexity, transmission bandwidth, and storage space, the requirement of a higher resolution is inefficient and inconvenient for practical applications. This same problem also exists in multiple other studies [27–29].

In order to overcome this problem, our group examined production of VM I_{ciph} by use of CS [30]. To begin, I_{orig} is compressed by CS into I_{sec}, which is then embedded into the DWT coefficients of a I_{car}. The final, visually meaningful I_{ciph} has the same size as the I_{orig}. The meaningful appearance of the I_{ciph} results in the I_{orig} being more secure with no additional bandwidth requirements. The encryption algorithm based on DWT embedding is illustrated in Fig. 2. In the encryption process of these algorithms, the I_{orig} is first pre-encrypted into a I_{sec}. The I_{sec} is then embedded into the DWT coefficients (LL, HL, LH and HH) of a I_{car}, with a VM I_{ciph} obtained by performing IDWT on the modified coefficients (LL1, HL1, LH1 and HH1). For the decryption process, the four coefficient matrices are first obtained by performing DWT on the I_{ciph}, with the extracted secret image 1 (I'_{sec}) being extracted using them. Finally, the I'_{orig} is recovered using I'_{sec}. Since the DWT transforms an integer matrix into a decimal matrix, it is irreversible. That is to say, the matrix consisting of LL1, HL1, LH1 and HH1 is not exactly the same as the matrix made up of LL2, HL2, LH2 and HH2. Hence, I'_{sec} and I_{sec} are also different. Therefore, when these wavelet coefficients are used to design the embedding method of I_{sec}, the quality of the recovered image is affected by the content I_{car}. If the I_{orig} is not compressed in pre-encryption, I'_{orig} will almost always be visually-similar to I_{orig} due to the high level of redundancy in image data. However, if the I_{orig} is compressed before embedding, the influence of the I_{car} on I'_{orig} is amplified in proportion to the compression ratio. To remedy this shortcoming and improve the visual quality of I'_{orig}, a dynamic LSB embedding is utilized to replace DWT embedding in this paper.

Based on the above analyses, a novel, VMICE scheme combining CS and LSB embedding is detailed. This algorithm is made up of two stages: 1) pre-encryption and 2) embedding. In pre-encryption, CS and Zigzag confusion are utilized to compress and encrypt the I_{orig} to get a texture-like I_{sec}. In the second stage, I_{sec} is embedded into a I_{car} by a dynamic LSB embedding. Finally, a VM I_{ciph} is obtained which is of the same resolution as I_{orig} and is visually-similar to I_{car}. The process of embedding the elements of I_{sec} is controlled by index arrays generated by sorting chaotic sequences from the 3-D Cat map, with the initial values and parameters of the Cat map calculated using the content of I_{orig}. Thus, each I_{orig} will utilize a different embedding method. Additionally,

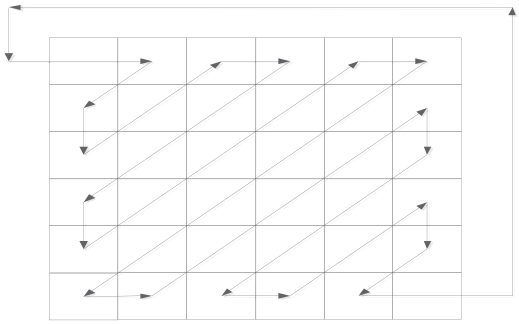

Fig. 3. The zigzag path.

10	25	16	127	105	56
46	89	102	62	45	54
38	111	106	181	252	274
122	124	192	172	93	83
118	75	2	9	11	34
21	91	30	106	52	128

111	122	118	124	106	62
105	56	45	181	192	75
21	91	2	172	252	54
274	93	9	30	106	11
83	34	52	128	10	25
46	38	89	16	127	102

Fig. 4. Zigzag confusion result with the starting position (3, 3).

these chaotic sequences are also used for designing the measurement matrix of CS, which is regarded as the secret key. Since the generation of chaotic sequences is sensitive to the content of I_{orig}, different I_{orig} also results in different measurement matrices, improving the security of pre-encryption. Additionally, reversible LSB embedding makes I'_{sec} independent of the content of I_{car}, enhancing the potential key space of the proposed algorithm.

The rest of this paper is organized as follows: Preliminaries of the work include CS and Zigzag confusion, which are described in Section 2. In Section 3, the proposed VMICE algorithm is presented. Simulation results and performance analyses are provided in Section 4, with Section 5 drawing conclusions from the work.

2. Preliminaries

2.1. Compressive sensing

CS says [31] that one may recover a length-N signal $\boldsymbol{x}$ from its linear measurement $\mathbf{y} = \boldsymbol{\Phi} x$, when signal $\boldsymbol{x}$ is naturally sparse or can be expressed by $\mathbf{x} = \boldsymbol{\psi} s$, and the exact solution can be realized. And here, $\boldsymbol{\Phi}$ is a measurement matrix sized of $M \times N$, $\mathbf{y}$ is a measurement vector sized of $M \times 1$, and $M \ll N$, $\boldsymbol{\psi}$ is the transformation matrix (such as DCT, DWT, FFT), and $\mathbf{s} = \{s_1, s_2, \cdots, s_N\}$ is a $N \times 1$ coefficient vector.

In this sense, the measurement process can be described by:

$$\mathbf{y} = \boldsymbol{\Phi} x = \boldsymbol{\Phi}\psi s = F\mathbf{s} \tag{1}$$

Here, $\boldsymbol{F}$ is the $M \times N$ sensing matrix.

The theory of CS also relies on the signal $\boldsymbol{x}$ can be reconstructed with overwhelming probability from $\boldsymbol{y}$ if the matrix $\boldsymbol{F}$ satisfy restricted isometry property (RIP) [32]. $\boldsymbol{x}$ can be obtained by solving the following convex optimization problem:

$$\min \|\mathbf{s}\|_1 \quad \text{s.t. } \mathbf{y} = \boldsymbol{\Phi}\psi s \tag{2}$$

where $\|\mathbf{s}\|_1$ represents the l_1-norm of vector $\boldsymbol{s}$. Matching pursuit (MP), orthogonal matching pursuit (OMP), smoothed l_0 norm (SL_0) are all effective algorithms to reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}$.

In this paper, CS is used to compress and encrypt I_{orig}, the measurement vectors can be regarded as the I_{ciph}, and measurement matrix is the secret key.

2.2. Zigzag confusion

Zigzag confusion [30] is utilized to shuffle sparse coefficient matrix of I_{orig}. By doing so, high correlation between adjacent sparse coefficients has been reduced, and the reconstruction effect of I'_{orig} can be largely improved. In the operation, the starting element is first selected. Subsequent elements are then traversed by a zigzag pattern, as shown in Fig. 3. Depending on the position of the starting element, the confusion matrix is also different. For example, for a matrix sized of 6×6, if the starting position is (3, 3), the confusion matrix starts from the element 111. The matrices before and after Zigzag confusion are illustrated in Fig. 4. In this paper, the starting position is computed using the contents of I_{orig}. Thus, there are different confusion matrix for different I_{orig}.

3. The proposed encryption scheme

3.1. Generation of some important parameters

3.1.1. Computing starting position for Zigzag confusion

The contents of I_{orig} are utilized to generate the starting position (x_0, y_0) for Zigzag confusion. The I_{orig} P is assumed to be an $m \times n$ sized image. First, the sum su of all pixels and the average value I are given as follows:

$$su = \sum_{i=1}^{m}\sum_{j=1}^{n} P(i,j) \tag{3}$$

$$I = \frac{su}{m \times n} \tag{4}$$

where $P(i, j)$ is the pixel of I_{orig} P located in the i-th row and j-th column.

Next, two parameters k_1 and k_2 are determined as follows:

$$k_i = \frac{t_i + I}{m \times n}, \quad i = 1, 2 \tag{5}$$

where t_1 and t_2 are two external key parameters.

Finally, the starting position (x_0, y_0) is computed according to the following equations:

$$\begin{cases} x_0 = \left\lfloor \left(\left(abs(k_1) - \lfloor k_1 \rfloor \right) \times 10^{14} \right) \bmod m \right\rfloor + 1 \\ y_0 = \left\lfloor \left(\left(abs(k_2) - \lfloor k_2 \rfloor \right) \times 10^{14} \right) \bmod n \right\rfloor + 1 \end{cases} \tag{6}$$

where $abs(x)$ is the absolute value of x, mod is the modular operator, and $\lfloor x \rfloor$ means getting the largest integer no more than x. For example, $\lfloor 2.7 \rfloor = 2$.

3.1.2. Obtaining the initial values and parameters of 3-D Cat map

In this paper, a 3-D Cat map is used to produce measurement matrix for CS and image permutation. Compared with 2-D Cat map, 3-D Cat map has higher randomness and it is defined as follows [33]:

$$\begin{pmatrix} X_{n+1} \\ Y_{n+1} \\ Z_{n+1} \end{pmatrix} = \begin{pmatrix} 1 + a_x a_z b_y & a_z & a_y + a_x a_z (a_y b_y + 1) \\ b_z + a_x b_y (a_z b_z + 1) & a_z b_z + 1 & a_y a_z + a_x (a_y b_y + 1)(a_z b_z + 1) \\ b_y (a_x b_x + 1) & b_x & (a_x b_x + 1)(a_y b_y + 1) \end{pmatrix} \times \begin{pmatrix} X_n \\ Y_n \\ Z_n \end{pmatrix} \bmod 1 \tag{7}$$

where X_0, Y_0, Z_0 are the initial values, and a_x, a_y, a_z, b_x, b_y, b_z are system parameters.

Logistic map is performed to calculate the initial values and parameters of 3-D Cat map. It is described as follows:

$$r_{n+1} = \mu \times r_n (1 - r_n), \quad n = 0, 1, 2, 3..... \tag{8}$$

where system parameter $\mu \in (0, 4]$, and r_n is a floating point number in range (0, 1). When $3.5699\ldots < \mu \leq 4$, this system exhibits chaotic features.

The generation process of initial values and system parameters of 3-D Cat map are presented as follows.

Step 1: Parameter Q is calculated by using the average value I.

$$Q = floor\left(\left((I - \lfloor I \rfloor) \times 10^{12}\right) \bmod 255\right) + 1 \tag{9}$$

Step 2: Logistic map is iterated L times under the condition that the system parameter is μ and the initial value is r_0, and a chaotic sequence $\{r_1, r_2, \ldots, r_L\}$ is obtained.

Step 3: Initial parameters of 3-D Cat map are computed according to the following approach:

$$\begin{cases} a_x = \lceil (r_{450+Q} \times 10^{14}) \bmod 1 \rceil \\ a_y = \lceil (r_{550+Q} \times 10^{14}) \bmod 1 \rceil \\ a_z = \lceil (r_{650+Q} \times 10^{14}) \bmod 1 \rceil \\ b_x = \lceil (r_{750+Q} \times 10^{14}) \bmod 1 \rceil \\ b_y = \lceil (r_{850+Q} \times 10^{14}) \bmod 1 \rceil \\ b_z = \lceil (r_{950+Q} \times 10^{14}) \bmod 1 \rceil \\ X_0 = (r_{1050+Q} \times 10^{14}) \bmod m \\ Y_0 = (r_{1150+Q} \times 10^{14}) \bmod m \\ Z_0 = (r_{1250+Q} \times 10^{14}) \bmod m \end{cases} \tag{10}$$

where m is the row number of I_{orig} P, and $\lceil x \rceil$ rounds the elements of x to the nearest integers greater than or equal to x. For example, $\lceil 2.7 \rceil = 3$.

As Eq. (9) shows, parameter Q is in the range of 1 and 256. Hence, parameter L is chosen as 2000 to improve the execution efficiency and avoid more iteration operations of computing chaotic map.

3.2. Generating the measurement matrix

In the ICE scheme based on CS, the measurement matrix is the key and is transmitted to the receiver for decryption. In order to reduce transmission bandwidth and storage burden, 3-D Cat map is used to generate the measurement matrix. Thus, fewer controlling parameters produce larger matrix. The detailed generation process is as follows:

Step 1: 3-D Cat map is iterated $L' = 1000 + MNd$ times with the initial values X_0, Y_0, and Z_0. By abandoning the former 1000 values, three random chaotic sequences X, Y, Z sized of $1 \times MNd$ are obtained and shown as follows:

$$\begin{cases} X = \{x_1, x_2, \ldots, x_{MNd}\} \\ Y = \{y_1, y_2, \ldots, y_{MNd}\} \\ Z = \{z_1, z_2, \ldots, z_{MNd}\} \end{cases} \tag{11}$$

where the parameter d is the sampling distance for chaotic sequences, and CR is the compression ratio of I_{orig}, and $M = CR \times m$, $N = m$.

