IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 31, 2019, accepted November 17, 2019, date of publication November 25, 2019, date of current version December 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955570

Meaningful Encryption: Generating Visually Meaningful Encrypted Images by Compressive Sensing and Reversible Color Transformation

PING PING[1], JIE FU[1], YINGCHI MAO[1], FENG XU[1], AND JERRY GAO[2]
[1]College of Computer and Information, Hohai University, Nanjing 210098, China
[2]College of Engineering, San Jose State University, San Jose, 95192, USA

Corresponding author: Ping Ping (pingpingnjust@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61902110, in part by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant 2018YFC0407105 and Grant 2016YFC0400910, in part by the Fundamental Research Funds for the Central Universities under Grant 2017B16814 and Grant 2017B20914, in part by the Key Technology Project of China Hueneng Group under Grant HNKJ13_H17_04, and in part by the National Key Research and Development Program of China under Grant 2018YFC1508603 and Grant 2018YFC0407905.

ABSTRACT Recently, compressive sensing (CS) and visual security (VS) have caught researchers attention in information security field. However, the measurement matrix is often reused in CS, which makes it vulnerable to chosen plaintext attack (CPA). In addition, when generating meaningful cipher images, the size of the carrier image is usually not less than the size of the plain image. In order to overcome these drawbacks, a new visually secure image encryption scheme using CS and reversible color transformation is proposed. The algorithm consists of two stages: compression and embedding. In the first stage, chaotic sequence is used to generate different structurally random matrices. When CS is performed, a random number is added during the process of sampling. By choosing different random numbers, different measurement matrices can be used to compress and encrypt the same image in different order. In the second stage, block pairing, color transformation and block replacement are employed to obtain a meaningful image. Different from the block replacement between two similar images, this paper first attempts to replace the block of the carrier image with a compressed noise-like image block. Thus, the carrier image can be smaller than the plain image, which saves the bandwidth of transmission. Both theoretical analysis and experimental results show that the proposed encryption scheme has good encryption performance, can effectively resist common attacks, and is suitable for meaningful image encryption.

INDEX TERMS Image encryption, compressive sensing, structurally random matrix, image camouflage, visually secure.

I. INTRODUCTION

In recent years, with the development of Internet technology and the advent of 5G era, a surging number of people use digital images to communicate on the Internet. A large quantity of images with secret nature or without authorization are disseminated on the Internet. Since the transmission channel is insecure [1], various image cryptosystems based on chaos system [2]–[6], cellular automata (CA) [7]–[10], SCAN [11]–[13], DNA encoding [14]–[16], quantum computation [17]–[20], wave transform [21], [22] have been suggested to ensure the security of the image in the transmission process.

According to the representation of the encrypted image, image secure schemes can be divided into two categories: one is that the plain image is transformed into a noise-like or texture-like cipher image after applying encryption algorithms and the attackers cannot obtain the original image without the secret key; the other is that the plain image is secretly embedded in another meaningful image by some technologies and the attacker cannot figure out what the source image looks like.

The first method mentioned above is usually based on the traditional permutation-diffusion structure [7], [23]–[28],

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqi Wang.

which was first proposed by Fridrich [29]. In this structure, the pixel position is firstly shuffled by permutation to reduce the strong correlation between pixels adjacent to each other, and then the pixel value is modified by diffusion to achieve avalanche effect. For example, Chen et al. [30] suggested a novel chaos based image encryption. In this approach, 3D chaotic cat map was utilized during the permutation stage to change the pixel positions while Logistic map was used in substitution process. In [31], a dynamic state variables selection mechanism was introduced to improve the permutation-substitution structure. By using this mechanism, the state variables generated from the hyper chaotic systems are dynamically and pixel-related distributed to each pixel in both permutation and diffusion. In [32], a rewriting function is applied to the permuted image before the diffusion operation in order to avoid the separate attack. In [33], Hua et al. proposed a new two-dimensional Logistic-Sine-coupling mapping, which can quickly shuffle pixel row and column positions simultaneously and spread few changes of plain-image to the whole cipher-image to obtain diffusion property. In [34], Zhang *et al.* proposed an algorithm with the same encryption and decryption process, which enables the scheme to use a device to complete the encryption and the decryption and to save hardware resources. Recently, image encryption technology using compressive sensing [35], [36] is also widely used [37]–[41]. In [42], Luo *et al.* decomposed the plain-image using Haar wavelet and only measured the detail matrix using the deterministic measurement matrix generated by chaotic map, which greatly improved the robustness of the compression encryption scheme. In [43], Gong et al. effectively resisted the chosen plaintext attack by using the hash value of the original image as the initial parameter. In [44], Ponuma and Amutha employed Chebyshev and Tent map to generate the measurement matrix, and the results show that it has high security. In [38], Zhou et al. proposed a new security scheme for encrypting and compressing images simultaneously using hyperchaotic system and two-dimensional CS. In [45], Zhang *et al.* proposed a medical image encryption and compression scheme based on CS and pixel permutation approach. It can also simultaneously encrypt and compress the medial images. In [46], Chai *et al.* used CS technology to reduce the size of plain-image after encryption and save the transmission time in the network. One of the common points of the above encryption schemes is that the plain images are transformed into noise-like or texture-like cipher images after encryption. The histogram of cipher images is uniform and flat, and the information entropy is close to 8. However, these noise-like or texture-like images can easily attract the attention of attackers in the transmission process. Therefore, it is necessary to encrypt plain images into meaningful visual images to enhance the security of image encryption.

The second method mentioned above is visually secure scheme, which usually processes the plain image and then transforms it into another meaningful image through some technology, so that the image is not easy to attract the attention of attackers and has better security when it is transmitted in the public channel. In [47], Lai and Tsai proposed a secret-fragment-visible mosaic image scheme. First, a new method of image similarity measurement is used to select the target image which is most similar to the original image in an image database. Next, the target image and the original image are divided into fixed size blocks. Then similar blocks are found in the target image for each original image block, and they are fitted to the target image. Finally, the secret image is obtained. However, the drawback of this scheme is that a large image database is needed to make the generated image sufficiently similar to the selected target image. To solve this problem, Lee and Tsai [48] divided the original image and the target image into rectangular blocks. Then, according to the similarity criterion based on the color transformations, the original image blocks are used to replace the target image blocks. Finally, the color features of each original image blocks are transformed into the corresponding color features of the target blocks, and a mosaic image similar to the target image is obtained. This scheme can make the choice of target image more free, but the target image needs to be adjusted to the same size as the original image when encrypted. In [49], Bao and Zhou proposed an encryption scheme, which used the existing encryption scheme to encrypt plain images, and then embedded the generated random-like secret images into the reference image to generate meaningful cipher images. However, the cipher images generated by this scheme are four times the plain image, which means that more encryption time, transmission bandwidth and storage space are needed. In [50], Chai *et al.* proposed a meaningful image encryption scheme using compressive sensing technology. First, the plain image is sparsified, then the transformed coefficients are confused by zigzag. Second, the confused image is encrypted into a compressed secret image based on compressive sensing. Third, the secret image is embedded into the carrier image to obtain a visually secure cipher image. In order to resist the CPA, additional hash values of plain images need to be transmitted. In addition, the quality of restored images in [50] depends on the selected carrier image, which makes the selection of images inflexible. In [51], Wang *et al.* proposed a counter mode based on parallel compressive sensing to solve the problem of CPA.

In order to reduce the size of carrier image and avoid reusing the same measurement matrix, a new visually secure scheme based on compressive sensing and reversible color transformation is proposed in this paper. In the compressive sensing stage, a random number is added so that different measurement matrices can be used to encrypt the same image in different order to obtain different results. Thus, it can resist the chosen plaintext attack. The embedding stage is improved on the basis of the method used in [47]. It realizes the flexible selection of carrier image while restoring the quality of the image independent of the carrier image. In addition, the size of the carrier image can not only be equal to that of the plain image, but also allow the selected carrier image to be smaller than that of the plain image. Our contributions are as follows:

(1) The measurement matrix of each column of signal is different, so the problem of reusing measurement matrix is avoided.

(2) In the encryption process, random numbers are added to resist the chosen plaintext attack, and random numbers are not transmitted as keys.

(3) In contrast with the method that all the blocks in the carrier images are completely replaced, the proposed method is new that the partial blocks in the carrier images are replaced, which improves the quality of cipher image and allows the size of carrier images to be smaller than plain image.

(4) Different from the block replacement between two similar images, this paper first attempts to replace the block of the carrier image with a noise-like image block.

The rest of this paper is organized as follows. In Section II, the basic theory about chaotic maps and CS is introduced. Section III describes the proposed images encryption and decryption algorithm. Section IV shows our computer simulations and results. Security and performance analyses are given in Section V and our conclusions are left to the final Section.

II. PRELIMINARIES

A. CHAOTIC MAPS

1) TWO-DIMENSIONAL LOGISTIC-ADJUSTED-SINE MAP

In the encryption scheme of this paper, a chaotic map called Two-dimensional Logistic-adjusted-Sine map (2D-LASM) [52] is adopted. It can obtain more complex structure and larger key space. It is developed on the basis of Logistic map and Sine map. They are defined by Eq. 1 and Eq. 2, respectively.

$$x_{i+1} = 4px_i\left(1 - x_i\right), \quad (1)$$
$$x_{i+1} = s\sin\left(\pi x_i\right), \quad (2)$$

where p and s are parameters. The Logistic map and Sine map are chaotic when $p \in [0.89, 1]$ and $s \in [0.87, 1]$. 2D-LASM is defined as follows

$$\begin{cases} x_{i+1} = \sin(\pi\mu(y_i + 3)x_i(1 - x_i)), \\ y_{i+1} = \sin(\pi\mu(x_{i+1} + 3)y_i(1 - y_i)), \end{cases} \quad (3)$$

where μ is a parameter.

According to the method proposed by Ramasubramanian and Sriram [53], 2D-LASM has chaos behavior when $\mu \in [0.37, 0.38]\bigcup[0.4, 0.42]\bigcup[0.44, 0.93]$, and behaves hyper-chaos when $\mu \in [0.44, 0.93]$.

2) 3D CAT MAP

Another chaotic map used in this paper is 3D cat map [30]. 3D cat map is an extension of classical 2D cat map [54]. Compared with 2D cat map, 3D cat map has better randomness and high sensitivity to control parameters. 3D cat map is defined by

$$X_{i+1} = \begin{bmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+1} \end{bmatrix} = A \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} (\mathrm{mod}\,1), \quad (4)$$

where

$$A = \begin{bmatrix} 1 + a_x a_z b_y & a_z & a_y + a_x a_z + a_x a_y a_z b_y \\ b_z + a_x b_y + a_x a_z b_y b_z & a_z b_z + 1 & a_y a_z + a_x a_y a_z b_y b_z + a_x a_z b_z + a_x a_y b_y + a_x \\ a_x b_x b_y + b_y & b_x & a_x a_y b_x b_y + a_x b_x + a_y b_y + 1 \end{bmatrix}, \quad (5)$$

and a_x,a_y,a_z,b_x,b_y and b_z are positive integers.

