A novel image compression-encryption scheme based on chaos and compression sensing

SHUQIN ZHU ¹, CONGXU ZHU ²,³ AND WENHONG WANG ¹

¹ School of Computer Science, Liaocheng University, Liaocheng, 252059, China
² School of Information Science and Engineering, Central South University, Changsha 410083, P. R. China
³ Guangxi Colleges and Universities Key Laboratory of Complex System Optimization and Big Data Processing, Yulin Normal University, Yulin 537000, P. R. China

Corresponding author: Congxu Zhu (zhucx@csu.edu.cn).

This work was supported by National Natural Science Foundation of China (No. 61472451), Shan Dong Province Nature Science Foundation (Grant. ZR2017MEM019) and the Open Project of Guangxi Colleges and Universities Key Laboratory of Complex System Optimization and Big Data Processing(No. 2016CSOBDP0103), Scientific research projects of universities in Shandong Province (Grant. J18KA336)

ABSTRACT In this paper, a novel image compression–encryption hybrid algorithm is proposed. Firstly, a Gauss random matrix and a random scrambling matrix are generated by using Chebyshev mapping and Logistic mapping respectively. Then, based on the principle that a scrambling Gauss matrix is still a Gauss matrix, a compression scheme for ciphertext images is designed. It is dependent on the Gauss random matrix and the random scrambling matrix, which mainly consists of three parts: permutation-based encryption using random scrambling matrix by Alice, encoding with Gauss random matrix by Charlie, and joint decryption and decoding by Bob. It has a special application scenario, that is, Alice requires semi integrity Charlie to transmit images to Bob through a channel. Experimental results show that the scheme has strong robustness against noise and chosen-plaintext attack, and further the peak signal to noise ratio (PSNR) and subjective visual quality of reconstructed images can be improved by comparing with the similar methods.

INDEX TERMS Compressed sensing; Gauss measurement matrix; random permutation matrix; chaos mapping; chosen plain text attack.

I. INTRODUCTION

In traditional signal processing, the sender first compresses and then encrypts, while the receiver first decrypts and then decompresses. The purpose of compression is to reduce data transmission and storage space as much as possible, and the purpose of encryption is to ensure data confidentiality. Considering such a special application scenario, the data sender Alice wants to transmit a message safely to the recipient Bob, but the channel provider Charlie is not completely trusted, so Alice needs to encrypt messages to a certain extent before sending the message to Charlie. Therefore, Charlie can not find the original message immediately after receiving the encrypted message. For recipient Bob, he performs decompression first and then decrypts to reconstruct the original image.

The theory of encrypted data compression was first proposed by Mark Johnson [1]. The method changed the traditional of performing compression preceding encryption, fully considering the security of data transmission and the requirement of data compression. On this basis, a codec model based on the spatial domain correlation of image was proposed in Ref. [2]. The model can apply the encrypted data compression theory to the general natural image, but it is difficult to obtain a good compression effect in the test of gray image. Based on the analysis sparse representation, a image compression-encryption hybrid algorithm was proposed in Ref. [3]. The algorithm can achieve good compression effect, in
II. PRELIMINARIES

A. COMPRESSED SENSING

The theory of compressed sensing was formally put forward in 2006 by Cands and Donoho [7, 8] and it has increasing applications in various fields such as image processing, pattern recognition, cloud computing, and internet of things, especially in the field of image encryption, because its dimensionality reduction and random projection characteristics can be utilized and integrated into the image cryptography system, thus image compression and encryption can be achieved [9, 10]. Zhang et al [11] pointed out that cryptographic features could be embedded in compressed sensing, and proposed several models of embedding cryptographic features in compressed sensing, which opened the way for compressed sensing to be applied in the field of encryption. Zhang et al [12] proposed a fast and efficient approach to color-image encryption based on compressive sensing and fractional Fourier transform and Huang et al [13] introduced a compression-diffusion-permutation scheme. In their schemes, images are compressed firstly, then the gray value of image pixels are changed and the positions of pixel are shuffled. In addition, the existing block compression sensing (BCS) image cryptography uses the same sampling rate for all blocks, which may cause important blocks to lose some useful information, while irrelevant blocks still retain insignificant information. To overcome this defect, a scalable encryption framework was proposed in [14], in which, lightweight subsampling and severe sensitivity encryption are used for important blocks, and heavy subsampling and lightweight robust encryption are used for unimportant blocks, so as to better protect important image areas. The schemes in Ref. [12-14] all adopt the strategy of first compression and then encryption.