Step 2: According to the following equation, a new random sequence $W = \{w_1, w_2, \ldots, w_{MNd}\}$ is determined by modifying X, Y, Z.

$$W_i = \frac{X_i + Y_i + Z_i}{3}, \quad i = 1, 2, \ldots, MNd \tag{12}$$

where X_i, Y_i, Z_i and W_i are the i-th element of chaotic sequence X, Y, Z and W, respectively.

Step 3: Sequence W' is obtained by sampling sequence W with interval d, and it is defined as follows:

$$W'_k = W_{1+kd}, \quad k = 0, 1, 2, \ldots, MN - 1 \tag{13}$$

where d is the sampling interval.

Step 4: In order to make the sequence more random, sequence W' is processed by using the following equation to obtain a new sequence W'':

$$W''_k = 1 - 2W'_k, \quad k = 0, 1, 2\ldots\ldots, MN - 1 \tag{14}$$

Step 5: The obtained sequence W'' is reordered in a column-wise manner, and then the measurement matrix $\phi_{M \times N}$ is constructed as follows:

$$\phi = \sqrt{\frac{2}{M}} \begin{pmatrix} W''_{(0)} & \cdots & W''_{(M(N-1))} \\ W''_{(1)} & \cdots & W''_{(M(N-1)+1)} \\ \vdots & \vdots & \vdots \\ W''_{(M-1)} & \cdots & W''_{(MN-1)} \end{pmatrix} \tag{15}$$

3.3. The VMICE algorithm

The proposed VMICE algorithm is illustrated in Fig. 5. From this figure, it can be seen that it is composed of two stages. In the first stage, CS and Zigzag confusion are combined to compress and encrypt I_{orig} into I_{sec}, with the texture-like I_{sec} protecting the information security of I_{orig}. In the second stage, I_{sec} is randomly embedded into I_{car} to obtain a VM I_{ciph}, with chaotic sequences generated from the 3-D Cat map being used to control the embedding operation.

3.3.1. Getting I_{sec} by using CS and Zigzag confusion

Step 1: After sparsifying I_{orig} P of resolution $m \times n$ by use of discrete wavelet transform (DWT), the sparse coefficient matrix P_1 is determined.

Step 2: As described in Section 3.1, the starting position (x_0, y_0) for Zigzag confusion is computed with the given secret key t_1 and t_2. Zigzag confusion is then applied on matrix P_1 to get a matrix P'_1. In order to increase the construction effect, the elements of matrix P'_1 are modified by changing the values less than threshold TS to 0, resulting in matrix P_2.

Step 3: As illustrated in Sections 3.1.2 and 3.2, the initial values and parameters of 3-D Cat map are produced with the secret key μ and r_0, with the measurement matrix $\phi(M \times N)$ for CS then being computed.

Step 4: By measuring matrix P_2 by use of ϕ, measurement value matrix P_3 is given. It is assumed that the compression ratio of I_{orig} P is CR. Then, the size of P_3 is $M \times n$.

Step 5: I_{sec} P_4 is determined by quantizing matrix P_3 into the range of [0, 255]. This operation is formulated as follows:

$$P_{4(i)} = floor\left(\frac{255 \times \left(P_{3(i)} - \min\right)}{\max - \min}\right) \tag{16}$$

where min and max are the minimum and maximum values of matrix P_3, and $1 \leq i \leq Mn$.

3.3.2. Embedding I_{sec} into I_{car}

Step 1: Let the size of I_{car} R be $m \times n$. Preprocess the pixel values of I_{car} R, and adjust intensity levels to the range of [10, 245] according to the following equation:

$$R' = \left\lfloor A + \frac{\mu - A}{255} R \right\rfloor \tag{17}$$

where $A = 10$ and $\mu = 245$. One refers to the resulted image as R'.

Step 2: The I_{car} is divided into four equal parts and matrices C_A, C_H, C_V and C_D are obtained, with their size each being $(m/2) \times (n/2)$. For compression ratio $CR = 0.25$, the size of I_{sec} P_4 is $(m/4) \times n$.

Step 3: The matrices C_A, C_H, C_V and C_D are transformed into array C_1, C_2, C_3 and C_4, and their length is $u = (m \times n)/4$.

Step 4: After picking out the former u values from the chaotic sequences X, Y, Z, W generated by 3-D Cat map in Section 3.2, four sequences are obtained and presented as $X_1 = (x_1, x_2, \ldots, x_u)$, $Y_1 = (y_1, y_2, \ldots, y_u)$, $Z_1 = (z_1, z_2, \ldots, z_u)$, and $W_1 = (w_1, w_2, \ldots, w_u)$.

Step 5: By sorting sequences X_1, Y_1, Z_1, and W_1 in ascending order, the four index arrays IA_1, IA_2, IA_3, and IA_4 are obtained.

Step 6: After scrambling and reordering vectors C_1, C_2, C_3 and C_4 with index arrays IA_1, IA_2, IA_3 and IA_4, respectively, four arrays are

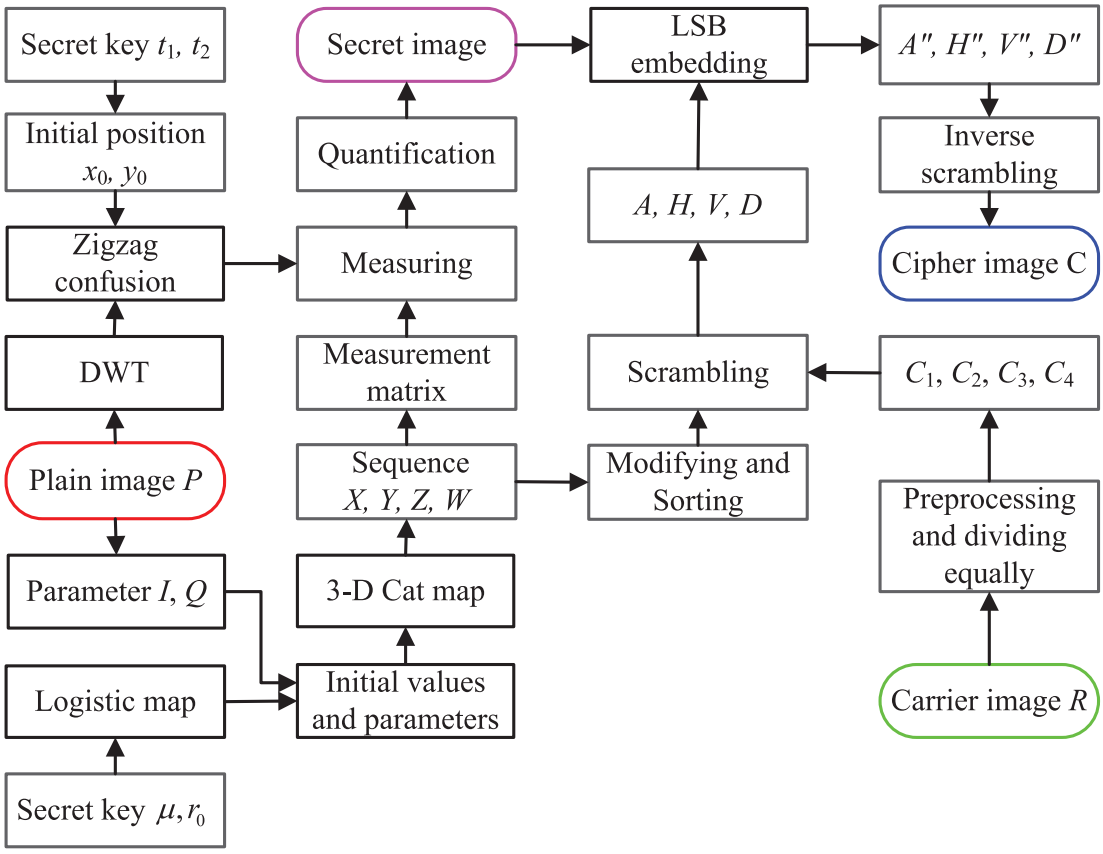

Fig. 5. The schematic of the proposed VMICE algorithm.

obtained and they are $A=(a_1, a_2, \ldots, a_u)$, $H=(h_1, h_2, \ldots, h_u)$, $V=(v_1, v_2, \ldots, v_u)$, and $D=(d_1, d_2, \ldots, d_u)$.

Step 7: I_{sec} P_4 is stretched to one dimensional array $F'=(f'_1, f'_2, \ldots, f'_u)$, and is then embedded into the arrays A, H, V, D via LSB embedding. The detailed process is as follows: First, the j-th element of array F' is represented in binary as $b_8b_7b_6b_5b_4b_3b_2b_1$ (and here b_8 is the highest bit, and b_1 is the lowest bit), and the j-th elements a_j, h_j, v_j, d_j of arrays A, H, V and D are also changed to their binary formats. Subsequently, b_1b_2, b_3b_4, b_5b_6, and b_7b_8 are embedded into the lowest two bits of a_j, h_j, v_j, and d_j, respectively. By keeping other bits constant, a'_j, h'_j, v'_j, and d'_j are obtained. After transforming the binary values to decimal values, four arrays A', H', V' and D' are obtained and represented as $A'=(a'_1, a'_2, \ldots, a'_u)$, $H'=(h'_1, h'_2, \ldots, h'_u)$, $V'=(v'_1, v'_2, \ldots, v'_u)$, and $D'=(d'_1, d'_2, \ldots, d'_u)$.

Step 8: By scrambling A', H', V', and D' with index arrays IA_1, IA_2, IA_3, and IA_4 according to the reverse order of Step 6, A'', H'', V'', and D'' are determined. The four matrices C'_A, C'_H, C'_V, and C'_D are then given by modifying them.

Step 9: The VM I_{ciph} C is finally obtained by recombining C'_A, C'_H, C'_V, and C'_D into one matrix. The encryption process is finished.

In the proposed VMICE algorithm, the resolution of I_{car} and I_{orig} are equal. To embed I_{sec} into I_{car} and save network bandwidth and storage space, the compression ratio is fixed as $CR=0.25$. When I_{car} is of higher resolution than I_{orig}, the compression ratio can be adjusted accordingly. In addition, in order to obtain high-quality reconstruction in I'_{orig} from a sparsely sampled dataset, the sparse coefficient matrix of I_{orig} is confused. In particular, when elements of this coefficient matrix are less than threshold TS, they are set as 0, and those greater than TS are constant. Hence, I_{orig} is sparser. TS is a positive number, and its value affects the sparsity level of images. Also, the visual effect of reconstructed images changes.

3.4. The image decryption algorithm

The image decryption algorithm is the inverse operation of the VMICE method, and is illustrated in Fig. 6. It consists of two phases: first, I'_{sec} is extracted from I_{ciph}; second, the I'_{orig} is reconstructed. The adoption of LSB embedding makes I'_{sec} the same as the compressed I_{sec} (see Section 3.3.1.). To facilitate decryption, some secret keys must be transmitted to the receiver, i.e. t_1, t_2, CR, I, μ, r_0 and max, min. The detailed decryption process is as follows:

3.4.1. Extracting I'_{sec} from the VM I_{ciph}

Step 1: As described in Sections 3.1 and 3.2, compute the starting position (x_0, y_0) for Zigzag confusion and initial values and parameters of 3-D Cat map.

Step 2: 3-D Cat map is iterated to get four chaotic sequences X, Y, Z, W. Then, by choosing the former u values from these sequences, one may get sequences X_1, Y_1, Z_1, W_1, and $u=(m\times n)/4$. Four index arrays IA_1, IA_2, IA_3, and IA_4 are obtained by sorting them in ascending order.

Step 3: The VM I_{ciph} C $(m\times n)$ is divided into four equal parts, and they are C'_A, C'_H, C'_V, and C'_D sized of $(m/2)\times(n/2)$.

Step 4: Resize C'_A, C'_H, C'_V, and C'_D into one-dimensional arrays A'', H'', V'', and D'', and then order them by index arrays IA_1, IA_2, IA_3, and IA_4, respectively. The obtained arrays are denoted as $A'=(a'_1, a'_2, \ldots, a'_u)$, $H'=(h'_1, h'_2, \ldots, h'_u)$, $V'=(v'_1, v'_2, \ldots, v'_u)$, and $D'=(d'_1, d'_2, \ldots, d'_u)$.