In this paper, as a special case, we let $a_x = a_y = a_z = 1$, $b_x = 2$ and $b_y = b_z = 3$. Thus Eq. 4 becomes

$$X_{i+1} = \begin{bmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+1} \end{bmatrix} = A \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} (\mathrm{mod}\,1), \quad A = \begin{bmatrix} 4 & 1 & 5 \\ 15 & 4 & 19 \\ 9 & 2 & 12 \end{bmatrix}, \quad (6)$$

As described in [55], three eigenvalues of A are: $\lambda_1 = 19.27641509$, $\lambda_2 = 0.0806929814$ and $\lambda_3 = 0.6428919346$. The leading Lyapunov characteristic exponent $\lambda_1 > 1$, meaning that Eq. 6 is chaotic.

B. COMPRESSIVE SENSING AND STRUCTURALLY RANDOM MATRIX

Traditional CS directly sampled one-dimensional signal

$$y = \Phi x, \quad (7)$$

where x represents the one-dimensional signal of length N, Φ is the measurement matrix of size $M \times N$, and y is the sampling result of size $M \times N$. Generally, natural signals are non-sparse, while in some transform domains (DCT, DWT) they are sparse. So signal x can be expressed as

$$x = \Psi\theta, \quad (8)$$

where Ψ denotes the $N \times N$ orthogonal matrix, θ denotes the transformation coefficients of x in the Ψ domain, and θ has at most K non-zero values. So Eq. 7 can be rewritten as

$$y = \Phi x = \Phi\Psi\theta = \Theta\theta, \quad (9)$$

for CS construction [37], it is pointed out that if the measurement matrix satisfies restricted isometry property (RIP) [35], x can be reconstructed from y with high probability. In this case, the recovery of signal x can be achieved by solving the following convex optimization problem

$$min \parallel \theta \parallel_1 s.t. \Theta\theta = y, \quad (10)$$

and we can get $x = \Psi\theta$. Solving the above equation can be achieved by Convex optimization algorithm or Greedy algorithm such as Orthogonal Matching Pursuit (OMP).

For a multi-dimensional signal, if it is reconstructed into a vector, it will become very large for the measurement matrix, which will increase the complexity of storage and calculation. To solve this problem, the multi-dimensional signal is reconstructed into two-dimensional signal, and then the two-dimensional signal is sampled sequentially using the same measurement matrix. This method is called parallel compressive sensing (PCS) [56]. Similarly, image reconstruction is also carried out column by column. For two-dimensional image, the size is $N \times N$, and the sampling process of parallel compressive sensing is shown by

$$y_i = \Phi x_i, \tag{11}$$

where $i = 1, 2, \ldots, N$. In [57] it is pointed out that in parallel compressive sensing, CS-based parallel encryption system cannot resist chosen plaintext attack when reusing the same measurement matrix. The reason is that for a fixed measurement matrix, encryption is deterministic. In this paper, when an image is compressed and sampled, different measurement matrices are used for each column of signals, and a random number *rand* is used to control the order of use of measurement matrices. It can be realized that different random numbers can be selected to achieve different encryption results for the same image sampling. Therefore, the parallel sampling of the above image can be expressed as

$$y_i = \Phi_{rand} x_i, \tag{12}$$

where Φ_{rand} is the measurement matrix selected by random number *rand* when signal x_i is sampled.

In order to recover the signal better, it is necessary to design the measurement matrix, which should also satisfy the requirements of good security and low transmission cost. Souyah et al. [58] proposed a new CS matrix framework-Structurally Random Matrix (SRM), which is defined as follows

$$\Phi = \sqrt{\frac{N}{M}} DFR, \tag{13}$$

where $R \in \mathbb{R}^{N \times N}$ is a uniformly random permutation matrix or a diagonal random matrix, and its diagonal entries R_{ii} are Bernoulli random variables with the same distribution $P(R_{ii} = \pm 1) = \frac{1}{2}$. A uniformly random permutation matrix permutes the sampling locations of the signal globally, while a Bernoulli random variables flips the sampling symbol of the signal. $F \in \mathbb{R}^{N \times N}$ is an orthogonal matrix, usually Fast Fourier Transform (FFT), the Discrete Cosine Transform (DCT), the Walsh-Hadamard Transform (WHT), or their block diagonal versions. $D \in \mathbb{R}^{M \times N}$ is a sub-sampling matrix, and a random subset of M rows is selected from the matrix FR of the size of $N \times N$. The scale coefficient $\sqrt{\frac{N}{M}}$ is to normalize the transform.

III. DESCRIPTION OF THE PROPOSED CRYPTOSYSTEM

The algorithm proposed in this paper consists of two stages. In the first stage, the plain image is permutated with the 2D-LASM chaotic sequence, and the Structurally Random Matrix for compressive sensing is constructed by the chaotic sequence generated by the 3D cat map. The compression and encryption are under the framework of parallel compressive sensing. In the second stage, the secret image obtained in the first stage is embedded into the carrier image, and at the same time, the order of the embedding position is encrypted by the 2D-LASM chaotic sequence so as to further improve the security of the scheme. In this paper, we assume that the plain image is denoted as P and the carrier image is denoted as T, the total procedure is shown in Fig. 1.

A. THE ENCRYPTION ALGORITHM

1) STAGE 1: COMPRESSION AND ENCRYPTION

In this part, the plain image is compressed and encrypted to obtain a secret image. Assuming that the size of the plain image P is $N \times N$, the procedure for performing image compression and encryption to generate a secret image Y_s is given as follows:

Step 1 Sparsify the plain image P using discrete wavelet transform (DWT) to obtain a sparse coefficients matrix P_1, and its size is the same as P. Set a threshold, if elements of $P1$ smaller than threshold, change them into zero.

Step 2 Iterate 2D-LASM for $d + N \times N$ times with initial values r_0, s_0 and parameter μ, and discard the first d iterated values to avoid the transient effect. Two chaotic sequences $R = (r_1, r_2, \ldots, r_{N \times N})$, $S = (s_1, s_2, \ldots, s_{N \times N})$ are obtained. Then, sort sequence R and get the position vector $L = l_1, l_2, \ldots, l_{N \times N}$. Next, transform P_1 from $N \times N$ to $1 \times N^2$, and obtain P_2 by $P_2(i) = P_1(l_i)$, $i = 1, 2, \ldots, N \times N$, and finally adjust P_2 from $1 \times N^2$ to $N \times N$.

Step 3 Iterate 3D cat map for N times with the initial values x_0, y_0, z_0 to get three sequences $X = (x_1, x_2, \ldots, x_N)$, $Y = (y_1, y_2, \ldots, y_N)$, $Z = (z_1, z_2, \ldots, z_N)$. Then, create SRM $\varphi_i = Func\_CSRM(M, N, x_i, y_i)$, according to Algorithm 1, and the measurement matrix sequence $\Phi = (\varphi_1, \varphi_2, \ldots, \varphi_N)$ can be obtained for $i = 1, 2, \ldots, N$. Assume the compression ratio of the plain image P is CR, the size of each measurement matrix φ_i in Φ is $M = CR \times N$.

Step 4 Sort the sequence Z and get the position vector Z_{sort}. For parallel compressive sensing, each column of a matrix P_2 denoted as a column vector p_i, is sampled using the measurement matrix φ_i according to Eq. 16, and the measurement value matrix C of size $M \times N$ is obtained,

$$Z_{sort} = sort(Z), \tag{14}$$

$$j = \mathrm{mod}(rand + Z_{sort}(i), N) + 1, \tag{15}$$

$$C_i = \varphi_j p_i, 1 \leq i \leq n, \tag{16}$$

where $sort()$ is the sorting function, $\mathrm{mod}(x, y)$ is the remainder function, C_i represents the column vector of the matrix C. Here, *rand* is a random number which can determine the order of measurement matrices used in each measurement. This random number is employed to make the cipher images totally different from each other even using the same secret key to compress and encrypt a plain image several times. Therefore, the proposed algorithm can well resist chosen plaintext attack.

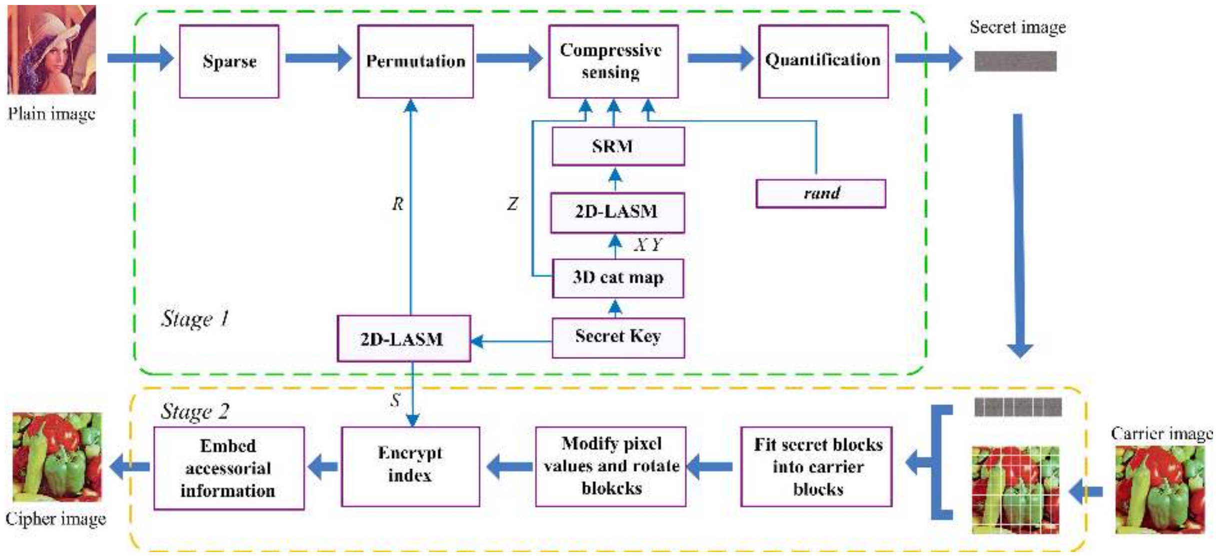


FIGURE 1. **The encryption flowchart of the proposed algorithm.**

Step 5 Quantize the measurement value matrix C to $[0, 255]$ according to Eq. 17, the secret image Y is finally obtained, and its size is $M \times N$,

$$Y = floor(255 \times \frac{C - \min}{\max - \min}), \tag{17}$$

where $floor()$ represents the largest integer not greater than x, min represents the minimum value in matrix C, and max represents the maximum value in matrix C.