As mentioned in [15], if a signal is sparse in a certain transform domain, the transformed high-dimensional signal can be projected into a low-dimensional space by an observation matrix unrelated to the transformation basis, and then the original signal can be reconstructed with high probability from these few projections by solving an optimization problem.

For a one-dimensional signal $x$ with the length of $N$, it can be represented as a linear combination of a set of orthonormal bases

\[ x = \sum_{i=1}^{N} x_i \psi_i = \psi s \]  

(1)

Where, $\psi$ is a basis matrix, $\psi = [\psi_1, \psi_2, ..., \psi_S]$. $\psi_i$ is a column vector of $\psi$. If the potential of the
support domain \( \{ \ell : s \neq 0 \} \) of the transformation coefficient \( S \) is less than equal to \( K \), then the \( S \) is a sparse representation of the signal \( x \), \( K \) is the sparsity of the signal \( x \). The measurement process of compressed sensing can be expressed as:

\[
y = \phi x = \phi \psi^T x
\]  
(2)

Here, \( \psi^T \) is a transposed matrix. \( \phi \) is a matrix of size \( M \times N \) \((M < N)\), called measurement matrix. The compressed sensing theory requires that the measurement matrix \( \phi \) and the sparse transform matrix \( \psi^T \) must obey the condition: the measurement matrix \( \phi \) is not related to the sparse transform matrix \( \psi^T \). At the same time, the length \( M \) of the measured value \( y \) should satisfy the following formula:

\[
K \geq cK \log(N / K)
\]  
(3)

Here, \( c \) is a very small constant, and \( K \) is sparsity. The recovery algorithm of the compressed sensing is to reconstruct sparse signals from a small number of linear observations, which can be seen as recovering the sparse signal \( x \in \mathbb{R}^N \) from the measurement vector \( y = \phi \psi^T x \). Theoretically, the simplest way is to solve the \( \ell_0 \) norm minimization problem, as shown in Eq. (4). The \( \ell_0 \) norm minimization problem is transformed to solve the \( \ell_1 \) norm minimization problem [16], as shown in Eq. (5).

\[
\min_{x \in \mathbb{R}^N} \| x \|_0 \quad \text{s.t.} \quad \phi x = y
\]  
(4)

\[
\min_{x \in \mathbb{R}^N} \| x \|_1 \quad \text{s.t.} \quad \phi x = y
\]  
(5)

Finally, the \( x \) is obtained by Eq. (1).

Thus, sparse signal representation, compression measurement and signal reconstruction are the three main components of CS theory. Curve transform, discrete cosine transform (DCT) and discrete wavelet transform (DWT) [17] are commonly used sparse representation methods. Signal reconstruction refers to the process of accurately reconstructing the original signal or high-dimensional image from the low-dimensional data and measurement matrix of the compressed image. There are many commonly used reconstruction algorithms: Orthogonal Matching Pursuit (OMP), Subspace Pursuit (SP) and Smoothing \( \ell_0 \) Norm (SLO) algorithm [18], this paper uses OMP algorithm. The design of observation matrix is the key to ensure the sampling quality of signals. At present, some scholars have proposed a number of measurement matrices, such as Gaussian random matrix, partial orthogonal matrix, Hadamard matrix and circular matrix [19]. In this paper the measurement matrix is a random matrix generated by Chebyshev map obeying Gauss distribution.