Step 5: Transform the j-th elements of arrays A', H', V', D' into their binary format. By picking up the lowest two bits from them, $b_8b_7b_6b_5b_4b_3b_2b_1$ is obtained, with element f'_j obtained by the following equation:

$$f'_j = b_8\times 2^7 + b_7\times 2^6 + b_6\times 2^5 + b_5\times 2^4 + b_4\times 2^3 + b_3\times 2^2 + b_2 \times 2^1 + b_1\times 2^0 \quad (18)$$

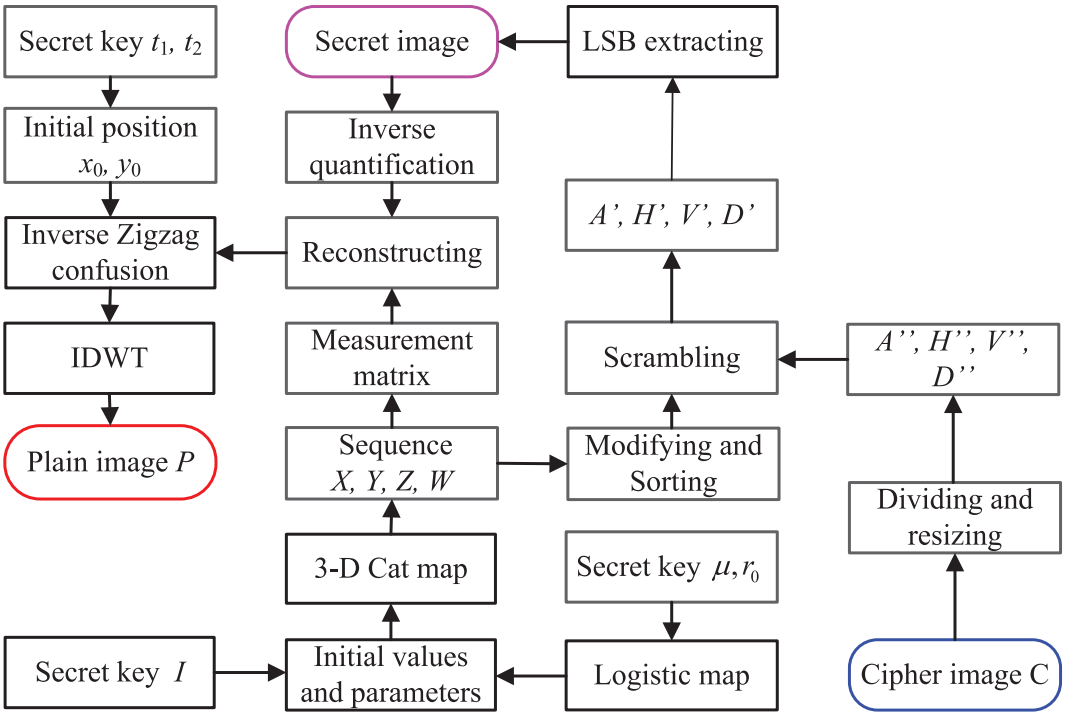

Fig. 6. The schematic of the proposed image decryption algorithm.

When all the elements are obtained, we have array $F'=(f'_1, f'_2, \ldots, f'_u)$ and $u=(m\times n)/4$.

Step 6: By converting array F' to a matrix, the I'_{sec} P_4 of size $(m/2)\times(n/2)$ is obtained.

3.4.2. Reconstructing I'_{orig}

Step 1: First, an inverse quantification operation is performed on I'_{sec} P_4, and the matrix P_3 is obtained. This process is formulated as follows:

$$P_{3(i)} = \frac{P_{4(i)} \times (max - min)}{255} + min,\ 1 \le i \le Mn \tag{19}$$

Step 2: The measurement matrix $\phi(M\times N)$ is generated as described in Section 3.2, with the orthogonal matching pursuit (OMP) algorithm then utilized to recover coefficient matrix P_2 from P_3. This process can be denoted as follows:

$$P_2 = \mathrm{OMP}(P_3, \phi) \tag{20}$$

Step 3: Inverse Zigzag confusion (IZC) is applied to matrix P_2 with the starting position (x_0, y_0), returning matrix P_1. By performing inverse discrete wavelet transform (IDWT) on P_1, the I'_{orig} P of size $m\times n$ is reconstructed. These operations can be denoted as follows:

$$P = \mathrm{IDWT}(\mathrm{IZC}(P_2, x_0, y_0)) \tag{21}$$

3.5. Discussion

First, we present a novel scheme to generate VM I_{ciph}. I_{orig} is compressed and encrypted into a I_{sec} by use of CS and Zigzag confusion, with a VM I_{ciph} obtained by embedding I_{sec} into a I_{car} via LSB embedding. During the transition and storage of the VM I_{ciph}, it has a lower likelihood of attracting the attention of hackers. Therefore, its security is greatly enhanced.

Moreover, content security of I_{orig} is guaranteed. In the proposed VMICE algorithm, the sparse coefficient matrix is shuffled by Zigzag confusion, and then the confusion matrix is compressed and encrypted into I_{sec} by CS. The measurement matrix of CS is controlled by 3-D Cat map. Additionally, in the embedding process, I_{car} is divided into four parts which are then ordered by the index arrays produced by chaotic sequences. Subsequently, I_{sec} is embedded into the lowest bits of I_{car} pixels by dynamic LSB embedding. The high randomness of the chaotic sequence determines the randomness of the embedded results, guaranteeing that I_{orig} is protected effectively.

Additionally, our VMICE algorithm has high sensitivity to I_{orig}. The average value I of I_{orig} is used to generate the parameter Q, with the parameters x_0, y_0, a_x, a_y, a_z, b_x, b_y, b_z, X_0, Y_0, and Z_0 also being obtained from parameter I. x_0, y_0 are the starting position for Zigzag confusion, and a_x, a_y, a_z, b_x, b_y, b_z, X_0, Y_0, Z_0 are the system parameters and initial values of the 3-D Cat map. Chaotic consequences generated by Cat map are not only used to construct the measurement matrix of CS, but also to determine the method of embedding I_{sec} to attain dynamic LSB embedding. Parameter Q will vary for different I_{orig}, causing the initial values for Zigzag confusion and 3-D Cat map to also vary with image contents, making I_{sec} and I_{ciph} different. In other words, even if the secret key is constant, there are different encryption results for different test images. That makes our proposed VMICE algorithm resistant against known-plaintext and chosen-plaintext attacks.

Additionally, in the proposed encryption scheme, CS is used to compress and encrypt I_{orig} into I_{sec}. When the compression ratio is 0.25, I_{ciph} has the same size as I_{orig}. That means no extra storage space or transmission bandwidth is required. In the embedding process, dynamic LSB embedding is also adopted. The lowest two bits of I_{car} pixels are replaced by I_{sec}. Also, the embedding positions of I_{sec} are controlled by the chaotic sequences generated from the 3-D Cat map. The lowest two bits of the image pixel are the unimportant bits, and they contain very little information. Dynamic LSB embedding ensures that 1) the distortion of I_{car} is very small, 2) all I_{sec} information can be embedded completely, and 3) the embedding process is random. Therefore, the final I_{ciph} is nearly identical to I_{car}, and is almost impossible to distinguish with the naked eye. Despite this, LSB embedding and LSB extracting are completely reversible. In the decryption process, when LSB extracting is manipulated, the I'_{sec} is fully recovered from I_{ciph}, and is identical to I_{sec}. That is, the quality of reconstructed image is independent of I_{car}, and it is determined only by the reconstruction algorithm and compression ratio. Therefore, we may select any I_{car} freely.

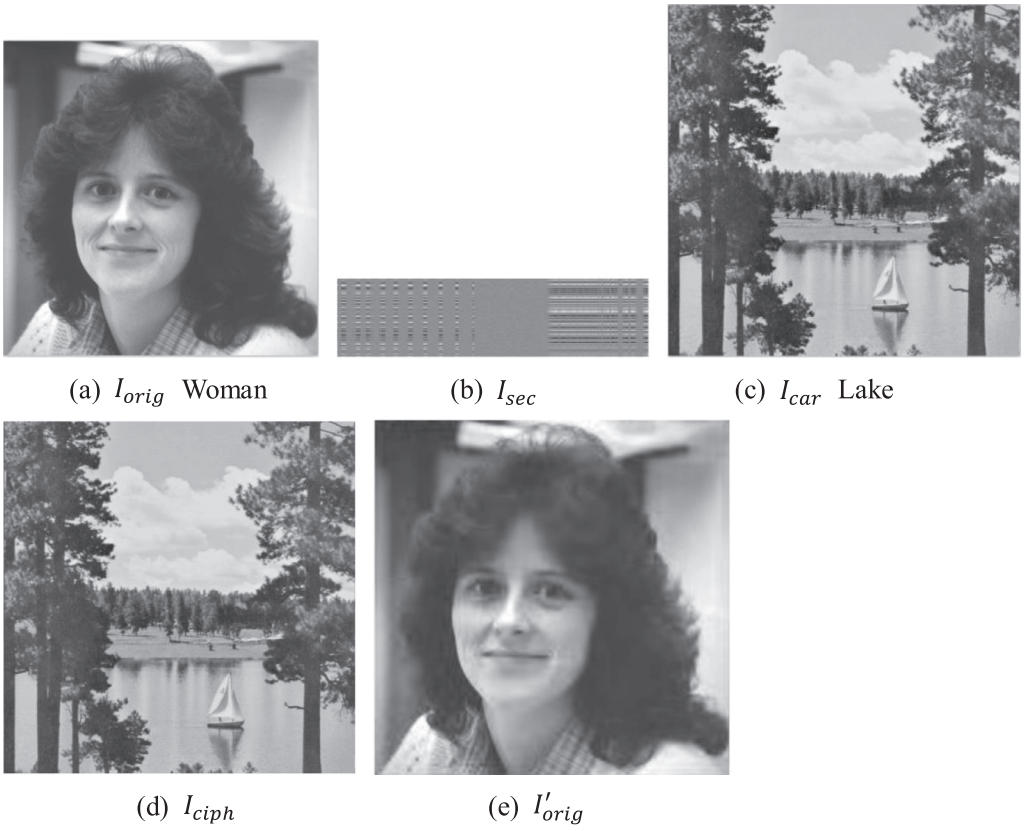

Fig. 7. Simulation results with I_{orig} as Woman (resolution 512 × 512).

4. Simulation results and performance analyses

In this section, simulation results of our VMICE algorithm will be given, and performance analyses are assessed from the perspective of visual effect, key space, key sensitivity, robustness, time complexity, known-plaintext and chosen-plaintext attacks, and comparison analysis. All experiments are conducted on a personal computer with 2.5 GHz CPU and 4GB memory, and the operating system is Microsoft Windows 10. Matlab R2016a is used as the coding tool. The secret keys are randomly set as: $\mu = 3.81$, $r_0 = 0.32$, $t_1 = 1.3842$, $t_2 = 0.0325$, $d = 25$, $CR = 0.25$, $TS = 50$. DWT is the sparsification method of I_{orig}, and OMP is used as the construction algorithm of CS.

4.1. Encryption and decryption results

In this subsection, Woman (Fig. 7(a)) is used as the test image and Lake (Fig. 7(c)) is the I_{car}, with both having a resolution of 512×512. In Fig. 7, (a) is I_{orig}, (b) is I_{sec}, (c) is I_{car}, (d) is I_{ciph} and (f) is I'_{orig}.

From Fig. 7, it can be seen that the size of I_{sec} shown in (b) is one-fourth that of the I_{orig} presented in (a), proving that the I_{orig} is compressed effectively. The obtained I_{sec} is texture-like, proving that we cannot obtain any useful information from it. Additionally, the I_{ciph} displayed in (d) is meaningful and appears identical to the corresponding I_{car} (c). When it is transmitted or stored among other natural images, the attackers are not obviously aware of it, making it more secure. Finally, I'_{orig} shown in (f) is visually similar to its respective I_{orig} presented in (a). The above results imply that our algorithm has good encryption and decryption effects. It may compress and encrypt I_{orig} successfully, and the meaningful I_{ciph} has higher security level than the noise-like or texture-like I_{ciph}.

4.2. Visual quality analysis

In the proposed encryption scheme, some parameters have important impact on the encryption and reconstruction results, and we will analyze them in the following two subsections.

4.2.1. Influence of different sparse and reconstruction methods on encryption and decryption results

First, one may analyze the influence of sparse methods and reconstruction methods of the I_{orig}. The test images are Lena (256×256) and Woman (512×512), the I_{car}s are Barbara (256×256) and Cameraman (512×512), the sparse methods are DWT and DCT2, and the reconstruction methods are OMP and SL_0. Fig. 8 illustrates the simulation results for Woman.

In Fig. 8, (a) is the I_{orig}, (b) is the I_{car}, (c) and (d) are the respective I_{sec} with DCT2 and DWT, (e) and (f) are the respective I_{ciph} with DCT2 and DWT. (g)–(j) are the respective histograms of (a), (b), (e) and (f). (k) and (l) are the reconstructed images of (f) with SL_0 and OMP, respectively. (m) and (n) are the respective reconstructed images of (e) with SL_0 and OMP, and (o)–(r) are the respective histograms of (k)-(n).

As observed from Fig. 8, when the sparse methods are DCT2 or DWT, the corresponding I_{sec} are meaningless, and I_{ciph} shown in (e) and (f) have the similar histogram distribution with I_{car} shown in (b). Secondly, it can be seen that among the histograms of reconstructed images, (o) and (p) are the closest to that of the I_{orig} shown in (g), which means that the quality of reconstructed images with DWT is higher than those with DCT2. Therefore, the adoption of DWT as the sparse method is more helpful for decryption.