2) STAGE 2: EMBEDDING

In this part, the secret image and the carrier image are divided into non-overlapping blocks, and each tile of the secret image is fitted to the carrier image. Since the color characteristics of these two blocks are different from each other, a color transformation [59] is used in [1], [48], [60] in order to convert one image's color characteristics to another. However, this technique of color transformation is not reversible, so Zhang *et al.* [60] modified Lee and Tsai method [48] to be reversible. This reversible color transformation is adopted in this paper. Unlike all the blocks need to be replaced [48], [60], we compressed the plain image in the first stage, and only some of the blocks in the carrier image need to be replaced. Assuming the size of the carrier image T is $m \times n$, the size of m and n is not limited, but it is related to the size of the tile image and determines the quality of the final generated cipher image. The embedding steps are as follows:

Step 1 The secret image Y and the carrier image T are divided into non-overlapping blocks (tile image) of size S_t respectively. So k secret tile images $Y_1, Y_2, \ldots, Y_k$ and l carrier tile images $T_1, T_2, \ldots, T_l$ are obtained. Here, $k \le l$.

Step 2 Calculate the mean and standard deviation (SD) σ_Y, σ_T of each tile image Y_i $(1 \le i \le k)$ and each carrier image T_j $(1 \le j \le l)$, according to Eq. 18 and Eq. 19,

$$\begin{cases} \mu_Y = \frac{1}{n}\sum_{t=1}^{n} p_t \\ \sigma_Y = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(p_t - \mu_Y)^2}, \end{cases} \tag{18}$$

$$\begin{cases} \mu_T = \frac{1}{n}\sum_{t=1}^{n} p_t{'} \\ \sigma_T = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(p_t{'} - \mu_T)^2}, \end{cases} \tag{19}$$

where p_t and p_t' are the pixel values in the secret tile image Y_i and the carrier tile image T_j, μ_Y and μ_T are the mean values of Y_i and T_j, n is the total number of pixels in each tile image.

Step 3 The SD sequences $\sigma_{Y1}, \sigma_{Y2}, \ldots, \sigma_{Yk}$ and $\sigma_{T1}, \sigma_{T2}, \ldots, \sigma_{Tk}$ are sorted in an ascending order separately to get the position vectors $Y_{sort} = \{a_1, a_2, \ldots, a_i, \ldots, a_k\}$ and $T_{sort} = \{b_1, b_2, \ldots, b_j, \ldots, b_l\}$. Make a one-to-one mapping of the elements in Y_{sort} and T_{sort} until the tile images in Y_{sort} are all mapped to the carrier image, resulting in a set of mapped sequences $MS = \{Y_{a_1} \leftrightarrow T_{b_1}, Y_{a_2} \leftrightarrow T_{b_2}, \ldots, Y_{a_k} \leftrightarrow T_{b_k}\}$.

Step 4 According to the mapping sequence set MS, the tile image in the set T_{sort} is replaced with the tile image in the set Y_{sort} until all the images in the set Y_{sort} are replaced, and the cipher image $T1$ of size $m \times n$ is obtained.

Step 5 The pixels in the block Y_{a_i} in the cipher image $T1$ are modified to be more similar to the carrier

Algorithm 1 Construction of the Structurally Random Matrix (CSRM)

Input: The size of the measurement matrix M, N, and chaotic sequence x_i, y_i

Output: Structurally Random Matrix φ_i of size $M \times N$

```
(1): construction of matrix R
k = 1
for i = 1 : N do
  for i = 1 : N do
    if x_k >= 0.5 then
      R(i, j) = -1
      k = k + 1
    else
      R(i, j) = 1
      k = k + 1
    end if
  end for
end for
(2): construction of matrix F
for i = 1 : N - 1 do
  for i = 1 : N - 1 do
    if i == 0 then
      a = sqrt(1/N)
    else
      a = sqrt(2/N)
      F(i + 1, j + 1) = a×cos(j+0.5)×π×i / N
    end if
  end for
end for
(3): construction of matrix D
Y_sort = sort(Y)
D = zeros(N)
k = M
for i = 1 : N do
  D(Y_sort(i), Y_sort(i)) = 1
  k = k - 1
  if k < 1 then
    break
  end if
end for
where sort() is the sorting function.
(4): construction of matrix φ_i
φ_i = sqrt(N/M) DFR
```

image T according to Eq. 20 and Eq. 21. Finally, a new pixels set $Y''_{a_i} = \{p''_1, p''_2, \ldots, p''_n\}$ is obtained,

$$\Delta u = round(\mu_T - \mu_Y), \tag{20}$$

$$p_t'' = p_t + \Delta u, \tag{21}$$

where $round(x)$ is the rounding function, μ_T is the mean of block T_{b_i}, μ_Y is the mean of Y_{a_i}. p_t is the pixel of Y_{a_i}, p''_t is the modified value of the pixel p_t.

Step 6 Each pixel value should be in the range of [0, 255]. Since the operation in Eq. 21 may overflow or underflow the pixel value, it is necessary to modify the pixel value and record the overflow or underflow value. Use *overflow* and *underflow* to record the transformed information, OV_{max} indicates the maximum value of overflow pixel, and UN_{min} indicates the minimum value of underflow pixel. If an overflow or underflow occurs, it needs to modify Δu according to Eq. 22, and the pixel value p is shifted with the modified Δu so as to make all the pixel values in the range of [0, 255],

$$\Delta u = \begin{cases} \Delta u + 255 - OV_{max}, & if\ \Delta u \geq 0 \\ \Delta u - UN_{min}, & if\ \Delta u < 0 \end{cases} \tag{22}$$

Step 7 Rotate the secret block Y_{a_i} into each direction $\theta = 0^o, 90^o, 180^o$ and 270^o, and calculate each root-mean-square error (RMSE) of the block Y_{a_i} with respect to its corresponding carrier block T_{b_i} after the rotation; rotate the block Y_{a_i} into the optimal direction with the smallest RMSE.

Step 8 In order to reduce the length of the embedded index information, $MS = \{Y_{a_1} \leftrightarrow T_{b_1}, Y_{a_2} \leftrightarrow T_{b_2}, \ldots, Y_{a_k} \leftrightarrow T_{b_k}\}$, $a_1, a_2, \ldots, a_k$ are sorted in an ascending order and the corresponding $b_1, b_2, \ldots, b_k$ are change the position too. Thus, the updated $b'_1, b'_2, \ldots, b'_k$ are obtained and the updated one to one map sequence is $MS' = \{Y_1 \leftrightarrow T_{b'_1}, Y_2 \leftrightarrow T_{b'_2}, \ldots, Y_k \leftrightarrow T_{b'_k}\}$. As a result, we only need to record one block index $b'_1, b'_2, \ldots, b'_k$ instead of two indexes $a_1, a_2, \ldots, a_k$ and $b_1, b_2, \ldots, b_k$. In order to improve the security of encryption algorithm, we use chaotic sequence $S = (s_1, s_2, \ldots, s_k)$ to do *xor* operation on the index $b'_1, b'_2, \ldots, b'_k$ according to Eq. 23 to get new values y'_i, and use the new index sequence $y' = (y'_1, y'_2, \ldots, y'_k)$ as the encrypted index information,

$$y'_i = xor(s_i, b_i), \tag{23}$$

where $xor(x, y)$ is exclusive *OR* operation, $1 \leq i \leq k$.

Step 9 The required information to recover the secret image includes: (1) random number *rand*, (2) minimum quantitative value *min*, (3) maximum quantitative value *max*, (4) block index y', (5) *overflow*/*underflow* values, (6) Δu, (7) rotation direction θ. Encrypted this additional information (AI) into sequence and embed it into image $T1$ by the reversible contrast mapping (RCM) scheme [61].

B. THE DECRYPTION ALGORITHM

The decryption process is the inverse operation of the encryption process, which includes two stages: the first stage is

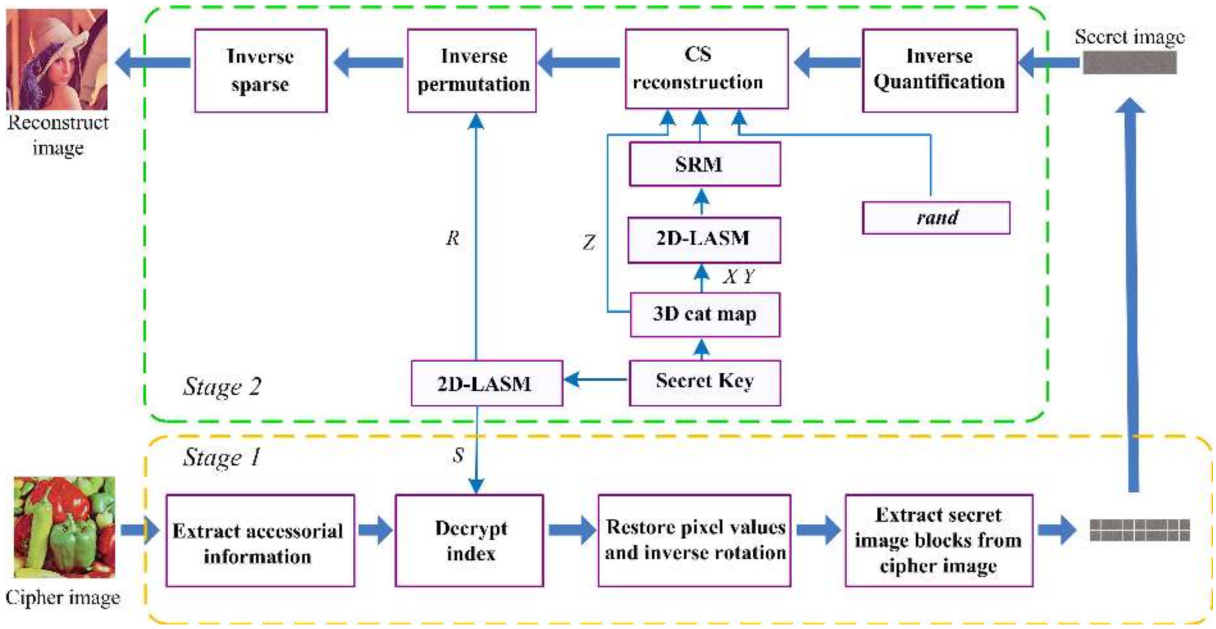


FIGURE 2. The decryption flowchart of the proposed algorithm.

to extract accessorial information from the cipher image and restore the secret image; the second stage is to reconstruct the plain image using the secret image. The decryption flow chart is shown in Fig. 2. Suppose the cipher image is $T1$.