### B. THE GENERATION OF THE MEASUREMENT MATRIX

The design of observation matrix is the key to ensure the sampling quality of signals, and is also the key to decide the hardware implementation of compressed sampling. The methods of constructing the observation matrix are divided into two kinds. One is the random observation matrix, which mainly includes the Gauss observation matrix, the Bernoulli observation matrix, the local Fourier observation matrix and so on [7, 20]. Although these matrices can reconstruct the original signal very well, but in practical applications, it is difficult to implement with hardware, and storage matrix elements require large storage space. The other is the deterministic observation matrix. When constructing such matrices, once the system parameters and the constructed parameters are determined, the elements of the matrix are also determined. De-vore RA [21] proposes a polynomial method to construct a polynomial observation matrix, and Li Shuxing proposes to use algebraic curves to construct an observation matrix [22], but these methods can not construct a observation matrix of arbitrary size, and the row and column of the matrix must be an integral square of 2. Yu Lei [23] proposes a chaotic sequence that is easy to implement with hardware to construct the observation matrix. But in a sequence generated by a chaotic system, a sequence generated at intervals of 15 or more points is more statistically independent. When generating chaotic sequences of structural observation matrix, a large number of useless data points need to be generated, thus wasting system resources.

In order to overcome the defects that the random matrix is not easy to implement by hardware, and the defects that the size of the deterministic matrix based on polynomial and algebraic curves can not be arbitrarily set. Some research results of the complexities of chaotic systems show that the complexity of continuous chaotic systems are much smaller than those of the discrete chaotic systems [24-26]. In this paper, therefore, two discrete chaotic systems are adopted, one is Chebyshev map, and another is Logistic map. Chebyshev map is used to generate the measurement matrix which obeys Gauss distribution. When the deterministic random observation matrix is stored or transmitted, only storage or transmission of system equations and parameters can be stored and transmitted, which can save storage space and improve the transmission efficiency, especially when the matrix is larger, the superiority of deterministic random observation matrix is more obvious.
The definition of Chebyshev map is as
\[ x(i+1) = \cos(4 \cdot \arccos(x(i))) \]  
\[ (6) \]

The probability density function of the chaotic sequence generated by Eq. (6) is shown as
\[ \rho_x(x) = \begin{cases} \frac{1}{\pi \sqrt{1-x^2}} & \text{if } -1 < x < 1 \\ 0 & \text{else} \end{cases} \]
\[ (7) \]

**Theorem 1** If the probability density function of random variable \( X \) is shown as Eq. (7), then the random variable \( Y \)
\[ Y = \frac{1}{\pi} \arcsin(X) \]
\[ (8) \]
obeys a uniform distribution on (-0.5, 0.5).

**Proof:** Suppose the distribution function of the random variable \( Y \) is \( F_Y(y) \), then
\[ F_Y(y) = p(Y \leq y) = p\left(\frac{1}{\pi} \arcsin(X) \leq y\right) \]
\[ = p(X \leq \sin(\pi y)) = \int_{-\infty}^{\infty} \rho_x(x) \, dx \]
\[ (9) \]
Then, the probability density function of the random variable \( Y \) is \( \rho_Y(y) \)
\[ \rho_Y(y) = F_Y'(y) = \rho_x(\sin(\pi y))(\sin(\pi y))' \]
\[ = \frac{1}{\pi} \frac{\pi \cos(\pi y)}{\sqrt{1-\sin^2(\pi y)}} = \begin{cases} 1, & y \in (-0.5, 0.5) \\ 0, & \text{else} \end{cases} \]
\[ (10) \]
Therefore, the random variable \( Y \) obeys a uniform distribution on (-0.5, 0.5).

Obviously, if \( Y \) obeys uniform distribution on (-0.5, 0.5), then \( Z = 2 \cdot |Y| \) obeys uniform distribution on (0, 1).

**Lemma 1** [27] Suppose that \( U_1 \) and \( U_2 \) are two independent random variables and obey uniform distribution on (0,1), and two random variables \( Z_1, Z_2 \) are obtained by formula (11) and (12).
\[ Z_i = \sqrt{-2 \ln(U_i)} \times \cos(2\pi U_i) \]  
\[ (11) \]
\[ Z_2 = \sqrt{-2 \ln(U_i)} \times \sin(2\pi U_2) \]  
\[ (12) \]
Then, \( Z_1 \) and \( Z_2 \) are two independent random variables, and the distribution of these two random variables is the standard Gauss distribution.