For I_{orig} Lena (256×256) and Woman (512×512), we also analyze the relationship of reconstructed effect and threshold TS. The results are plotted in Fig. 9 and Fig. 10. As shown in these two figures, the maximum PSNR values between the reconstructed image and I_{orig} will be obtained with $TS = 50$ for Lena (256×256) and $TS = 30$ for

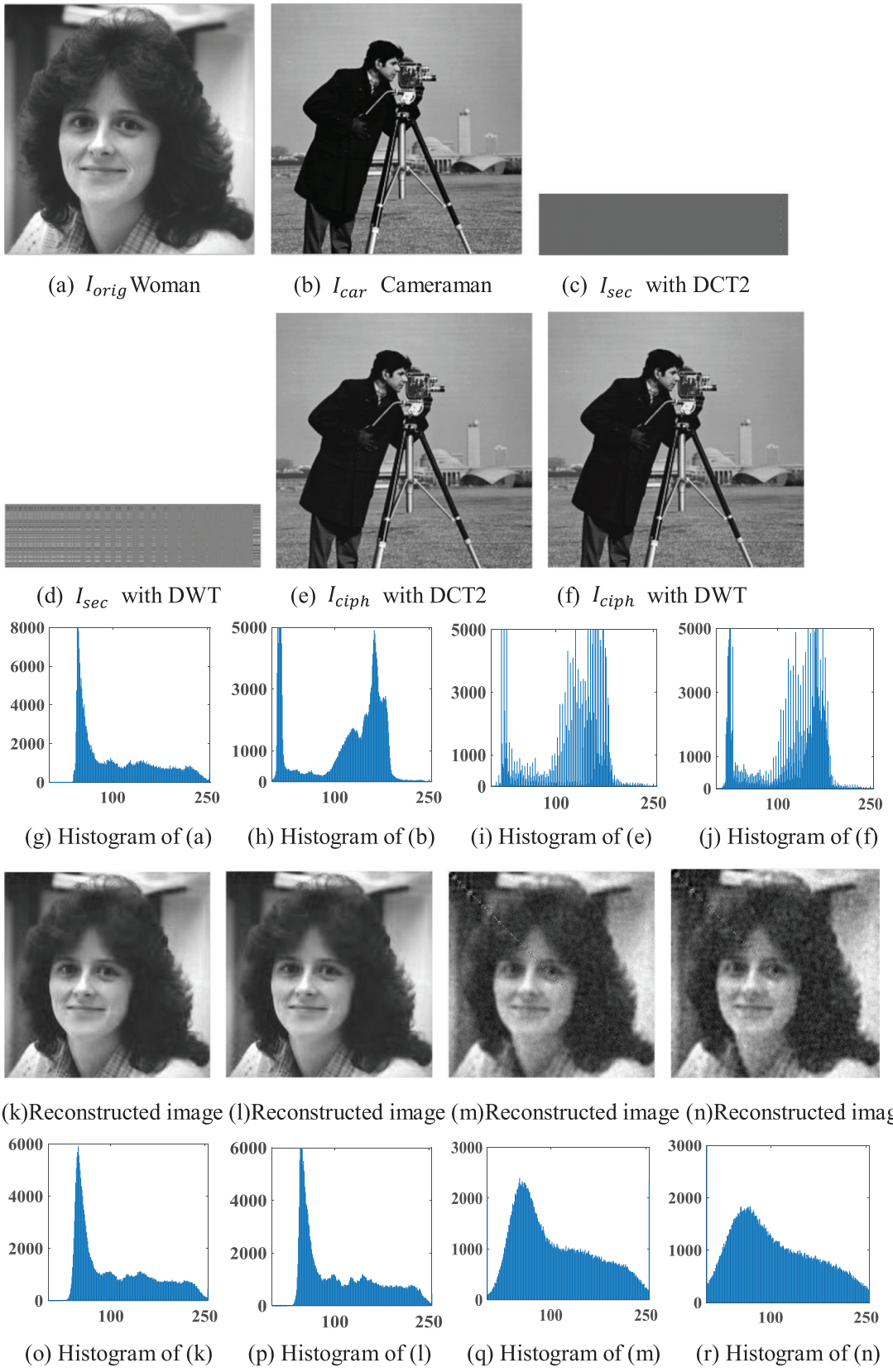

Fig. 8. Simulation results of different sparse and reconstruction methods for I_{orig} Woman.

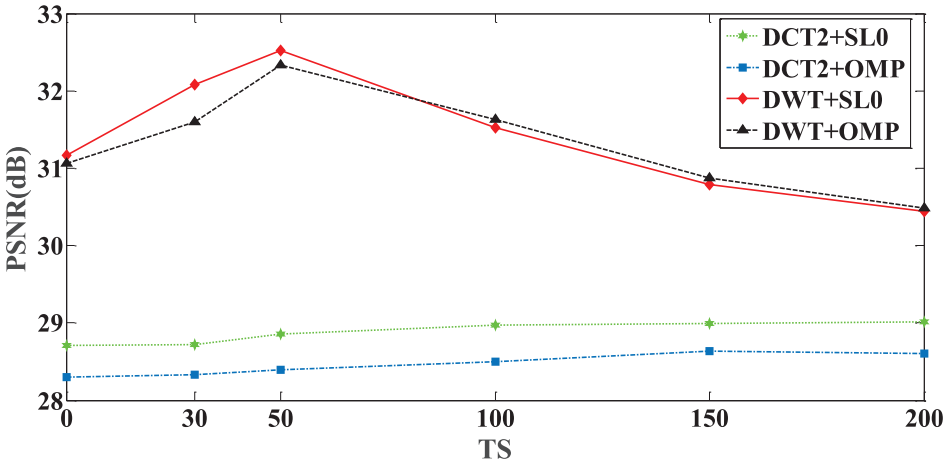

Fig. 9. PSNR vs TS for Lena with different sparse and reconstruction methods.

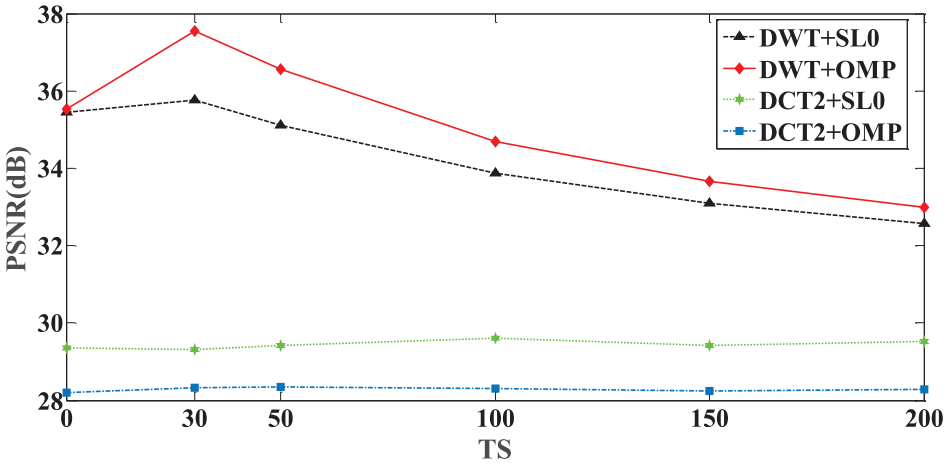

Fig. 10. PSNR vs TS for Woman with different sparse and reconstruction methods.

Woman(512×512). When DWT is used to sparsify the I_{orig}, PSNR value is around 32 dB for Lena and 38 dB for Woman. With the increasing of *TS*, PSNR values between the reconstructed image and I_{orig} are decreasing. When DCT2 is utilized, PSNR values are less than 30 dB. Conclusively, *TS* and reconstruction methods have little impact on the reconstruction effect.

4.2.2. *Influence of different* I_{car} *on encryption and decryption results*

In this subsection, simulation results for Peppers (512×512) with different I_{car} are presented and illustrated in Figs. 11 and 12. In Fig. 11, the I_{orig}, I_{car}, and their histograms are provided. The I_{car}s are all 512×512, and they are Goldhill, Brone, Aerial and Bridge shown in Fig. 11(b)-(e). The reconstruction method is OMP. In Fig. 12, (i)–(l) are the subtraction images between the I_{ciph} and their respective I_{car}s, and (m)–(p) are the respective reconstructed images of I_{ciph} shown in (a)–(d). Mean Structural Similarity (MSSIM) is applied for measuring the similarity between two images, and it is calculated by [23]

$$\begin{cases} l(X,Y) = \dfrac{2\mu_X\mu_Y + C_1}{\mu_X^2 + \mu_Y^2 + C_1} \\ c(X,Y) = \dfrac{2\sigma_X\sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2} \\ s(X,Y) = \dfrac{\sigma_{XY} + C_3}{\sigma_X\sigma_Y + C_3} \\ \mathrm{SSIM}(X,Y) = l(X,Y) \times c(X,Y) \times s(X,Y) \\ \mathrm{MSSIM}(X,Y) = \dfrac{1}{M}\sum\limits_{k=1}^{M} \mathrm{SSIM}(x_k, y_k) \end{cases} \tag{22}$$

where μ_X and μ_Y are the average values of I_{orig} X and reconstructed image Y, σ_X and σ_Y are variance values of X and Y, respectively, σ_{XY} is the covariance of X and Y, $C_1 = (k_1 \times L)^2$, $C_2 = (k_2 \times L)^2$, $C_3 = \frac{C_2}{2}$, $k_1 = 0.01$, $k_2 = 0.03$, the total number M of image blocks is 64, L is the gray level of I_{orig}, and $L = 255$.

Table 1
PSNR, MSSIM and CC between I_{ciph} and I_{car}.

I_{car}	I_{ciph}	PSNR(dB)	MSSIM	CC
Goldhill	Fig. 12(a)	36.49	0.9976	0.9993
Brone	Fig. 12(b)	30.74	0.9625	0.9996
Aerial	Fig. 12(c)	44.88	0.9993	0.9990
Bridge	Fig. 12(d)	35.96	0.9904	0.9995

Moreover, the correlation coefficient (CC) is also computed by the following equation [24]:

$$CC = \frac{L\sum\limits_{i=1}^{L}(x_i y_i) - \sum\limits_{i=1}^{L} x_i \sum\limits_{i=1}^{L} y_i}{\sqrt{\left(L\sum\limits_{i=1}^{L} x_i^2 - \left(\sum\limits_{i=1}^{L} x_i\right)^2\right)\left(L\sum\limits_{i=1}^{L} y_i^2 - \left(\sum\limits_{i=1}^{L} y_i\right)^2\right)}} \tag{23}$$

where x_i and y_i are the values of two pixels in two different images or two different pixels in one image, L is the total number of selected pixels, and here $L = 512 \times 512 = 262{,}144$.

The PSNR, MSSIM, and CC values between I_{ciph} and I_{car} are listed in Table 1. As can be seen, I_{ciph} are all meaningful, there are different I_{ciph} for different I_{car}, and they are all visually similar with the corresponding I_{car}. PSNR values are larger than 30 dB, MSSIM are more than 0.96, and CC values are close to 1. The decrypted images are the same as the I_{orig}, and their histograms are all identical. That means that the I_{car} have no effect on the reconstruction results of the I_{orig}.