1) STAGE 1: EXTRACTING THE SECRET IMAGE

Step 1 Extract additional information (AI) from the cipher image by the RCM scheme [61] and decompress the sequence to obtain random number *rand*, minimum quantitative value *min*, maximum quantitative value *max*, block index y', *overflow/underflow* values, Δu, rotation direction θ.

Step 2 Through the set y' and the chaotic sequence $S = (s_1, s_2, \ldots, s_k)$ to obtain $b'_1, b'_2, \ldots, b'_k$ according to Eq. 24. Sequence $Y1 = \{Y1_1, Y1_2, \ldots, Y1_k\}$ is obtained by extracting blocks from cipher images according to $b'_1, b'_2, \ldots, b'_k$,

$$b'_i = xor(s_i, y'_i). \tag{24}$$

Step 3 According to the block rotation information θ, rotate each transformed block in $Y1$ in the anti-direction and $Y2$ is obtained.

Step 4 Modify the pixels in $Y2$ according to *overflow/underflow* values and Δu, and $Y3$ is obtained.

Step 5 Merge blocks in $Y3$ to get the secret image, denoted as D.

2) STAGE 2: RECONSTRUCTING THE PLAIN IMAGES

After obtaining the secret image D, the plain image is restored by reconstruction. The steps are as follows:

Step 1 The secret image D is inversely quantized according to Eq. 25, and the image $D1$ is obtained,

$$D1 = \frac{D \times (\max - \min)}{255} + \min. \tag{25}$$

Step 2 The SRM $\Phi = (\varphi_1, \varphi_2, \ldots, \varphi_n)$ is calculated by key and Algorithm 1.

Step 3 The orthogonal matching pursuit (OMP) algorithm is applied to each column of $D1$ to obtain the matrix $D2$,

$$D2_i = OMP(D1_i, \varphi_j). \tag{26}$$

Step 4 Adjust $D2$ to $1 \times N^2$, the matrix D' is obtained by inverse permutation according to Eq. 27 of the chaotic sequence $D = (r_1, r_2, \ldots, r_{N\times N})$ of Eq. 3, and the reconstructed image D' is obtained by reshaping the matrix D' to $N \times N$,

$$D'_i(r(i)) = D2(i). \tag{27}$$

IV. SIMULATION RESULTS

For the image encryption algorithm proposed in this paper, it should be able to encrypt different types of images into another selected image. In addition, only the correct key can restore the image. In this section, the image encryption algorithm is simulated. The simulation is performed on MATLAB R2014b, which runs on a personal computer with 2.20 GHz CPU and 8 GB memory. Fig. 3 shows the simulation results of plain image and carrier image are both gray-scale or color images, in which the size of gray-scale or color images includes 256×256 and 512×512, respectively. For color images, we can use the same key to encrypt red, green and

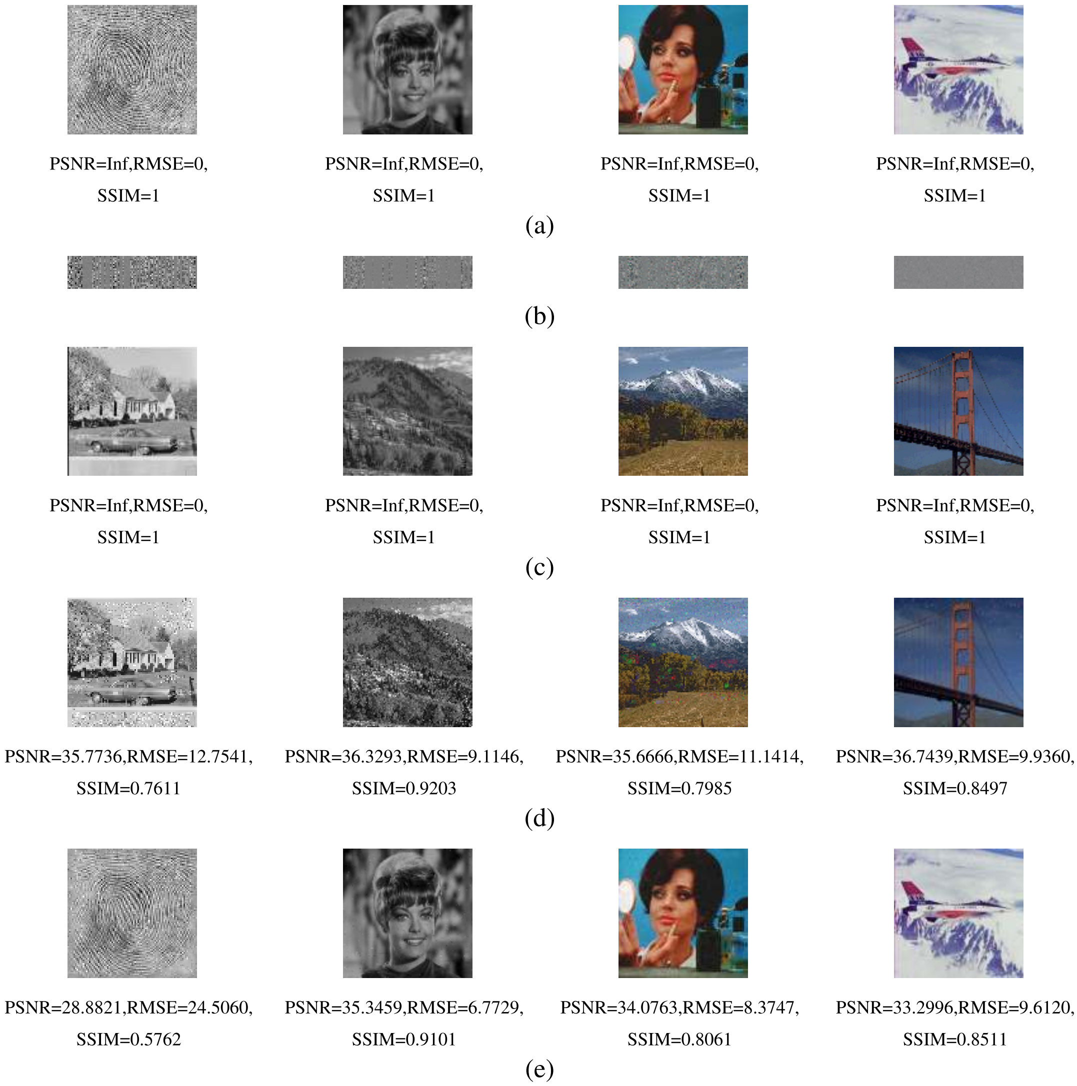


FIGURE 3. Simulation results: (a) plain images; (b) secret images; (c) carrier images; (d) cipher images; (e) decrypted images.

blue channels respectively. In addition, this section also simulates the results of different carrier image sizes when the plain image size is fixed. The results are shown in Fig. 4 and Fig. 5, respectively. Where plain images are 512×512 gray images and 512×512 color images, and carrier images are 512×512 gray image, 512×512 color image, 384×512 gray image, 384×512 color image, 384×384 gray image and 384×384 color image. The plain image and the carrier image in Fig. 4 are both gray-scale images. The plain image and the carrier image in Fig. 5 are both color images. It is obvious that the plain image which is encrypted by proposed scheme has similar appearance with the carrier image, so this scheme has better encryption performance. In addition, the size of the cipher images Fig. 4 (e) and (g) and Fig. 5 (e) and (g) are smaller than the plain images, so the scheme can reduce the bandwidth in transmission.

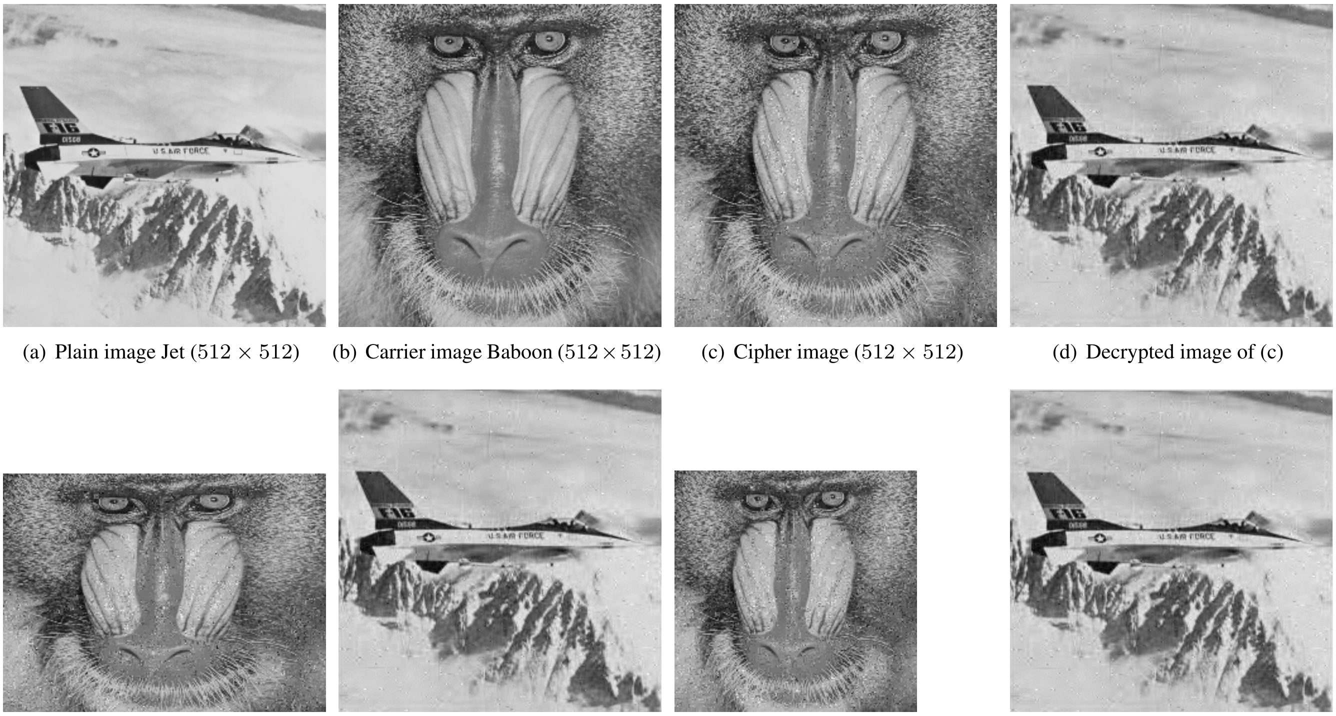
V. SECURITY ANALYSIS

Several experiments and different kinds of security analysis methods are performed to evaluate the robustness of the proposed algorithm in these section.