From the above **Theorem 1** and **Lemma 1**, we can get the specific steps of obtaining the measurement matrix which obeys the standard Gauss distribution.

**Step 1** Setting the \( x_0(0) \) as the initial value of the chaotic system (6), and iterating the chaotic system (6) to get a chaotic sequence \( X_1 \) of length \( K \times M \). Resetting the initial value of chaotic system (6) by using the \( x_0(0) \), a chaotic sequence \( X_2 \) with the same length as chaotic sequence \( X_1 \) is generated.

**Step 2** Two new random sequences \( U_1 \) and \( U_2 \) are constructed by random sequences \( X_1 \) and \( X_2 \) as shown in Eqs. (13) and (14).
\[ U_i = 2 \times \frac{1}{\pi} \arcsin(X_i) \]
\[ (13) \]
\[ U_2 = 2 \times \frac{1}{\pi} \arcsin(X_2) \]
\[ (14) \]

**Step 3** Using sequences \( U_1 \) and \( U_2 \), we generate two sequences of \( Z_1 \) and \( Z_2 \) which obey the standard Gauss distribution through formula (11) and (12). Convert any of \( Z_1 \) or \( Z_2 \) into a matrix of size \( K \times M \), which is used as the measurement matrix \( \phi \) in compressed sensing.

The histogram of the measurement matrix \( Z_1 \) and \( Z_2 \) is shown in Fig.1, which obviously obeys the standard Gauss distribution.

![Histogram of the two measurement matrices](image)
C. THE GENERATION OF THE RANDOM PERMUTATION MATRIX

A random permutation matrix $R$ is a matrix in which only one element in each row or column is the number 1 and the rest is the number 0. It can be generated by Logistic mapping. The definition of Logistic mapping is shown in formula (15).

$$x(i+1) = a \times x(i) \times (1-x(i))$$ (15)

Where, $a$ is a branch parameter, $x(i) \in (0, 1)$. When $3.5699456 < a < 4$, the Logistic map is in a chaotic state, the specific steps of obtaining the random permutation matrix $R$ are as follows.

**Step1** Setting the initial value of logistic map $x(0)$, iterating the logistic map (15) to produce a sequence denoted as $X=\{x(1), x(2), x(3), …, x(M)\}$. Then sequences $X$ is sorted as $[a_l, a_x] = \text{sort}(X)$, (16)

Where $[\cdot, \cdot]$ is the sequence index function, $a_l$ is the new sequence, whereas $a_x$ is the index sequence of $a_l$. So $a_x$ is a vector containing a random permutation of the integers $1:M$.

**Step2** According to the sequence $a_x$, a random scrambling matrix $R$ of size $M \times M$ can be generated, and the MATLAB pseudo code is described as follows

```matlab
R=zeros(M);
for i=1:length(R)
    R(ax(i), i)=1;
End
```

The generated matrix $R$ is a random scrambling matrix, in which each row or column has exactly one and the rest are all zero. If the scrambling matrix $R$ is multiplied on the right side of the image matrix $P$, that is $R \times P$, each element of the image matrix $P$ is scrambled. However, if the scrambling matrix $R$ is multiplied on the right side of the image matrix, that is $P \times R$, the elements of each row of the image matrix $P$ are scrambled. For example,

$$R = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad P = \begin{bmatrix} 10 & 19 & 65 & 230 \\ 22 & 89 & 80 & 125 \\ 34 & 77 & 90 & 170 \\ 128 & 29 & 69 & 187 \end{bmatrix},$$

then

$$R \times P = \begin{bmatrix} 22 & 89 & 80 & 125 \\ 128 & 29 & 69 & 187 \\ 10 & 19 & 65 & 230 \\ 34 & 77 & 90 & 170 \end{bmatrix},$$

$$P \times R = \begin{bmatrix} 65 & 10 & 230 & 19 \\ 80 & 22 & 125 & 89 \\ 90 & 34 & 170 & 77 \\ 89 & 128 & 187 & 29 \end{bmatrix}.$$}

We can see that $R \times P$ is the result of internal scrambling of $P$ columns, and $P \times R$ is the result of internal scrambling of $P$ rows.