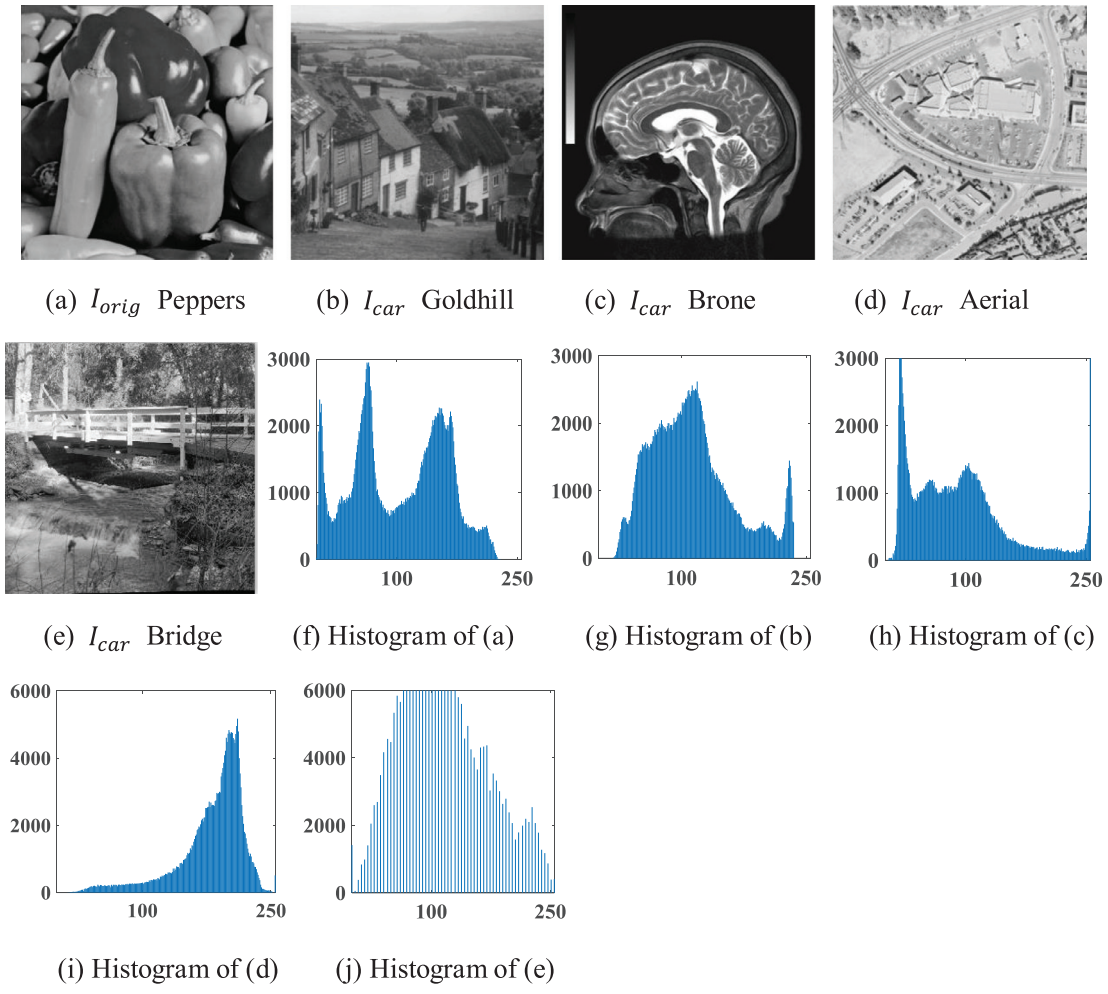

Fig. 11. The I_{orig}, I_{car} and their histograms.

Table 2
MSSIM values of different images.

Images	Lena	Finger	Airplane	Woman	Brone	Cameraman
Size	256×256	256×256	256×256	512×512	512×512	512×512
$CR=0.25$	0.9913	0.9811	0.9701	0.9987	0.9915	0.9813
$CR=0.5$	0.9920	0.9891	0.9704	0.9980	0.9933	0.9844
$CR=0.75$	0.9923	0.9892	0.9703	0.9983	0.9939	0.9884

4.3. Simulation results of compression and encryption method based on CS and Zigzag confusion

In our VMICE algorithm, we also present a good compression and encryption scheme based on CS and Zigzag confusion shown in Section 3.3.1. Fig. 13 is the encrypted and decrypted results for Woman (512×512) by this scheme. It is evident from the figure that when compression ratio *CR* changes from 0.25, 0.45, 0.65 to 0.85, I_{sec} are compressed and visually noisy, proving that I_{orig} information may not be extracted from them. Also, it is shown that the decoded images have good visual appearance, and they are just like I_{orig}.

In order to quantitatively analyze the reconstruction effect of the proposed compression and encryption method, six different I_{orig}s are chosen and tested. They are Lena, Airplane, Finger, Woman, Cameraman, and Brone. The former three images are 256×256, and the latter are 512×512. Simulation results are plotted in Fig. 14. As Fig. 14 shows, when *CR* increases, more data are sampled from the I_{orig}s for encryption and reconstruction, and thus PSNR values are gradually increasing. When *CR* is 0.25, for example, PSNR values for Woman and Cameraman are larger than 30 dB. Also, when *CR* is 0.5, PSNR values for these six images are all greater than 30 dB, which means the satisfactory reconstruction effect can be achieved sampling just half of the data.

Table 3
PSNR values for comparison.

Image	*CR*	Ours	Ref. [23]	Ref. [34]
Lena (256×256)	$CR=0.25$	26.56 dB	26.06 dB	25.93 dB
	$CR=0.5$	29.83 dB	29.82 dB	29.82 dB
	$CR=0.75$	31.62 dB	29.56 dB	34.19 dB

Table 2 listed the test results for the above six images. As can be observed from Table 2 that when the *CR* varies from 0.25, 0.5, to 0.75, MSSIM values are greater than 0.97. Specifically, MSSIM is larger than 0.99 for Lena, Woman, and Brone, which means that the reconstruction image and I_{orig} are highly similar, and our algorithm has good decryption effect. Additionally, the PSNR values between the decrypted image and I_{orig} Lena are also calculated and listed in Table 3. As shown in Table 3, the proposed VMICE algorithm is just good with Refs. [23,34]. There-

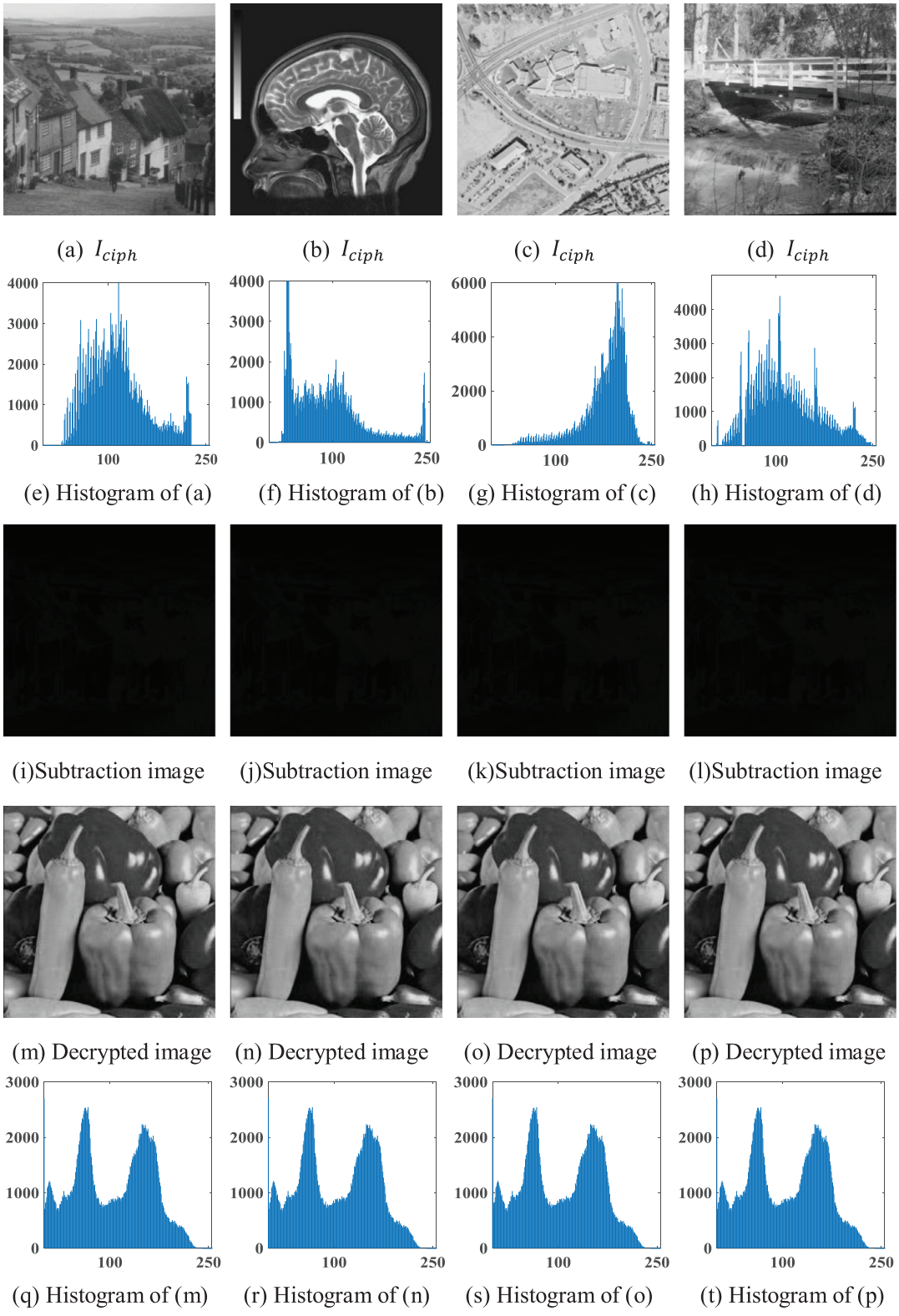

Fig. 12. I_{ciph} and decrypted images for Peppers (512 × 512) with different I_{car}.

fore, we can say that our algorithm can be applied for image secure communication.

4.4. Key space analysis

The security of an image cryptosystem is proportional to its key space. In general, the larger the key space, the stronger the ability of IE algorithm to withstand brute force attacks. In the proposed VMICE algorithm, the parameters t_1, t_2 for computing the starting position (x_0, y_0) of Zigzag confusion are the secret keys, system parameter μ and initial value r_0 of the Logistic map are regarded as keys, and sampling distance d for constructing measurement matrix and threshold TS are also part of the secret keys. Consequently, if the computational precision of a computer system is 10^{-14}, the complete key space is larger

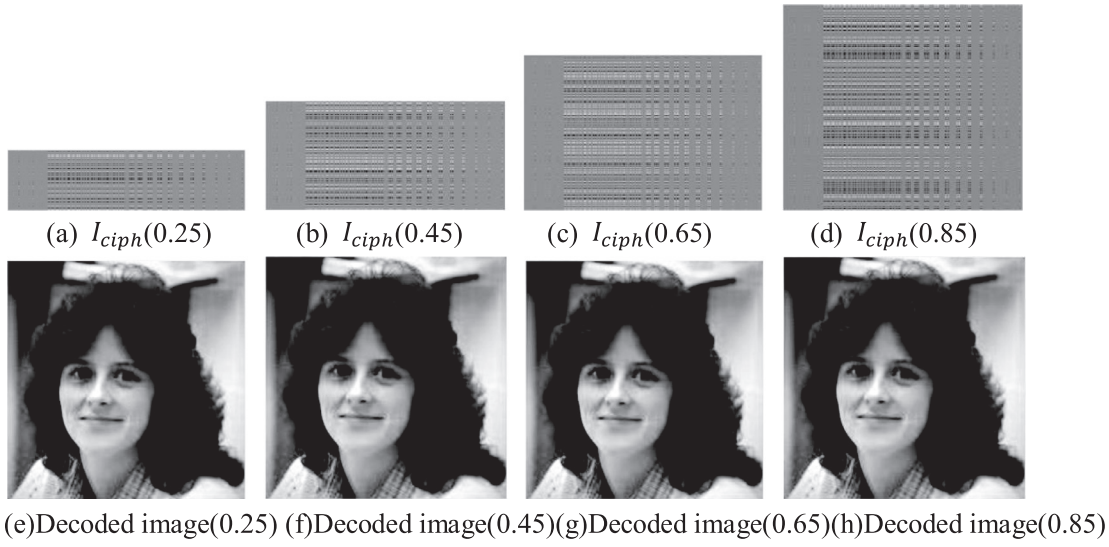

Fig. 13. Encrypted and decrypted results of Woman (512 × 512) when *CR* varies from 0.25, 0.45, 0.65 and 0.85.

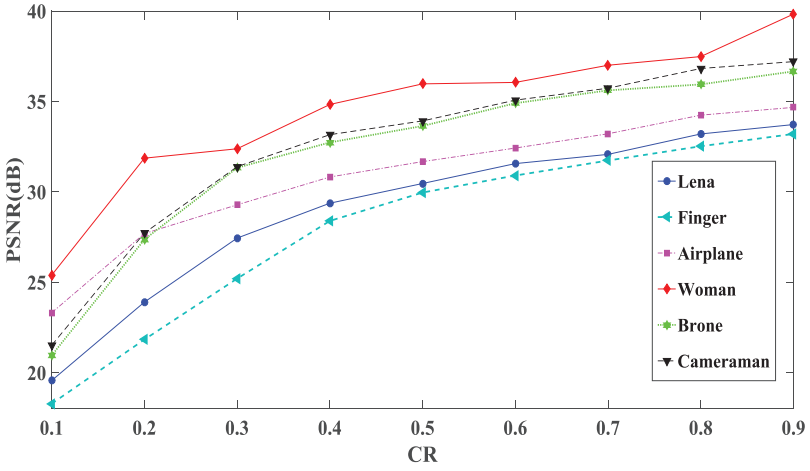

Fig. 14. PSNR vs CR for different I_{orig}.

Table 4
Key space comparison results.

Algorithm	Ours	Ref. [28]	Ref. [35]	Ref. [36]	Ref. [37]	Ref. [38]
Key space	10^{56}	10^{42}	2^{128}	2^{96}	$10^{44}\times5$	2^{78}

than 10^{56}. As can be seen in Table 4, this key space is larger than those in Refs. [28,35–38]. Thus, the proposed VMICE algorithm has enough security to withstand any previously considered brute force attacks. If we regard the I_{car} as keys, the key space is even larger.