A. KEY SPACE

The key space refers to the total number of different keys that can be used in the cryptosystem, which reflects the ability of the cyptosystem against a brute-force attack. In this paper, the secret key includes the initial chaotic values of 2D-LASM system (r_0, s_0) and 3D cat map (x_0, y_0, z_0). If the accuracy of the computer is 10^{-15}, the key space of the proposed image encryption system is

$$\begin{aligned} \text{key space} &= 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \times 10^{15} \\ &= 10^{75} > 2^{249}, \end{aligned} \tag{28}$$



(a) Plain image Jet (512 × 512) (b) Carrier image Baboon (512 × 512) (c) Cipher image (512 × 512) (d) Decrypted image of (c)

(e) Cipher image (384 × 512) (f) Decrypted image of (e) (g) Cipher image (384 × 384) (h) Decrypted image of (g)

FIGURE 4. Different size of gray-scale carrier image.

(a) Plain image Lena (512 × 512) (b) Carrier image Peppers (512 × 512) (c) Cipher image (512 × 512) (d) Decrypted image of (c)

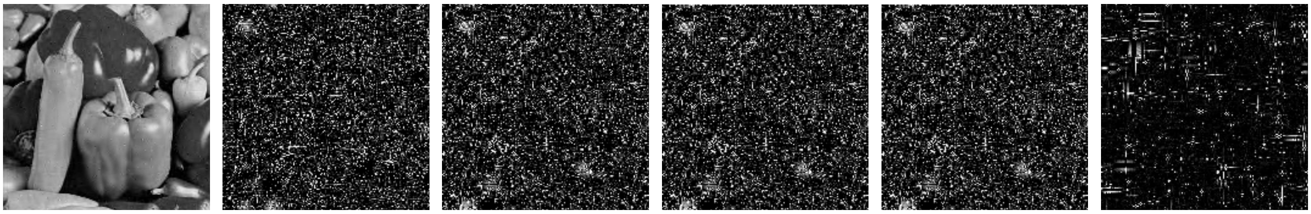
(e) Cipher image (384 × 512) (f) Decrypted image of (e) (g) Cipher image (384 × 384) (h) Decrypted image of (g)

FIGURE 5. Different size of color carrier image.

Therefore, the key space of the scheme is large enough to resist brute-force attack. Table 1 shows the key space of the proposed algorithm and others.

B. KEY SENSITIVITY

The key sensitivity is an important criteria for evaluating a good image cipher. One aspect of key sensitivity for a



(a) The correct key (b) $r_0 + 10^{-15}$ (c) $d_0 + 10^{-15}$ (d) $x_0 + 10^{-15}$ (e) $y_0 + 10^{-15}$ (f) $z_0 + 10^{-15}$

FIGURE 6. Key sensitivity results in decryption process. (a) the correct key; (b) - (f) the modified key r_0,s_0,x_0,y_0,z_0, respectively.

TABLE 1. The key space of proposed algorithm and other different schemes.

Scheme	Proposed	Ref. [50]	Ref. [51]	Ref. [55]
Key space	10^{75}	$37^2 \times 2^{296}$	10^{75}	2^{235}

secure cipher is the failure of restoring the correct plain image from cipher image even if the key is changed a little. On the other hand, a small changing in the encryption key must result in extremely different cipher images. To test the high sensitivity of the proposed cipher to the modification of the secret key, the following tests are carried out.

Test 1: key sensitivity for decryption

In this part, the key sensitivity of the algorithm is analyzed. The plain image is Peppers (512 × 512) and the carrier image is Lena (512 × 512). Other parameters used in this scheme are as follows: $rand = 37$, $CR = 0.25$, threshold = 50, $S_t = 8 \times 8$, initial value of 2D-LASM system $r_0 = 0.655477890177557$, $s_0 = 0.171186687811562$, initial value of 3D cat map system $x_0 = 0.6557406956587$, $y_0 = 0.757740130578333$, $z_0 = 0.392227019534168$. We use the modified key to decrypt the cipher image. The result is shown in Fig. 6. Fig. 6 (a) is the decrypted image with the correct key, and Fig. 6 (b) - (f) are the decrypted images with the modified key. The results show that only with the correct key we can decrypt the correct image. Therefore, whether in the process of encryption or decryption, the proposed algorithm is very sensitive to the key.

In our scheme, the generation of cipher image is also related to the selected random number *rand*. If the number *rand* is incorrect, the image cannot be correctly decrypted. Only with the correct random number *rand* the image can be correctly decrypted. The results are shown in Fig. 7. Fig. 7 (a) is the plain image, Fig. 7 (b) is the decrypted image obtained using $rand = 37$, and Fig. 7 (c) is an image decrypted using $rand = 128$.

Test 2: key sensitivity for encryption

In the encryption process, we add r_0, s_0, x_0, y_0 and z_0 to 10^{-15} respectively, and keep the other values unchanged when modifying each value. In order to evaluate the difference between the correct secret image and the modified secret image after key modification, number of pixel change

(a) Plain image (b) $rand = 37$ (c) $rand = 128$

FIGURE 7. Decryption results with different number *rand*. (a) plain image; (b) *rand* = 37; (c) *rand* = 128.

rate (NPCR) is adopted, and its calculation formula is defined as follows,

$$NPCR = \frac{1}{M \times N} \sum_{i}^{M} \sum_{j}^{N} D(i,j) \times 100\%, \tag{29}$$

$$D(i,j) = \begin{cases} 0, & if\ S_1(i,j) = S_2(i,j) \\ 1, & if\ S_1(i,j) \neq S_2(i,j), \end{cases} \tag{30}$$

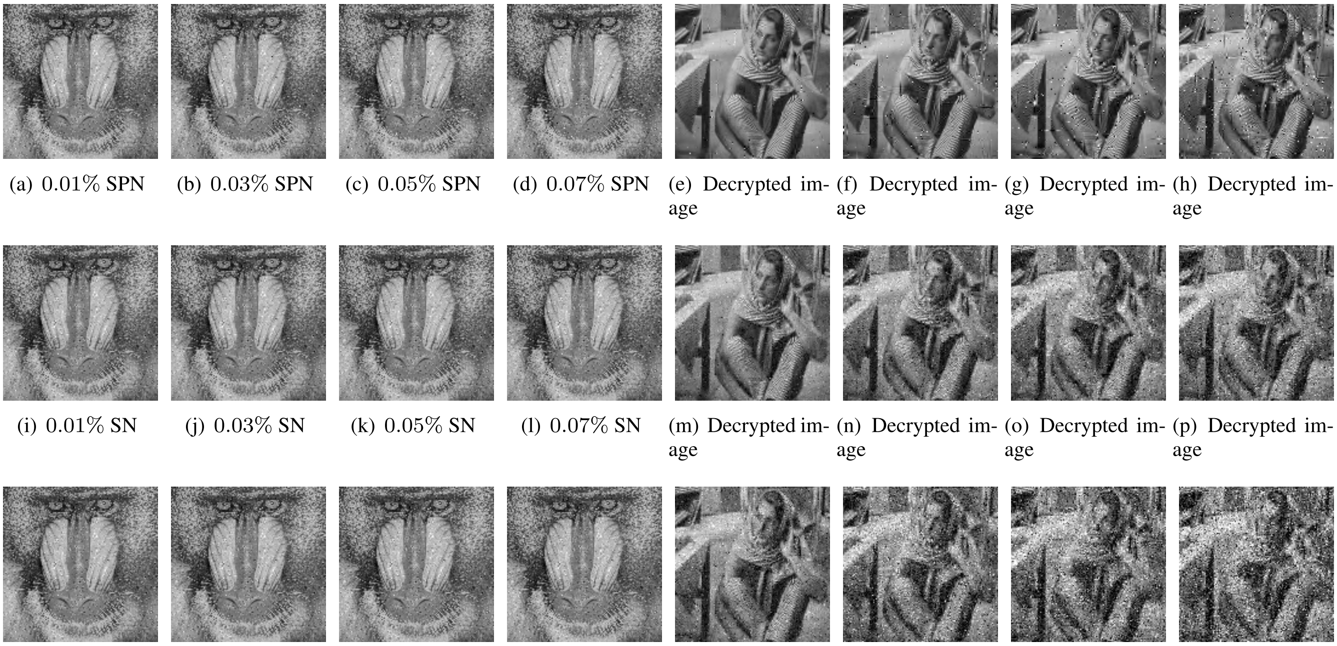
where $M \times N$ is the size of image. S_1 and S_2 are the secret images and modified secret image, respectively. Fig. 8 shows the difference between the secret image obtained by the correct key and the secret image obtained by the modified r_0, s_0, x_0, y_0 and z_0, respectively. It can be seen that after slightly modifying the key, most of the pixels of the secret image are changed, which shows that the key of the scheme has high sensitivity in encryption.

C. DATA LOSS AND NOISE ATTACKS

Data loss and noise interference are inevitable in digital transmission channels. Therefore, if an image encryption algorithm is sensitive to the noise, errors in the encrypted images may produce numerous errors in recovered image which would make the reconstruction of the secret image completely fail. So, it is important that a good image encryption algorithm should be robust during transmission to resist data loss and noise attacks. In this part, we will verify the capability of the proposed scheme to resist the noise attack and data loss attack. In the simulation, plain image Barbara (512 × 512) and carrier image Baboon (512 × 512) are chosen to test. The other parameters are the same as Section V-B.

(a) Secret image (b) NPCR=97.5845% (c) NPCR=95.6680% (d) NPCR=97.5845% (e) NPCR=95.6680% (f) NPCR=98.8021%

FIGURE 8. Key sensitivity analysis in the encryption process. (a) cipher image; (b) difference between correct key and modified r_0; (c) difference between correct key and modified s_0; (d) difference between correct key and modified x_0; (e) difference between correct key and modified y_0; (f) difference between correct key and modified z_0.



(a) 0.01% SPN (b) 0.03% SPN (c) 0.05% SPN (d) 0.07% SPN (e) Decrypted image (f) Decrypted image (g) Decrypted image (h) Decrypted image

(i) 0.01% SN (j) 0.03% SN (k) 0.05% SN (l) 0.07% SN (m) Decrypted image (n) Decrypted image (o) Decrypted image (p) Decrypted image

(q) 0.01% GN (r) 0.03% GN (s) 0.05% GN (t) 0.07% GN (u) Decrypted image (v) Decrypted image (w) Decrypted image (x) Decrypted image

FIGURE 9. Noise attack. (a) - (d) cipher image with SPN intensities 0.01%, 0.03%, 0.05% and 0.07%; (e) - (h) decrypted image of (a) - (d); (i) - (l) cipher image with SN 0.01%, 0.03%, 0.05%, and 0.07%; (m) - (p) decrypted image of (i) - (l); (q) - (t) cipher image with GN 0.01%, 0.03%, 0.05% and 0.07%; (u) - (x) decrypted image of (q) - (t).