D. THE GENERATION OF SCRAMBLING SEQUENCE

The scrambling sequence is used to scramble the image pixel's positions, which is generated by logistic map. Setting the initial value of logistic map $x(0)$, According to the first step of section C of chapter II, generate a position index sequence $T$ of length $K \times N$.

III. THE COMPRESSION-ENCRYPTION SCHEME

Compressing encrypted images is a challenging task, because encryption will weaken the correlation between pixels or change the histogram distribution of the image (making the ciphertext image more evenly distributed), and the compression itself needs to make use of the redundancy between pixels or the inhomogeneity of the pixel distribution. In fact, images that are lightly encrypted can still be compressed to certain extent. The so-called light encryption means only permute the pixels or mask the pixel values by a Random sequence [28–31]. The lightweight encryption schemes are usually not secure enough, but lightweight encryption can satisfy the needs of some application scenarios. When Alice sends an image to Bob, the lightweight encryption and decryption will save a lot of resources for Alice and Bob, and the channel provider Charlie, often assuming that he is just curious, does not deliberately destroy it. In other words, suppose that Charlie can not immediately observe or infer plaintext information from the transmitted lightweight encrypted ciphertext by Alice. At the same time, he must faithfully do the job of channel transmission. The function of Charlie is somewhat similar to that of semi trusted public cloud [32-34], curious and honest.

The proposed scheme is composed of three steps: Alice encrypts images with scrambling technology; Charlie encode the ciphertext images by using the newly generated Gauss matrix; Finally, when decrypting, the joint decryption and decompression will restore the plaintext image by Bob.

A THE GENERATION OF INITIAL VALUES OF CHAOTIC SYSTEMS

The security of the proposed algorithm depends on the Gauss random matrix, the random scrambling matrix and the scrambling sequence $T$. In order to increase the relationship between encryption scheme and plaintext image, we use the SHA 256 hash function of the plain image as part of the initial value of chaotic system. The SHA-256 hash value of the plaintext $H$ is divided into 32 groups with 8 bits, so the $H$ can be represented as $H=h_1,h_2,h_3,…,h_{32}$. (each $h_i$ is an 8-bit binary composed of 0 and 1). The initial values of the chaotic system are set by formulas (17) - (24).

$$l_i = \text{mod}(h_i \oplus h_j \oplus h_k \oplus h_l \oplus h_m \oplus h_n \oplus h_o \oplus h_i + \sum_{i=1}^{32} h_i / 256, 256) \quad (17)$$
l_i = \text{mod}(h_0 \oplus h_1 \oplus h_2 \oplus h_3 \oplus h_4 \oplus h_5 \oplus h_6 + \sum_{i=0}^{n} h_i, 256) / 256 \quad (18)
l_i = \text{mod}(h_0 \oplus h_1 \oplus h_2 \oplus h_3 \oplus h_4 \oplus h_5 \oplus h_6 + \sum_{i=0}^{n} h_i, 256) / 256 \quad (19)
l_i = \text{mod}(h_0 \oplus h_1 \oplus h_2 \oplus h_3 \oplus h_4 \oplus h_5 \oplus h_6 + \sum_{i=0}^{n} h_i, 256) / 256 \quad (20)

x_1(0) = \text{mod}(x_1(0) + l_1, 1) \quad (21)
x_1(0) = \text{mod}(x_1(0) + l_1, 1) \quad (22)
x_1(0) = \text{mod}(x_1(0) + l_1, 1) \quad (23)
x_1(0) = \text{mod}(x_1(0) + l_1, 1) \quad (24)

Where x_1(0), x_2(0), x_3(0), x_4(0) are the initial value. In this way, the initial values x_1(0), x_2(0), x_3(0), x_4(0) of a chaotic system consists of two parts, namely, x_1(0), x_2(0), x_3(0), x_4(0) and l_1, l_2, l_3. The cryptographic system’s key sets are keys = \{x_1(0), x_2(0), x_3(0), x_4(0)\}, SHA-256 hash value of the image. Therefore, when encrypting different images, even if the same keys x_1(0), x_2(0), x_3(0), x_4(0) are used, the generated Gaussian measurement matrix, the random scrambling matrix and the scrambling sequence T are different.