4.5. Key sensitivity analysis

Key sensitivity is an important parameter to assess the security of an ICE algorithm [27,30]. In this subsection, Woman sized of 512 × 512 is used as the I_{orig}, and Cameraman of the same size as the I_{car}. In the simulation, key parameters are fixed as described in Section 4.1. The modified keys are obtained by adding 10^{-14} to one of t_1, t_2, μ, and r_0. Each time one paremeter varies, and others are constant.

Key sensitivity results in encryption are illustrated in Fig. 15. As is observed from this figure that the I_{ciph} with the correct key and modified keys are almost the same as the I_{car}, and their histograms are shown in (f), (k)-(n) are slightly different from histogram of I_{car} illustrated in (e). Concisely, modifying keys has little effect on the I_{ciph}s.

Table 5
Pixel change ratio between the decrypted image and I_{orig}.

Decrypted image	Decryption key	Pixel change ratio
Fig. 16(a)	Correct key	0
Fig. 16(b)	t_1+10^{-14}	99.92%
Fig. 16(c)	t_2+10^{-14}	99.88%
Fig. 16(g)	$\mu+10^{-14}$	99.88%
Fig. 16(h)	r_0+10^{-14}	99.87%

Fig. 16 plots key sensitivity results in the decryption phase. The pixel change ratio between the decrypted image and I_{orig} are listed in Table 5. As shown in Fig. 16 and Table 5, the I_{orig} can be successfully decrypted with the correct key. Also, the PSNR between Fig. 16(a) and the I_{orig} is 39.0461 dB. The recovered images are all noisy with the modified keys, and there are more than 99% pixel changed between the wrong decrypted images and I_{orig}. These results clearly show that the proposed encryption method has high sensitivity to secret keys in decryption process.

4.6. Robustness analysis

In the transmission of I_{ciph}, it will be contaminated by noises or occlusion. Thus, the resistances of the cryptosystem against noise and data loss are also important features [30].

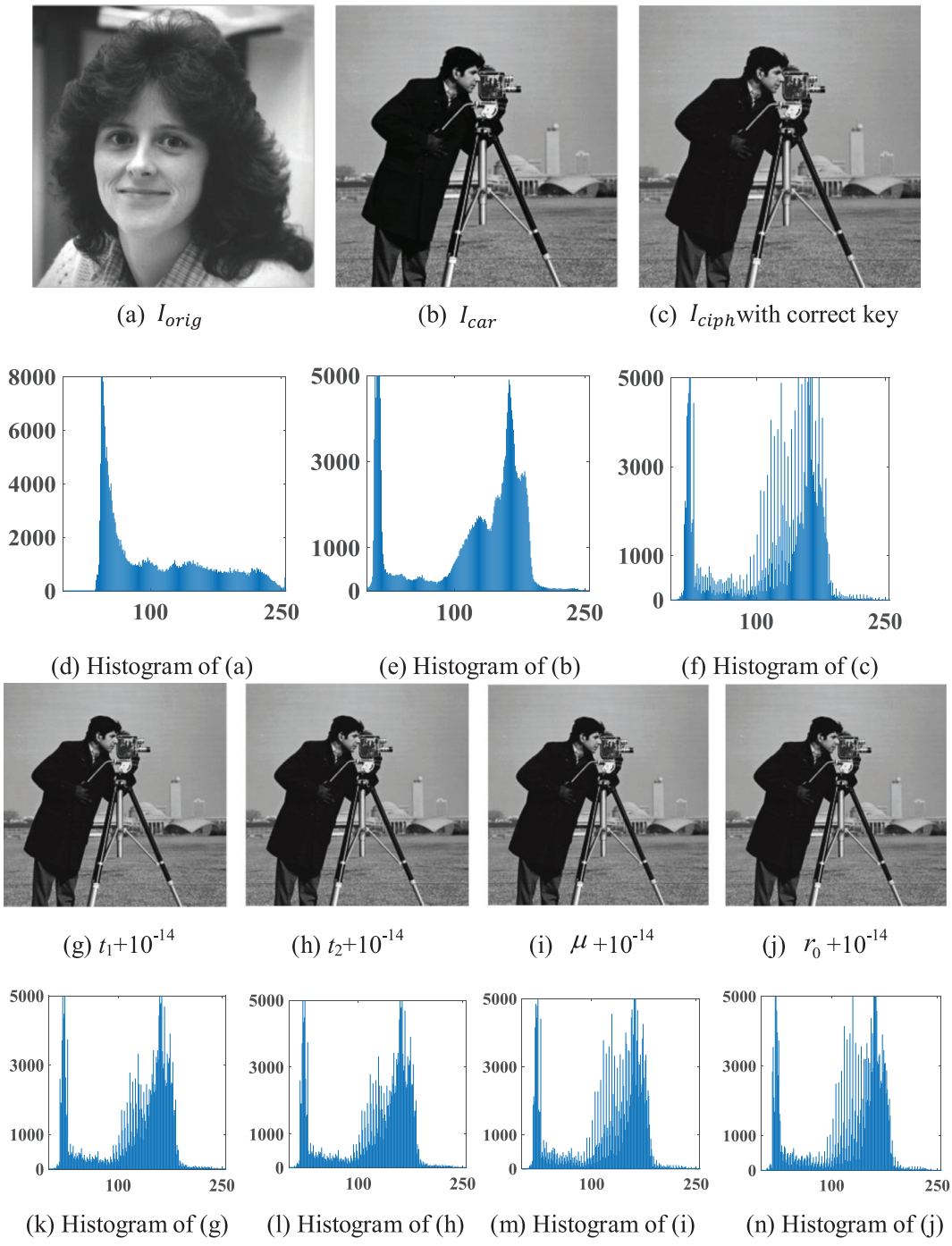

Fig. 15. Key sensitivity results in encryption process.

4.6.1. Noise attack

In this subsection, the I_{ciph} shown in Fig. 7(d) is the test image. Salt and pepper noise with different intensities are added to the I_{ciph}, with the noisy I_{ciph} and the resultant decrypted images shown in Fig. 17. Also, the corresponding PSNR values for Lena and Woman under different salt and pepper noise attacks are listed in Table 6. Airfield sized of 256 × 256 is used as the I_{car} for I_{orig} Lena, and they are of the same size. As shown in Fig. 17 and Table 6, when the noise intensity changes from 0 to 0.01, visual quality of the decrypted image is reduced, but one may find the I_{orig} information from the decrypted image. Additionally, the PSNR values for Woman decrease to 28.79 dB from 39.05 dB. These results indicate that the proposed VMICE algorithm may resist salt and pepper noise attack.

4.6.2. Data loss attack

To test the robustness against data loss attack, the I_{ciph} shown in Fig. 7(d) is tampered with different levels of data loss, and the simulation results are displayed in Fig. 18. In Fig. 18, the first row are the four I_{ciph}s, data loss sizes are from 32 × 32, 64 × 64, 100 × 100 to 130 × 130, and the second row are their corresponding decrypted images. Table 7 gives the PSNR, MSSIM and CC values of decrypted images and I_{orig}. As is evident from Fig. 18 and Table 7, when more data are lost in the I_{ciph}, visual quality of the decrypted image is reduced. However, the content of the I_{orig} is still be easily observed when data of 130 × 130 is lost, with the corresponding PSNR is 29.15 dB, MSSIM is 0.9194 and CC value is 0.9042.

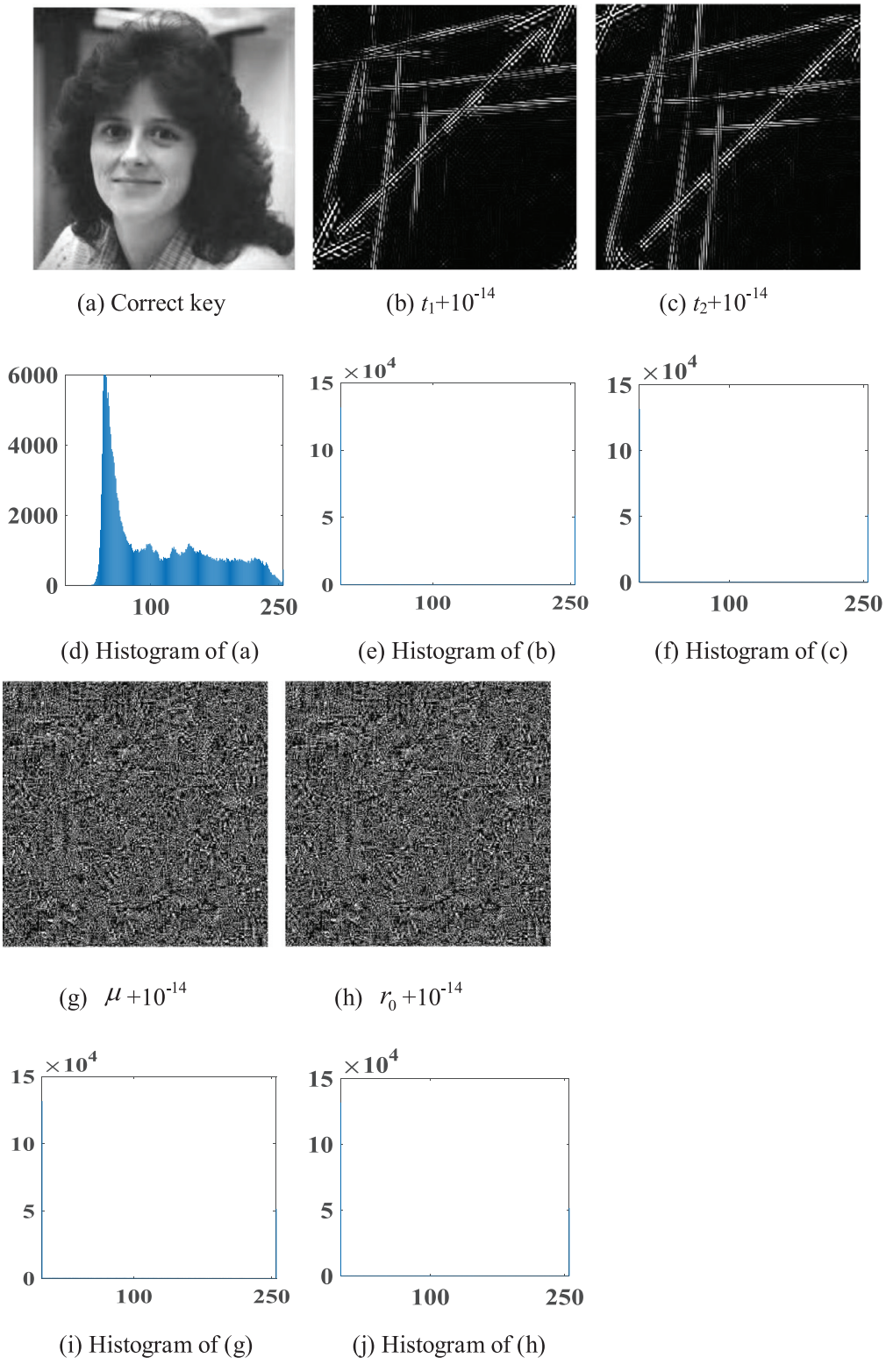

Fig. 16. Key sensitivity results in decryption process.

Table 6
PSNR values under different salt and pepper noise attack.

Image	Noise Intensity					
	0	0.000001	0.00001	0.0001	0.001	0.01
Lena (256×256)	32.83 dB	32.03 dB	31.99 dB	31.60 dB	30.91 dB	28.70 dB
Woman (512×512)	39.05 dB	38.75 dB	37.05 dB	35.22 dB	30.67 dB	28.79 dB

Fig. 17. Robustness test results against noise attack for Woman (512×512).