1) NOISE ATTACK

In order to evaluate the performance of the proposed scheme to resist noise attack, we apply three types of different noises including Salt&Pepper Noise (SPN), Speckle Noise (SN), and Gauss Noise (GN) to the final cipher image. The simulation results are shown in Fig. 9. Firstly, we add SPN with density of 0.01%, 0.03%, 0.05% and 0.07% to the cipher image. The contaminated image and the decrypted image are shown in Fig. 9 (a) - (h). Then, SN with the intensity of 0.01%, 0.03%, 0.05%, and 0.07%, respectively, are added to the cipher image, and the results are shown in Fig. 9 (i) - (p). Finally, GN with the intensity of 0.01%, 0.03%, 0.05% and 0.07% are added to the cipher image. Fig. 9 (q) - (x) are contaminated image and corresponding decrypted image. The simulation results show that the recovered images could be recognized without doubt in these cases, so the proposed algorithm is robust to noise attack.

2) DATA LOSS

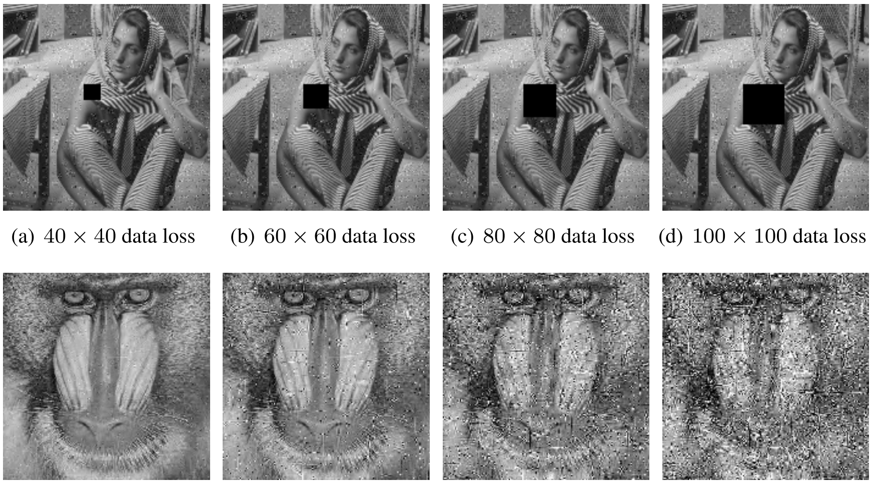
In this section, we will test the impact of data loss on the recovered image. There are four different sizes of data loss in cipher images, as shown in Fig. 10 (a) - (d). The corresponding decrypted image is shown in Fig. 10 (e) - (h). When the size of data loss is from 40×40 to 100×100, the quality of the recovered image decreases gradually, but the recovered image is still readable, so it can be concluded that the proposed algorithm is able to resist a certain degree of data loss attack.

D. INFORMATION ENTROPY ANALYSIS

Information entropy is used to measure the randomness of information. The calculation of information entropy is defined by

$$H(m) = -\sum_{i=0}^{2^n-1} P(m_i)\log_2 P(m_i), \tag{31}$$

where $P(m_i)$ denotes the probability of symbol m_i, n denotes the length of the pixel value in bit. For a random gray image with 256 gray levels, $n = 8$, and the ideal information entropy is 8. In the experiment, several images and corresponding cipher images were tested. The results of information entropy are listed in Table 2. It can be seen that the information entropy of the carrier image and its corresponding cipher



(a) 40 × 40 data loss (b) 60 × 60 data loss (c) 80 × 80 data loss (d) 100 × 100 data loss

(e) Decrypted image (f) Decrypted image (g) Decrypted image (h) Decrypted image

FIGURE 10. Data loss attack. (a) - (d) cipher image with data loss 40 × 40, 60 × 60, 80 × 80, 100 × 100, respectively; (e) - (h) decrypted image of (a) - (d).

TABLE 2. Information entropy.

Plain image	Carrier image	Entropy		
		Plain image	Carrier image	Cipher image
Brain (256 × 256)	Cameraman (256 × 256)	4.8215	7.0097	7.3239
Barbara (512 × 512)	Girl (512 × 512)	7.6321	7.0428	7.2931
Lena (512 × 512)	Goldhill (512 × 512)	7.4456	7.4778	7.5340

image are close to each other, which indicates that the randomness of the carrier image is not obviously destroyed.

E. HISTOGRAM ANALYSIS

Histogram can reflect the distribution of image pixel intensity. Generally speaking, the histograms of plain images have specific patterns, and opponents can use them to capture some information by statistical methods. In order to effectively cover the original information, it is necessary to uniformly distribute the histograms of encrypted images. However, for meaningful image encryption schemes, the histogram of the cipher image should be similar to the histogram of the carrier image. We have simulated Jet (512 × 512), Baboon (512 × 512), Lena (512 × 512), Peppers (512 × 512). The results are shown in Fig. 11, where (a), (e) are the carrier image, (b), (f), (g), (h) are the corresponding carrier image histogram, (c), (i) are the cipher image, (d), (j), (k), (l) are the corresponding cipher image histogram. It can be seen that the pixel intensity distribution of the cipher image is similar to that of the histogram of the carrier image.

F. DIFFERENT RESULTS OF BLOCK SIZE

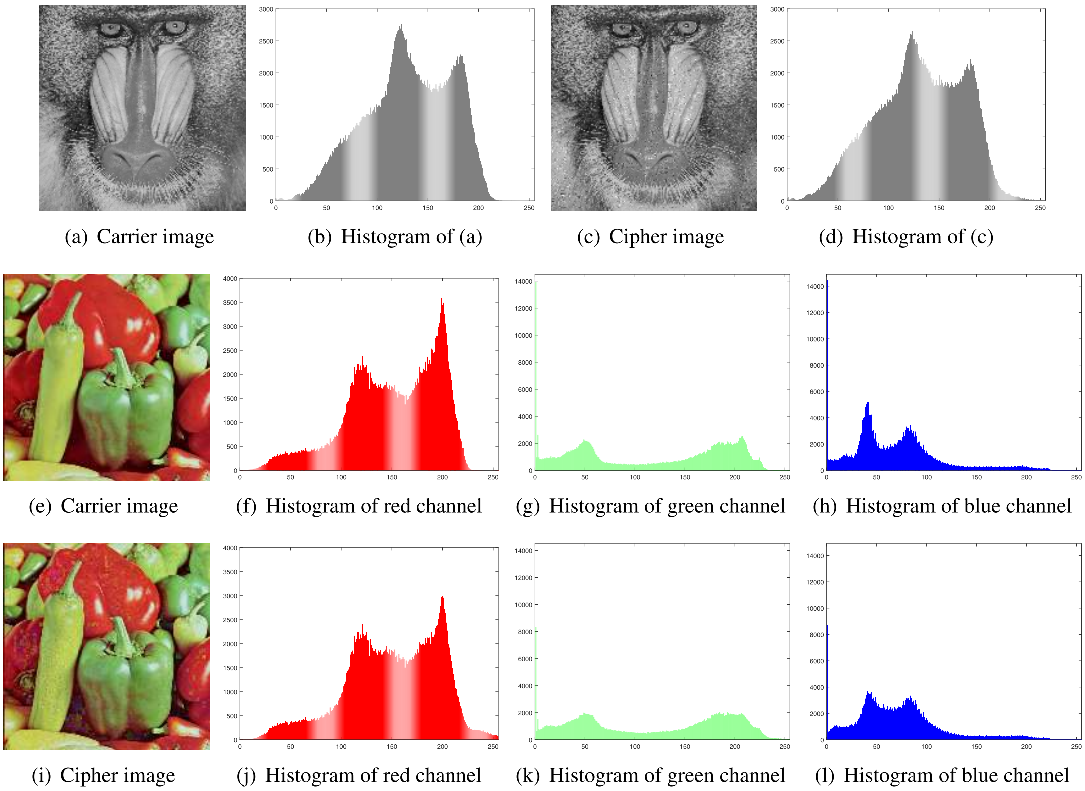
Different size of blocks directly affects the visual results of cipher images. We tested the cipher images with block sizes of 4 × 4, 8 × 8, 16 × 16 and 32 × 32 respectively. The plain image is Lena (512 × 512), and the carrier image is Hsewoods (512 × 512). The result is shown in the Fig. 12. Fig. 13 shows square enlargement area of the cipher image. The structural similarity index (SSIM) of different cipher images are shown in the Table 3, and the peak signal to noise ratio (PSNR) is shown in the Table 4. The results show that the similarity between cipher images and carrier images decreases with the increase of block size. It can be seen that when the block size is small, the cipher image shows better visual results, but more accessorial information is needed. With the increase of

TABLE 3. SSIM values of cipher images.

Carrier image	Block size			
	4 × 4	8 × 8	16 × 16	32 × 32
Hsewoods (512 × 512)	0.9709	0.8609	0.8344	0.8239

TABLE 4. PSNR values of cipher images.

Carrier image	Block size			
	4 × 4	8 × 8	16 × 16	32 × 32
Hsewoods (512 × 512)	36.5945	35.0562	34.3234	33.7372

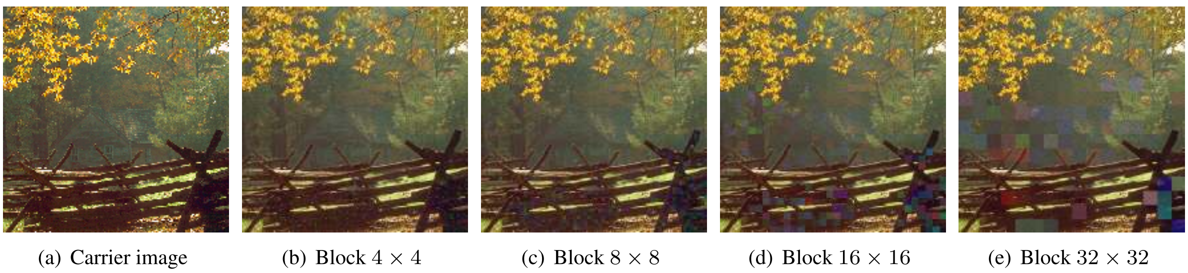


(a) Carrier image (b) Histogram of (a) (c) Cipher image (d) Histogram of (c)

(e) Carrier image (f) Histogram of red channel (g) Histogram of green channel (h) Histogram of blue channel

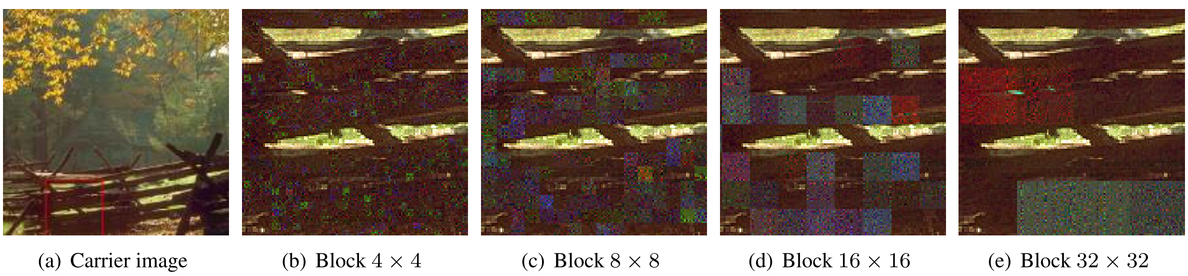
(i) Cipher image (j) Histogram of red channel (k) Histogram of green channel (l) Histogram of blue channel

FIGURE 11. Histograms analysis. (a) gray carrier image "Baboon"; (b) histograms of (a); (c) cipher image; (d) histograms of (c); (e) color image "Peppers"; (f) - (h) the histogram of red, green, blue of (e), respectively; (i) color cipher image; (j) - (l) the histogram of red, green, blue of (i), respectively.