**B. PERMUTATION-BASED ENCRYPTION**

Assume the size of the plain image P is M × N. Alice first set x_k(0) to get the initial value x_k(0) of chaotic system (15), and then use the method proposed in section C of chapter II to generate random scrambling matrix R of size M × M. Then she encrypts P to get the cipher P_w by applying a random permutation matrix R governed by

\[ P_w = R \times P \quad (25) \]

P_w is then sent to Charlie. Obviously, R is an orthogonal matrix, i.e., \( R^T = R^{-1} \). So \( P = R^T \times P_w = R^T \times P \). This means that decryption does not require the inverse matrix of R, which greatly saves the time of decryption. Although simple scrambling technology can not completely guarantee the security of images, it can be applied to occasions where high secrecy is not a must.

**C. COMPRESSION P_w USING COMPRESSION SENSING TECHNOLOGY**

After the encrypted image has been received, Charlie first set x_k(0) and x_k(0) to get the two initial values x_1(0) and x_2(0) of Chebyshev map (6), then constructs a Gauss measurement matrix \( \phi \) of size \( K \times M(K < M) \) using the method proposed in the section B of chapter II sample it. Encoding using \( \phi \) is expressed as

\[ y = \phi \times P_w = \phi \times R \times P = (\phi \times R) \times P \quad (26) \]

Where, \( \phi \times R \) can be regarded as a scrambling matrix for every row of matrix \( \phi \) and it is also a Gauss matrix. From Eq.(18), we can see that measuring \( P_w \) with \( \phi \) is equivalent to measuring \( P \) with \( \phi \times R \).

The measured values y of secret images after compressed sensing are relatively large, and the magnitude can reach 10^5. Charlie can use the following transformation, as shown in formula (27) to adjust its range to 0~255. and saves it as an unsigned integer, and the storage space for each number is only one byte. The \( y_{\text{max}} \) is the maximum value in the measured value, and the \( y_{\text{min}} \) is the minimum in the measured value.

\[ y = (y - y_{\text{min}}) \times 255 / (y_{\text{max}} - y_{\text{min}}) \quad (27) \]

To further strengthen the security of the system, a pixel-scrambling method is employed to re-encrypt the compressed and encrypted image yy. Firstly, set x_4(0) to get the initial value x_4(0) of Logistic map (15), then generate sequence T by using the method proposed in the section D of chapter II. Second transform yy into a sequence yy* of length K×N, then the pixel’s positions of yy* are scrambled with the scrambling sequence T and the compressed-encrypted image C is obtained.

**D. JOINT DECRYPTION AND DECODING**

At the receiving side, Bob obtains the compressed-encrypted image C. Get yy by reverse scrambling C, then to get y by the following transformation, as shown in formula (28)

\[ y = yy \times (y_{\text{max}} - y_{\text{min}}) / 255 + y_{\text{min}} \quad (28) \]

Then applies joint decryption and decoding to recover the original image using the following algorithm:

\[ \min_{x \in \mathbb{R}^n} \left\| x - R \times \phi \times P \right\|_1 \quad (29) \]

That is to say, Bob regards \( \phi \times R \) as a measurement matrix to recover plain images.

**IV. EXPERIMENTAL SIMULATION**

Two natural images including Cameraman and Rice are used for testing. The sparse transformation is wavelet transform. The reconstruction algorithm is OMP. The cryptographic system’s key sets are keys = \{x_1(0), x_2(0), x_3(0), x_4(0)\}, SHA-256 hash value of the image. Setting the initial values x_1(0) = 0.8964, x_2(0) = 0.0447, x_3(0) = 0.4765, x_4(0) = 0.3908 and calculating hash value of plaintext image. The plain images Cameraman (256 × 256) and Rice (128 × 128) and the corresponding ciphertext images with different compression rates are shown in Fig. 2. The compression ratio CR is computed by [35]

\[ CR = C / (h \times C) \quad (30) \]
Where, $I_h$ and $I_w$ denote the height and width of the original image, respectively. And $C_h$ and $C_w$ are the corresponding height and width of the cipher image. It can be seen that the ciphertext image can not reveal any information of the plaintext image, and the ciphertext image with different compression ratio can be effectively decrypted.