4.7. Running efficiency analysis

Running time of the proposed encryption is also an important performance index, especially for real time applications. Tables 8–12 give the encryption time and decryption time for different images. In these tables, 'Compression' denotes the compression and encryption process, 'Extraction' represents extracting the I_{sec} from I_{ciph}, and 'Reconstruction' is recovering the I_{orig} from the I_{sec}. As shown in Tables 8–12, when the size of I_{orig} is from 256×256 to 512×512 and 1024×1024, the total encryption time changes from 0.345 *s*, to 2.565 *s*, to 10.71 *s*, and the time

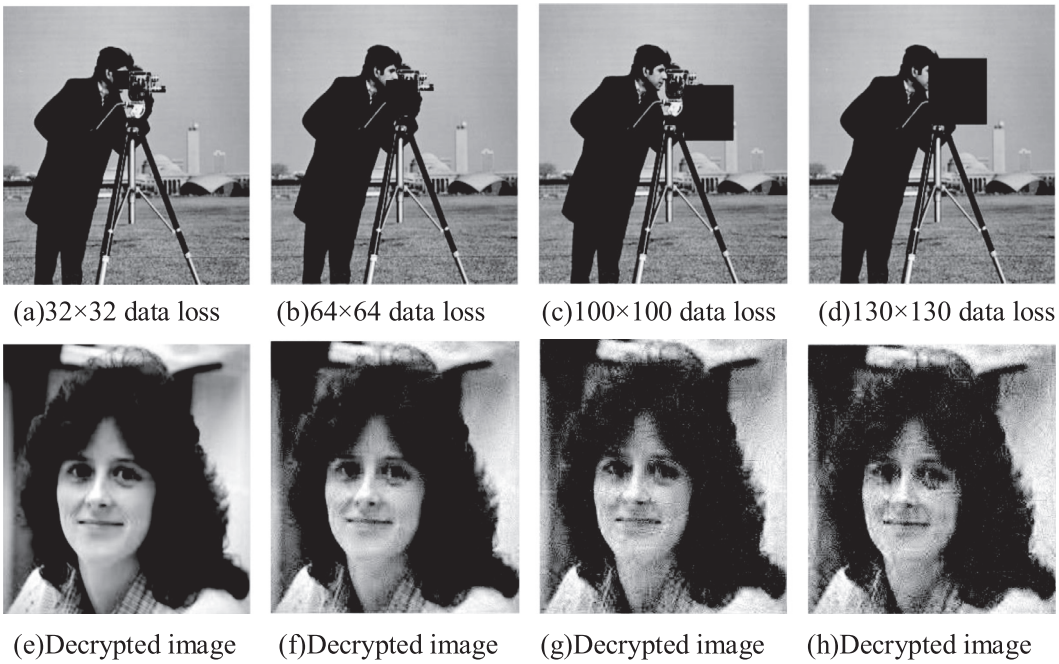

Fig. 18. Robustness test against data loss.

Table 7
PSNR, MSSIM and CC values for different data loss.

Size of data loss	PSNR	MSSIM	CC
32×32	38.19 dB	0.9966	0.9950
64×64	34.05 dB	0.9742	0.9692
100×100	31.48 dB	0.9355	0.9244
130×130	29.15 dB	0.9194	0.9042

Table 8
Encryption time for images sized of 256×256 (Unit: *s*).

Item	Lena	Cameraman	Peppers	Baboon	Average
Compression	0.10	0.11	0.12	0.10	0.1075
LSB embedding	0.24	0.25	0.22	0.23	0.235
Total	0.35	0.36	0.34	0.33	0.345

Table 9
Decryption time for images sized of 256×256 (Unit: *s*).

Item	Lena	Cameraman	Peppers	Baboon	Average
Extraction	0.28	0.27	0.25	0.20	0.25
Reconstruction	1.25	1.26	1.25	1.22	1.245
Total	1.53	1.53	1.50	1.42	1.495

Table 10
Encryption time for images sized of 512×512 (Unit: *s*).

Item	Lena	Cameraman	Peppers	Baboon	Average
Compression	0.53	0.55	0.54	0.56	0.545
LSB embedding	2.00	2.05	2.03	2.05	2.0325
Total	2.53	2.55	2.57	2.61	2.565

Table 11
Decryption time for images sized of 512×512 (Unit: *s*).

Item	Lena	Cameraman	Peppers	Baboon	Average
Extraction	1.70	1.73	1.72	1.72	1.7175
Reconstruction	6.68	6.80	5.81	6.15	6.36
Total	8.38	8.33	7.53	7.87	8.0275

consumed on decryption varies from 1.495 *s*, to 8.0275 *s*, to 106.43 *s*. In the encryption phase, the time consumed on compression is very little, with about 80% of total time instead spent on LSB embedding. When the image size is larger, more data are needed to embed and thus more time is spent. In the decryption phase, for a I_{orig} of size 256×256 and 512×512, the time spent on Extraction is about 40% of the total time. But when the image size is 1024×1024, around 94% of total time is used for Reconstruction. Therefore, our VMICE algorithm can be directly applied for the encryption and decryption of small and medium images. When the test image is large, assistance from the cloud can be required, with the encryption operation occurring locally, and the decryption phase being done in the cloud.

4.8. Known-plaintext and chosen-plaintext attack analysis

Until now, many IE algorithms have been cracked by known-plaintext and chosen-plaintext attacks [39–42]. In this paper, the average value I of a I_{orig} is first computed, and parameters Q, x_0, y_0, a_x, a_y, a_z, b_x, b_y, b_z, X_0, Y_0, Z_0 are further obtained by I_{orig} information. In the encryption phase, x_0, y_0 are used as the starting positions of Zigzag confusion, and a_x, a_y, a_z, b_x, b_y, b_z, X_0, Y_0, Z_0 are utilized as the system parameters and initial values of 3-D Cat map. Chaotic sequences are gen-

Table 12
Encryption time and decryption time for images sized of 1024×1024 (Unit: *s*).

Image size	Encryption phase			Decryption phase		
	Compression	LSB embedding	Total	Extraction	Reconstruction	Total
1024×1024	1.23	9.48	10.71	6.83	99.60	106.43

erated by iterating Cat map, and then utilized to produce measurement matrix. Therefore, the compressed I_{sec} will vary for different I_{orig}. Also, the keys need not be changed frequently. Just for the same test image, when secret keys are modified, the I_{sec} will change largely.

In LSB embedding phase, the chaotic sequences X, Y, Z and W are obtained by I_{orig} information and 3-D Cat map, and index arrays IA_1, IA_2, IA_3 and IA_4 are obtained by sorting them. Then, the four equal parts of the I_{car} are shuffled by these index arrays. Finally, the I_{sec} is embedded into the permutated I_{car}. From the embedding process, it is evident that for the same I_{car} and secret keys, the embedded results are changing with the I_{orig}.

To summarize, the proposed VMICE algorithm is highly sensitive to the I_{orig}, and it may withstand known-plaintext and chosen-plaintext attacks effectively.

4.9. Comparison with the existing work

4.9.1. Qualitative analysis and comparison

At present, the ICE algorithms usually transform a natural image into a noise-like or texture-like one, thus the secret information of I_{orig} is hidden and protected. Unfortunately, during the process of transmission and storage, the noise-like I_{ciph} easily arouse the attention of hackers and are selectively analyzed. In the proposed VMICE algorithm, the I_{orig} is first compressed and encrypted into a noisy I_{sec}, and then the I_{sec} is embedded into a I_{car}, and ultimately, the obtained I_{ciph} has the same appearance with the I_{car}. Disguising themselves as regular images may prevent I_{ciph} from being detected and analyzed. In this sense, our algorithm is highly secure.

Bao and Zhou [26] first proposed the concept of VM encrypted image. In their scheme, the I_{orig} of size $m \times n$ was encrypted into a I_{ciph} using the existing encryption method, and then the I_{ciph} was embedded into the I_{car} of size $2m \times 2n$ by DWT embedding. The final VM I_{ciph} is four times as large as the I_{orig}. Thus, when this I_{ciph} is transmitted and stored, more bandwidth and space are occupied. The ICE algorithms proposed by Wen et al. [27], Kanso and Ghebleh [28], and Manikandan and Masilamani [29] also have this shortcoming. To resolve this problem, CS is adopted to compress and encrypt the I_{orig} into a I_{sec} in our work. Via this method, its size is reduced to a quarter of I_{orig}, and the final I_{ciph} and I_{orig} are of the same size. Our cryptosystem is more efficient from this point.

The embedding operation is mostly utilized to obtain a VM I_{ciph}. In the adoption of embedding method, DWT embedding is often used [27,30]. However, DWT may transform one integer to another decimal, it is not reversible. That is to say, when I_{ciph} is obtained by DWT embedding, I'_{sec} extracted from I_{ciph} is not equal to I_{sec}. Thus, the embedding process is lossful and it is affected by I_{car}. Just for the same I_{orig}, different decrypted image can be obtained by choosing different I_{car}. To solve this question, LSB embedding is used in our algorithm. The proposed image cryptosystem are independent of I_{car}, and more I_{car} can be applied.

Additionally, in the embedding process of Ref. [26], the LH and HH sub-bands of I_{car} are respectively replaced with the tens and units of I_{ciph} element in sequence. The orderly embedding increases the probability the algorithm will be cracked. In the proposed cryptosystem, the random-like chaotic sequences obtained by the 3-D Cat map are utilized to decide where to embed the I_{sec}. Therefore, the security level has been greatly improved. Additionally, the initial values and system parameters of the 3-D Cat map are generated by the average value of I_{orig}. Different embedding methods are designed for different I_{orig}, which increases the ability of the proposed algorithm to resist against known-plaintext and chosen-plaintext attacks.

4.9.2. Quantitative analysis and comparison

A good visually meaningful image encryption algorithm requires that the I_{ciph} has high similarity with the I_{car}. Table 13 shows the PSNR and MSSIM values between the I_{ciph} and I_{car}, and also shows comparison results with other methods. It is worth noting that the size of I_{ciph} is similar to that of the I_{car}. The maximum values for PSNR and MSSIM are in boldface. As displayed in Table 13, the I_{orig} of size 256×256 is embedded into the four I_{car}s of size 512×512 in Ref. [26] and Ref. [28], and the size of the I_{car} is four times of that of I_{orig}. But for Ref. [30] and our algorithm, the I_{car} has the same size as the I_{orig}, and thus system resources are saved for no additional transmission bandwidth and storage space. The PSNR and MSSIM values generated by our algorithm are all higher than those in Refs. [26,28,30]. Specifically, the PNSR values obtained by our method are more than 33 dB, and MSSIM values are larger than 0.99. These results show that the I_{ciph} obtained by the proposed encryption are more similar with the I_{car}.

Table 13
Comparison with other studies.

I_{orig}	Algorithm	I_{car} (512×512)	PSNR(dB)	MSSIM
Lena (256×256)	Ref. [26]	Peppers	27.18	0.9790
		Splash	27.50	0.9788
		Tiffany	26.20	0.9228
		House	26.23	0.9671
	Ref. [28]	Peppers	34.16	0.9949
		Splash	35.70	0.9961
		Tiffany	31.51	0.9691
		House	30.55	0.9856
Lena (512×512)	Ref. [30]	Peppers	27.74	0.9638
		Splash	32.37	0.9915
		Tiffany	27.08	0.9865
		House	30.15	0.9964
	Ours	Peppers	**34.51**	**0.9952**
		Splash	**35.81**	**0.9982**
		Tiffany	**36.74**	**0.9894**
		House	**33.29**	**0.9980**

To analyze the effect of I_{car} on decrypted performance, the PSNR values between the decrypted image and I_{orig} are computed and shown in Fig. 19. In order to compare it with Ref. [30], the I_{orig} is Peppers, the I_{car} are Goldhill, Brone, Aerial, and Bridge, and threshold TS varies from 0 to 200. The corresponding result in Ref. [30] is illustrated in Fig. 20. As shown in Figs. 19 and 20, the PSNR values are changing with the I_{car} in Ref. [30], which limits the selection of the I_{car} to attain the reconstruction requirements. However, the PSNR values achieved by our algorithm are the same. That means that the I_{car} does not affect the decrypted image in our algorithm, and provides a meaningful appearance for protecting the I_{orig}. Therefore, the proposed VMICE algorithm has wider selection range for I_{car}.

The decrypted effect of our algorithm and other methods is compared and listed in Table 14. Barbara (512×512) is used as the I_{orig}, Lena, Bridge, Girl, and Peppers of the same size are employed as I_{car}, and PSNR and MSSIM values between the decrypted image and I_{orig} are computed. It is evident from Table 14 that PSNR and MSSIM values are varying for different I_{car} in Ref. [30], which means that I_{car} has some effect on the quality of the decrypted images. But for Ref. [43] and our algorithm, when the I_{car} is changing, PSNR and MSSIM values are fixed, and those results achieved by our algorithm are all larger than those in Ref. [43]. These results demonstrate that the decrypted results obtained by Ref. [43] and our method have nothing to do with the I_{car}, and the proposed encryption scheme has higher reconstruction quality.

Additionally, we compare the running time with other ICE algorithms, and the compared results are illustrated in Table 15. In this table, the test images are Finger (256×256) and Baboon (256×256), Cameraman (256×256) is used as the I_{car}, and the shortest times for encryption, decryption, and total processes are in boldface. As is observed from this table, when the size of the image is 256×256, the algorithm in Ref. [24] has the shortest encryption time, it is no more than 0.1 s; the decryption time of our algorithm is the shortest, and it is about 1.5 s. Also, the proposed encryption has a shorter time than those in Refs. [24,30,43] by integrating encryption and decryption time.