(a) Carrier image (b) Block 4×4 (c) Block 8×8 (d) Block 16×16 (e) Block 32×32

FIGURE 12. Different results of block size. (a) carrier image "Hsewoods"; (b) size of block 4 × 4; (c) size of block 8 × 8; (d) size of block 16 × 16; (e) size of block 32 × 32.



(a) Carrier image (b) Block 4×4 (c) Block 8×8 (d) Block 16×16 (e) Block 32×32

FIGURE 13. Square enlargement area of the cipher image. (a) carrier image; (b) 4 × 4; (c) 8 × 8; (d) 16 × 16; (e) 32 × 32.

the block size, the block effect of the cipher image becomes more and more obvious, and the visual results becomes worse and worse.

G. DIFFERENT COMPRESSION RATIOS

Different compression ratios will produce different sizes of secret image, and the size of secret image will affect

(a) PSNR=inf, RMSE=0, SSIM=1 (b) PSNR=35.6322, RMSE=10.7761, SSIM=0.9007 (c) PSNR=32.1233, RMSE=17.2425, SSIM=0.7834 (d) PSNR=30.1842, RMSE=23.3649, SSIM=0.6354

FIGURE 14. Different results of compression ratio. (a) carrier image "Utahmtn"; (b) $CR = 0.25$; (c) $CR = 0.5$; (d) $CR = 0.75$.

TABLE 5. Comparison of the PSNR values of cipher images.

Plain image	Carrier image	PSNR(dB)		
		Ref. [50]	Ref. [51]	Proposed
Brain (256×256)	Cameraman (256×256)	24.8700	34.8967	35.3634
Lena (512×512)	Peppers (512×512)	18.5136	32.3513	35.1347
Jet (512×512)	Baboon (512×512)	23.3967	37.1058	36.4906
Girl (512×512)	Goldhill (512×512)	28.2318	36.1125	36.2169
Barbara (512×512)	Bridge (512×512)	25.2321	35.5629	36.1070
Average		24.0488	35.2058	35.8625

the quality of cipher image. This section tests the quality of cipher images at different compression ratios, including SSIM, PSNR and RMSE. The plain image is Tiffany (512×512), and the carrier image is Utahmtn (512×512). Compression ratio CR is 0.25, 0.5 and 0.75, respectively. The experimental results are shown in the Fig. 14. As can be seen from the Fig. 14, the quality of cipher image decreases with the increasing compression ratio. This is caused by the enlargement of secret images that need to be embedded.

H. ABILITY OF RESISTING CHOSEN-PLAINTEXT ATTACK

Chosen plaintext attack (CPA) is a powerful attack in cryptanalysis. As pointed in [62], some CS-based ciphers [63], [64] cannot resist chosen-plaintext attack when a single measurement matrix is used to encrypt multiple signals. To make the chosen-plaintext attack more difficult, our scheme generates multiple different measurement matrices in order to prevent reusing the measurement matrix and a random number is employed to disrupt the order of measurement matrices. As a result, the obtained cipher images (secret images) are totally different from each other even using the same secret key to encrypt a plain-image several times. Therefore, the proposed scheme can well withstand the chosen-plaintext attack.

I. TIME COMPLEXITY ANALYSIS

Time complexity is an important factor affecting encryption algorithm. The time complexity of the proposed scheme includes the time of CS process and the time of embedded process. The time complexity of CS depends on the size of the measurement matrix and the size of the plain image. If the plain image is $N \times N$ and the measurement matrix is $M \times N$, the time complexity is $O(MN^2)$. Since the generation of the measurement matrix can be performed in parallel, the theoretical minimum complexity is $O(MN)$. In the embedding stage, the time complexity depends on the number of blocks in the carrier image, and the other steps are linear. Assuming that the carrier image is divided into n blocks and sorted by standard deviation, the minimum complexity of the embedding stage based on sorting algorithm is as follows $O(n + n \log_2 n)$.

J. COMPARISON WITH OTHER ENCRYPTION ALGORITHMS

Firstly, from the point of view of cipher image, the size of cipher image generated based on this scheme can not only be equal to that of plain image, but also smaller than that of plain image. The size of the cipher image in the image encryption scheme in [50], [51] is equal to that of the plain image, while the cipher image in the image encryption scheme in [49] is larger than that of the plain image. The size of cipher image is related to encryption time and transmission bandwidth, so smaller size can improve encryption efficiency and transmission speed.

Then, the imperceptibility of the cipher image and the quality of image reconstruction are listed in Table 5. It can be seen that the average PSNR of the cipher image is better than that of [50], [51]. The comparative results of the

TABLE 6. Comparison of the PSNR values of the reconstruction images.

Carrier image	PSNR(dB)		
	Ref. [50]	Ref. [51]	Proposed
Lena (512 × 512)	28.4808	28.4422	32.4225
Bridge (512 × 512)	18.4706	28.4422	32.4225
Girl (512 × 512)	12.8477	28.4422	32.4264
Peppers (512 × 512)	12.3628	28.4422	32.4225
Average	18.0405	28.4422	32.4235

TABLE 7. Comparison of the MSSIM values of the reconstruction images.

Carrier image	MSSIM		
	Ref. [50]	Ref. [51]	Proposed
Lena (512 × 512)	0.8142	0.8128	0.8885
Bridge (512 × 512)	0.3210	0.8128	0.8885
Girl (512 × 512)	0.1083	0.8128	0.8883
Peppers (512 × 512)	0.0782	0.8128	0.8885
Average	0.3304	0.8128	0.8885

reconstructed images are listed in Tables 6 and 7. Among them, Barbara (512 × 512) is chosen as the plain image, and the carrier images are Lena (512 × 512), Bridge (512 × 512), Girl (512 × 512) and Peppers (512 × 512). It can be seen that the reconstructed image in [50] is greatly affected by the carrier image, while [51] is relatively small. For the proposed scheme, the reconstructed performance is better than [50], [51].

VI. CONCLUSION

In this paper, an image encryption scheme based on compressive sensing and reversible color transformation is proposed. In the stage of CS, different measurement matrices can be used to measure each column of the image signal by choosing a random number. For the same image, different random numbers can be selected to obtain different secret images even if the secret key is unchanged. With the help of CS, the stage of embedding allows the size of the carrier image to be smaller than the plain image when the method of block pairing and replacement is applied, so no additional bandwidth is needed in transmission. The simulation results and security analysis show that the scheme has large key space, high key sensitivity and good robustness against common attacks.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable suggestions to improve the quality of this paper.

REFERENCES

[1] C. Wang, B. Zhang, K. Ren, and J. M. Roveda, "Privacy-assured outsourcing of image reconstruction service in cloud," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 1, pp. 166–177, Jun. 2013.

[2] Y. Li, C. Wang, and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Opt. Lasers Eng.*, vol. 90, pp. 238–246, Mar. 2017.

[3] L. Xu, X. Gou, Z. Li, and J. Li, "A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion," *Opt. Lasers Eng.*, vol. 91, pp. 41–52, Apr. 2017.

[4] M. Sundararajan, S. Venkatraj, and S. Anbazhagan, "Image encryption scheme using 2D hyper-chaos," *J. Comput. Theor. Nanosci.*, vol. 16, no. 4, pp. 1560–1562, 2019.

[5] M. Essaid, I. Akharraz, A. Saaidi, and A. Mouhib, "A novel image encryption scheme based on permutation/diffusion process using an improved 2D chaotic system," in *Proc. Int. Conf. Wireless Technol., Embedded Intell. Syst. (WITS)*, Apr. 2019, pp. 1–6.

[6] L. Huang, S. Cai, M. Xiao, and X. Xiong, "A simple chaotic map-based image encryption system using both plaintext related permutation and diffusion," *Entropy*, vol. 20, no. 7, p. 535, 2018.

[7] W. Zhang, Z. Zhu, and H. Yu, "A symmetric image encryption algorithm based on a coupled logistic–Bernoulli map and cellular automata diffusion strategy," *Entropy*, vol. 21, no. 5, p. 504, 2019.

[8] P. Ping, F. Xu, and Z. J. Wang, "Image encryption based on non-affine and balanced cellular automata," *Signal Process.*, vol. 105, pp. 419–429, Dec. 2014.

[9] Y. Wang, Y. Zhao, Q. Zhou, and Z. Lin, "Image encryption using partitioned cellular automata," *Neurocomputing*, vol. 275, pp. 1318–1332, Jan. 2017.

[10] Y. Su, Y. Wo, and G. Han, "Reversible cellular automata image encryption for similarity search," *Signal Process., Image Commun.*, vol. 72, pp. 134–147, Mar. 2019.

[11] Z. Tang, Y. Yang, S. Xu, C. Yu, and X. Zhang, "Image encryption with double spiral scans and chaotic maps," *Secur. Commun. Netw.*, vol. 2019, Jan. 2019, Art. no. 8694678.

[12] R.-J. Chen and S.-J. Horng, "Novel SCAN-CA-based image security system using SCAN and 2-D von Neumann cellular automata," *Signal Process., Image Commun.*, vol. 25, no. 6, pp. 413–426, Jul. 2010.

[13] A. S. Mahmood and M. S. M. Rahim, "Novel method for image security system based on improved SCAN method and pixel rotation technique," *J. Inf. Secur. Appl.*, vol. 42, pp. 57–70, Oct. 2018.

[14] L. Liu, Q. Zhang, and X. Wei, "A RGB image encryption algorithm based on DNA encoding and chaos map," *Comput. Electr. Eng.*, vol. 38, no. 5, pp. 1240–1248, Sep. 2012.

[15] H. Liu, X. Wang, and A. Kadir, "Image encryption using DNA complementary rule and chaotic maps," *Appl. Soft. Comput.*, vol. 12, no. 5, pp. 1457–1466, 2012.