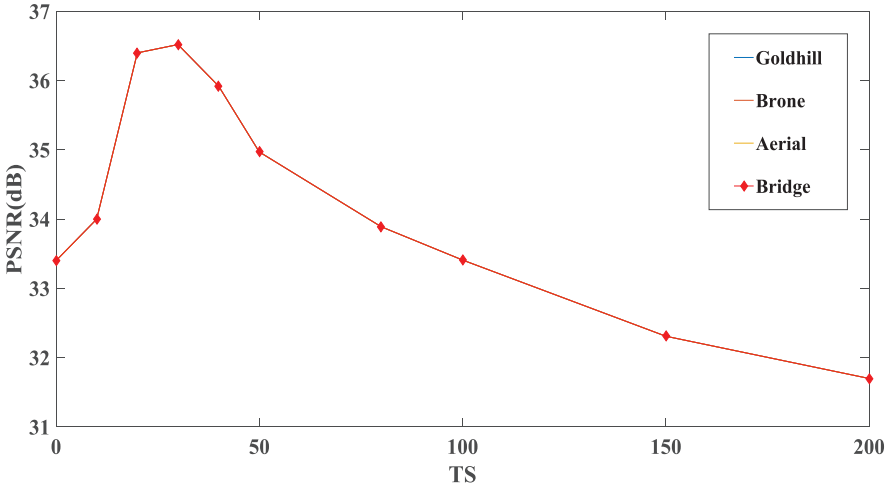

Fig. 19. PSNR values obtained by our algorithm.

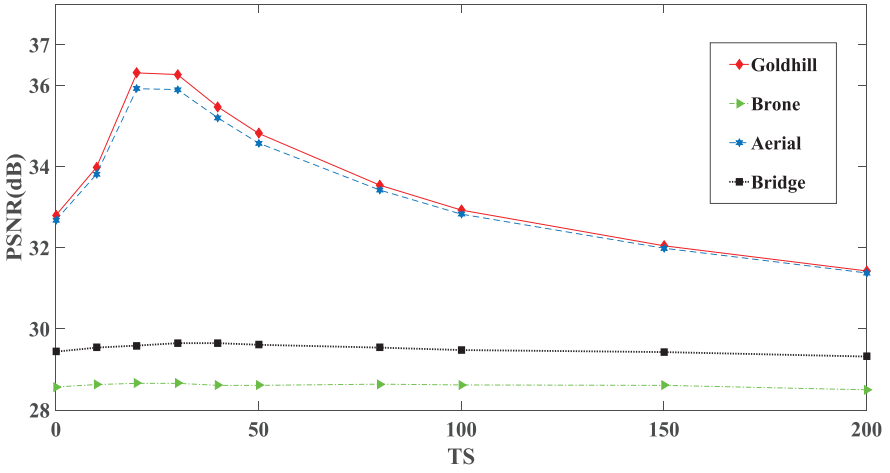

Fig. 20. PSNR values obtained from Ref. [30].

Table 14
PSNR and MSSIM between the decrypted image and I_{orig}.

I_{orig}	I_{car}		Ref. [30]	Ref. [43]	Ours
Barbara (512×512)	Lena (512×512)	PSNR(dB)	28.48	28.44	28.55
		MSSIM	0.9915	0.8128	0.9932
	Bridge (512×512)	PSNR(dB)	28.17	28.44	28.55
		MSSIM	0.9865	0.8128	0.9932
	Girl (512×512)	PSNR(dB)	28.19	28.44	28.55
		MSSIM	0.9872	0.8128	0.9932
	Peppers (512×512)	PSNR(dB)	28.23	28.44	28.55
		MSSIM	0.9891	0.8128	0.9932

Table 15
Comparison results of running time (Unit: *s*).

I_{orig}	Item	Ref. [24]	Ref. [30]	Ref. [43]	Ours
Finger (256×256)	Encryption time	**0.0819**	0.1159	0.1544	0.3356
	Decryption time	2.2841	2.2295	2.2489	**1.5216**
	Total	2.366	2.3454	2.4033	**1.8572**
Baboon (256×256)	Encryption time	**0.0782**	0.1181	0.1523	0.3319
	Decryption time	2.2818	2.3235	2.2870	**1.5342**
	Total	2.360	2.4416	2.4393	**1.8661**

5. Conclusions

This paper introduces an efficient VMICE algorithm to protect images. It is made up of two components: pre-encryption and embedding. In pre-encryption, a compressed I_{sec} is obtained by use of CS and Zigzag confusion –the resolution of the image is reduced and the data content is protected. Dynamic LSB embedding is then used to embed the I_{sec} into a I_{car}. Appearance of the final I_{ciph} is meaningful, which may better protect the I_{orig}. Moreover, the embedding process is controlled by random chaotic sequences generated from a 3-D Cat map, and out-of-order embedding increases the difficulty of the decoding process. The size of the I_{ciph} and I_{orig} is the same, and no extra transmission bandwidth and storage space are required. Additionally, the utilization of LSB embedding makes the decryption of algorithm completely independent of the I_{car}, which may increase the generalizability of the proposed ICE algorithm.

Simulation results and performance analyses have been carried out. Specifically, image encryption and decryption effect, visual quality, key space, key sensitivity, robustness analysis, running efficiency, known-plaintext, and chosen-plaintext attack and comparison analysis are comprehensively measured. The satisfactory performance makes the proposed scheme a suitable solution for protecting digital images in the transmission and storage over public networks. In future work, we will design VMICE algorithm for one or more color images to protect the security of color images.

Declaration of Competing Interest

None.

Acknowledgments

All the authors are deeply grateful to the editors for smooth and fast handling of the manuscript. The authors would also like to thank the anonymous referees for their valuable suggestions to improve the quality of this paper. This work is supported by the National Natural Science Foundation of China (grant no. 41571417, U1604145, 61802111, 61872125, 61871175), Science and Technology Foundation of Henan Province of China (grant no. 182102210027, 182102410051), China Postdoctoral Science Foundation (grant no. 2018T110723, 2016M602235), Key Scientific Research Projects for Colleges and Universities of Henan Province (grant no. 19A413001), and Guangxi Key Laboratory of Trusted Software (grant no. kx201904).

References

[1] Amina S, Mohamed FK. An efficient and secure chaotic cipher algorithm for image content preservation. Commun Nonlinear Sci Numer Simul 2018;60:12–32.
[2] Hua ZY, Zhou YC. Image encryption using 2D logistic-adjusted-sine map. Inf Sci 2016;339:237–53.
[3] Chen JX, Zhu ZL, Fu C, Zhang LB, Zhang YS. An efficient image encryption scheme using lookup table-based confusion and diffusion. Nonlinear Dyn 2015;81:1151–66.
[4] Li XW, Lee I-K. Modified computational integral imaging-based double image encryption using fractional Fourier transform. Opt Lasers Eng 2015;66:112–21.
[5] Wang Y, Quan C, Tay CJ. Asymmetric optical image encryption based on an improved amplitude-phase retrieval algorithm. Opt Lasers Eng 2016;78:8–16.
[6] Wu XJ, Kan HB, Kurths J. A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps. Appl Soft Comput 2015;37:24–39.
[7] Chai XL, Fu XL, Gan ZH, Lu Y, Chen YR. A color image cryptosystem based on dynamic DNA encryption and chaos. Signal Process 2019;155:44–62.
[8] Chai XL, Chen YR, Broyde L. A novel chaos-based image encryption algorithm using DNA sequence operations. Opt Lasers Eng 2017;88:197–213.
[9] Li XW, Xiao D, Wang QH. Error-free holographic frames encryption with CA pixel-permutation encoding algorithm. Opt Lasers Eng 2018;100:200–7.
[10] Niyat A Y, Moattar MH, Torshiz MN. Color image encryption based on hybrid hyper-chaotic system and cellular automata. Opt Lasers Eng 2017;90:225–37.
[11] Zhou NR, Yan XY, Liang HR, Tao XY, Li GY. Multi-image encryption scheme based on quantum 3D Arnold transform and scaled Zhongtang chaotic system. Quantum Inf Process 2018;17:338.
[12] Hua ZY, Zhou YC, Huang HJ. Cosine-transform-based chaotic system for image encryption. Inf. Sci. 2019;480:403–19.
[13] Huang XL, Ye GD. An image encryption algorithm based on hyper-chaos and DNA sequence. Multimed Tools Appl 2014;72(1):57–70.
[14] Belazi A, EI-Latif AAA, Diaconu AV, Rhouma R, Belghith S. Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms. Opt Lasers Eng 2017;88:37–50.
[15] Manjit Kaur, Vijay Kumar. Beta chaotic map based image encryption using genetic algorithm. Int J Bifurcat Chaos 2018;28:1850132.
[16] Wu XJ, Wang KS, Wang XY, Kan HB. Lossless chaotic color image cryptosystem based on DNA encryption and entropy. Nonlinear Dyn 2017;90:855–75.
[17] Luo YL, Zhou RL, Liu JX, Cao Y, Ding XM. A parallel image encryption algorithm based on the piecewise linear chaotic map and hyper-chaotic map. Nonlinear Dyn 2018;93:1165–81.
[18] Li HJ, Wang YR, Zuo ZW. Chaos-based image encryption algorithm with orbit perturbation and dynamic state variable selection mechanisms. Opt Lasers Eng 2019;115:197–207.
[19] Zhang YS, Huang H, Xiang Y, Zhang LY, He X. Harnessing the hybrid cloud for secure big image data service. IEEE Internet Things 2017;4:1380–8.
[20] Zhang YS, Leo Zhang Y, Zhou JT, Liu LC, Chen F, He X. A review of compressive sensing in information security field. IEEE Access 2016;5:2507–19.
[21] Fang H, Vorobyov SA, Jiang H, Taheri O. Permutation meets parallel compressed sensing: how to relax restricted isometry property for 2D sparse signals. IEEE Trans Signal Process 2014;62:196–210.
[22] Zhou NR, Jiang H, Gong LH, Xie XW. Double-image compression and encryption algorithm based on co-sparse representation and random pixel exchanging. Opt Lasers Eng 2018;110:72–9.
[23] Chai XL, Zheng XY, Gan ZH, Han DJ, Chen YR. An image encryption algorithm based on chaotic system and compressive sensing. Signal Process 2018;148:124–44.
[24] Hu GQ, Xiao D, Wang Y, Xiang T. An image coding scheme using parallel compressive sensing for simultaneous compression-encryption applications. J Vis Commun Image Represent 2017;44:116–27.
[25] Chen JX, Zhang Y, Qi L, Fu C, Xu LS. Exploiting chaos-based compressed sensing and cryptographic algorithm for image encryption and compression. Opt Laser Technol 2018;99:238–48.
[26] Bao L, Zhou YC. Image encryption: generating visually meaningful encrypted images. Inf Sci 2015;324:197–207.
[27] Wen WY, Zhang YS, Fang YM, Fang ZJ. Image salient regions encryption for generating visually meaningful ciphertext image. Neural Comput. Appl. 2018;29:653–63.
[28] Kanso A, Ghebleh M. An algorithm for encryption of secret images into meaningful images. Opt Lasers Eng 2017;90:196–208.
[29] Manikandan VM, Masilamani V. An efficient visually meaningful image encryption using Arnold transform. In: Proceedings of the 2016 IEEE Students' Technology Symposium; 2016. p. 266–71.
[30] Chai XL, Gan ZH, Chen YR, Zhang YS. A visually secure image encryption scheme based on compressive sensing. Signal Process 2017;134:35–51.
[31] Candes EJ, Romberg J, Tao T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans Inf Theory 2006;52:489–509.
[32] Donoho DL. Compressed sensing. IEEE Trans Inf Theory 2006;52:1289–306.
[33] Chen G, Mao Y, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solit. Fract. 2004;21:749–61.
[34] Zhou NR, Zhang AD, Zheng F, Gong LH. Novel image compression-encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing. Opt Laser Technol 2014;62(10):152–60.
[35] Mohamed FK. A parallel block-based encryption schema for digital image using reversible cellular automata. Eng Sci Technol 2014;17(2):85–94.
[36] Georgs SN, Augustine N, Pattathil DP. Audio security through compressive sampling and cellular automata. Multimed Tools Appl 2015;74(23):10393–417.
[37] Bakhshandeh A, Eslami Z. An authenticated image encryption scheme based on chaotic maps and memory cellular automata. Opt Lasers Eng 2013;51(6):665–73.
[38] George SN, Pattathil DP. A secure LFSR based random measurement matrix for compressive sensing. Sens Imaging 2014;15(1):1–29.
[39] Alvarez G, Li SJ. Some basic cryptographic requirements for chaos-based cryptosystems. Int J Bifurc Chaos 2006;16:2129–51.
[40] Li CQ, Lin DD, Lü JH. Cryptanalyzing an image-scrambling encryption algorithm of pixel bits. IEEE Multimed 2017;24:64–71.
[41] Li CQ, Lin DD, Feng BB, Lü JH. Cryptanalysis of a chaotic image encryption algorithm based on information entropy. IEEE Access 2018;2018:2883690. doi:10.1109/ACCESS.
[42] Li CQ, Lin DD, Lü JH, Hao F. Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography. IEEE Multimed 2018;2018:2873472. doi:10.1109/MMUL.
[43] Wang H, Xiao D, Li M. A visually secure image encryption scheme based on parallel compressive sensing. Signal Process 2019;155:218–32.