[16] X. Wei, L. Guo, Q. Zhang, J. Zhang, and S. Lian, "A novel color image encryption algorithm based on DNA sequence operation and hyper-chaotic system," *J. Syst. Softw.*, vol. 85, no. 2, pp. 290–299, 2012.

[17] R.-G. Zhou, Q. Wu, M.-Q. Zhang, and C.-Y. Shen, "Quantum image encryption and decryption algorithms based on quantum image geometric transformations," *Int. J. Theor. Phys.*, vol. 52, no. 6, pp. 1802–1817, 2013.

[18] Y.-G. Yang, J. Xia, X. Jia, and H. Zhang, "Novel image encryption/decryption based on quantum Fourier transform and double phase encoding," *Quantum Inf. Process.*, vol. 12, no. 11, pp. 3477–3493, 2013.

[19] A. Akhshani, A. Akhavan, S. C. Lim, and Z. Hassan, "An image encryption scheme based on quantum logistic map," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, no. 12, pp. 4653–4661, 2012.

[20] Y.-G. Yang, Q.-X. Pan, S.-J. Sun, and P. Xu, "Novel image encryption based on quantum walks," *Sci. Rep.*, vol. 5, p. 7784, Jan. 2015.

[21] G. Bhatnagar, Q. M. J. Wu, and B. Raman, "Discrete fractional wavelet transform and its application to multiple encryption," *Inf. Sci.*, vol. 223, pp. 297–316, Feb. 2013.

[22] X. Li, X. Meng, X. Yang, Y. Wang, Y. Yin, X. Peng, W. He, G. Dong, and H. Chen, "Multiple-image encryption via lifting wavelet transform and XOR operation based on compressive ghost imaging scheme," *Opt. Lasers Eng.*, vol. 102, pp. 106–111, Mar. 2018.

[23] H. Wang, D. Xiao, X. Chen, and H. Huang, "Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map," *Signal Process.*, vol. 144, pp. 444–452, Mar. 2018.

[24] G. Ye and X. Huang, "Spatial image encryption algorithm based on chaotic map and pixel frequency," *Sci. China Inf. Sci.*, vol. 61, no. 5, 2018, Art. no. 058104.

[25] Z. Hua, S. Yi, and Y. Zhou, "Medical image encryption using high-speed scrambling and pixel adaptive diffusion," *Signal Process.*, vol. 144, pp. 134–144, Mar. 2018.

[26] M. Kaur and V. Kumar, "Efficient image encryption method based on improved Lorenz chaotic system," *Electron. Lett.*, vol. 54, no. 9, pp. 562–564, 2018.

[27] S. F. Raza and V. Satpute, "A novel bit permutation-based image encryption algorithm," *Nonlinear Dyn.*, vol. 95, no. 2, pp. 859–873, 2019.

[28] Y. Luo, J. Yu, W. Lai, and L. Liu, "A novel chaotic image encryption algorithm based on improved baker map and logistic map," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 22023–22043, 2019.

[29] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.

[30] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons Fractals*, vol. 21, pp. 749–761, Jul. 2004.

[31] J.-X. Chen, Z.-L. Zhu, C. Fu, H. Yu, and L.-B. Zhang, "A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism," *Commun. Nonlinear Sci.*, vol. 20, no. 3, pp. 846–860, Mar. 2015.

[32] G. Ye, C. Pan, X. Huang, and Q. Mei, "An efficient pixel-level chaotic image encryption algorithm," *Nonlinear Dyn.*, vol. 94, no. 1, pp. 745–756, Oct. 2018.

[33] Z. Hua, F. Jin, B. Xu, and H. Huang, "2D Logistic-Sine-coupling map for image encryption," *Signal Process.*, vol. 149, pp. 148–161, Aug. 2018.

[34] Y. Zhang and Y. Tang, "A plaintext-related image encryption algorithm based on chaos," *Multimedia Tools Appl.*, vol. 77, no. 6, pp. 6647–6669, 2018.

[35] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[36] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[37] J. Chen, Y. Zhang, L. Qi, C. Fu, and L. Xu, "Exploiting chaos-based compressed sensing and cryptographic algorithm for image encryption and compression," *Opt. Laser Technol.*, vol. 99, pp. 238–248, Feb. 2018.

[38] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, "Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing," *Opt. Laser Technol.*, vol. 82, pp. 121–133, Aug. 2016.

[39] Q. Hu, D. Xiao, Y. Wang, and T. Xiang, "An image coding scheme using parallel compressive sensing for simultaneous compression-encryption applications," *J. Vis. Commun. Image Represent.*, vol. 44, pp. 116–127, Apr. 2017.

[40] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.

[41] J. Wang, L. Y. Zhang, J. Chen, G. Hua, Y. Zhang, and Y. Xiang, "Compressed sensing based selective encryption with data hiding capability," *IEEE Trans. Ind. Informat.*, to be published.

[42] Y. Luo, J. Lin, J. Liu, D. Wei, L. Cao, R. Zhou, Y. Cao, and X. Ding, "A robust image encryption algorithm based on Chua's circuit and compressive sensing," *Signal Process.*, vol. 161, pp. 227–247, Aug. 2019.

[43] L. Gong, K. Qiu, C. Deng, and N. Zhou, "An image compression and encryption algorithm based on chaotic system and compressive sensing," *Opt. Laser Technol.*, vol. 115, pp. 257–267, Jul. 2019.

[44] R. Ponuma and R. Amutha, "Encryption of image data using compressive sensing and chaotic system," *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 11857–11881, 2019.

[45] L.-B. Zhang, Z.-L. Zhu, B.-Q. Yang, W.-Y. Liu, H.-F. Zhu, and M.-Y. Zou, "Medical image encryption and compression scheme using compressive sensing and pixel swapping based permutation approach," *Math. Problems Eng.*, vol. 2015, Jul. 2015, Art. no. 940638.

[46] X. Chai, X. Zheng, Z. Gan, D. Han, and Y. Chen, "An image encryption algorithm based on chaotic system and compressive sensing," *Signal Process.*, vol. 148, pp. 124–144, Jul. 2018.

[47] I.-J. Lai and W.-H. Tsai, "Secret-fragment-visible mosaic image–a new computer art and its application to information hiding," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 936–945, Sep. 2011.

[48] Y.-L. Lee and W.-H. Tsai, "A new secure image transmission technique via secret-fragment-visible mosaic images by nearly reversible color transformations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 695–703, Apr. 2014.

[49] L. Bao and Y. Zhou, "Image encryption: Generating visually meaningful encrypted images," *Inf. Sci.*, vol. 324, pp. 197–207, Dec. 2015.

[50] X. Chai, Z. Gan, Y. Chen, and Y. Zhang, "A visually secure image encryption scheme based on compressive sensing," *Signal Process.*, vol. 134, pp. 35–51, May 2017.

[51] H. Wang, D. Xiao, M. Li, Y. Xiang, and X. Li, "A visually secure image encryption scheme based on parallel compressive sensing," *Signal Process.*, vol. 155, pp. 218–232, Feb. 2019.

[52] Z. Hua and Y. Zhou, "Image encryption using 2D Logistic-adjusted-Sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.

[53] K. Ramasubramanian and M. S. Sriram, "A comparative study of computation of Lyapunov spectra with different algorithms," *Phys. D, Nonlinear Phenomena*, vol. 139, no. 1, pp. 72–86, May 2000.

[54] C. G. Rong and D. Xiaoning, *From Chaos To Order: Methodologies, Perspectives And Applications*, vol. 24. Singapore: World Scientific, 1998.

[55] A. Kanso and M. Ghebleh, "An algorithm for encryption of secret images into meaningful images," *Opt. Lasers Eng.*, vol. 90, pp. 196–208, Mar. 2017.

[56] H. Fang, S. A. Vorobyov, H. Jiang, and O. Taheri, "Permutation meets parallel compressed sensing: How to relax restricted isometry property for 2D sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 196–210, Jan. 2014.

[57] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran, "Fast and efficient compressive sensing using structurally random matrices," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 139–154, Jan. 2012.

[58] A. Souyah and K. M. Faraoun, "Fast and efficient randomized encryption scheme for digital images based on quadtree decomposition and reversible memory cellular automata," *Nonlinear Dyn.*, vol. 84, no. 2, pp. 715–732, Apr. 2016.

[59] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, Sep./Oct. 2001.

[60] W. Zhang, H. Wang, D. Hou, and N. Yu, "Reversible data hiding in encrypted images by reversible image transformation," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1469–1479, Aug. 2016.

[61] D. Coltuc and J. M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Process. Lett.*, vol. 14, no. 4, pp. 255–258, Apr. 2007.

[62] R. Fay and C. Ruland, "Compressive Sensing encryption modes and their security," in *Proc. 11th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, Dec. 2016, pp. 119–126.

[63] Y. Zhang, J. Zhou, F. Chen, L. Y. Zhang, K.-W. Wong, X. He, and D. Xiao, "Embedding cryptographic features in compressive sensing," *Neurocomputing*, vol. 205, pp. 472–480, Sep. 2016.

[64] L. Zeng, X. Zhang, L. Chen, Z. Fan, and Y. Wang, "Scrambling-based speech encryption via compressed sensing," *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, p. 257, Dec. 2012.

PING PING was born in China. She received the B.Sc. degree in communication engineering and the Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2005 and 2009, respectively. She is currently an Associate Professor with the College of Computer and Information, Hohai University, Nanjing, China. Her research interests include data security and privacy, multimedia security, cloud computing security, and many other aspects of cryptography.

JIE FU was born in China. He received the B.Sc. degree in computer science and technology from Hohai University, Nanjing, China, where he is currently pursuing the master's degree with the College of Computer and Information. His research interests include information security and privacy, multimedia security, and many other aspects of cryptography.

YINGCHI MAO was born in China. She received the B.Sc. and M.Sc. degrees in computer science and technology from Hohai University, in 1999 and 2003, respectively, and the Ph.D. degree in computer science and technology from Nanjing University, China, in 2007. She is currently an Associate Professor with the College of Computer and Information, Hohai University, Nanjing, China. Her research interests include distributed computing, wireless sensor networks, and distributed data management.

FENG XU was born in China. He received the Ph.D. degree in computer science and technology from Nanjing University. He is currently a Professor and Ph.D. Supervisor with Hohai University, Nanjing, China. His research interests include cloud computing, network information security, and domain software engineering.

JERRY GAO is currently a Professor with the Department of Computer Engineering, San Jose State University. He has over 15 years of academic research and teaching experience and over ten years of industry working and management experience on software engineering and IT development applications. His current research areas include cloud computing, TaaS, software engineering, test automation, mobile computing, and cloud services.