![Image](image1.png)

![Image](image2.png)

**FIGURE 2.** Experimental simulation results. (a) Plain image cameraman. (b) Cipher image of (a) with CR=0.5. (c) Cipher image of (a) with CR=0.65. (d) Decrypted image for (b). (e) Decrypted image for (c). (f) Plain image rice. (g) Cipher image of (f) with CR=0.5. (h) Cipher image of (f) with CR=0.65.; (i) Decrypted image for (g). (j) Decrypted image for (h).

V. PERFORMANCE ANALYSES

A. PERFORMANCE ANALYSIS OF PSNR

The peak signal to noise ratio (PSNR) is usually used to evaluate the quality of an image compression compared with the original image. The higher the PSNR, the smaller the distortion after compression. Two values are defined here, one is the mean square deviation MSE and the other is the peak signal-to-noise ratio PSNR. The MSE and PSNR values are described by the next expressions [36]
$MSE = \left\{ \frac{1}{M_1 \times M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} (I_0(i,j) - I_r(i,j))^2 \right\}^{\frac{1}{2}}$ (31)

$PSNR = 20 \log \left( \frac{I_{\text{max}}}{MSE} \right)$ (32)

Where, $M_1$ represents the number of rows of an image, and $M_2$ represents the number of columns of the image. $I_0$ represents the original image, $I_r$ means the restored image, $I_{\text{max}}$ represents the maximum pixel value of the image, and it is 255 for the grayscale image. Table 1 lists the experimental PSNR results with different compression ratio and we also used other algorithms in Ref.s [5, 37, 38] to test the PSNR results of “cameraman” and “rice” at different compression rates. Table 1 also lists the PSNR values of reconstruction effect in these algorithms. The results show that our reconfiguration effect is better than that of Ref.s [5, 38] and slightly less effective than that in [37].

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Tested images</th>
<th>PSNR (CR=0.55)</th>
<th>PSNR (CR=0.65)</th>
<th>PSNR (CR=0.75)</th>
<th>PSNR (CR=0.85)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref. [38]</td>
<td>Cameraman</td>
<td>25.0872</td>
<td>25.5210</td>
<td>26.8198</td>
<td>27.8392</td>
</tr>
<tr>
<td>This paper</td>
<td>Cameraman</td>
<td>25.6457</td>
<td>25.9679</td>
<td>27.4550</td>
<td>27.9987</td>
</tr>
<tr>
<td>Ref. [37]</td>
<td>Rice</td>
<td>24.4637</td>
<td>25.3397</td>
<td>26.8792</td>
<td>29.0176</td>
</tr>
<tr>
<td>Ref. [38]</td>
<td>Rice</td>
<td>23.9310</td>
<td>24.6156</td>
<td>26.0098</td>
<td>28.5398</td>
</tr>
<tr>
<td>This paper</td>
<td>Rice</td>
<td>24.3298</td>
<td>24.9909</td>
<td>26.7584</td>
<td>28.8987</td>
</tr>
<tr>
<td>Ref. [5]</td>
<td>Peppers</td>
<td>26.5484</td>
<td>27.9798</td>
<td>27.9910</td>
<td>27.7173</td>
</tr>
<tr>
<td>Ref. [37]</td>
<td>Peppers</td>
<td>28.2312</td>
<td>28.8796</td>
<td>29.5409</td>
<td>30.0982</td>
</tr>
<tr>
<td>Ref. [38]</td>
<td>Peppers</td>
<td>27.3387</td>
<td>27.9921</td>
<td>28.6732</td>
<td>29.6076</td>
</tr>
<tr>
<td>This paper</td>
<td>Peppers</td>
<td>27.9484</td>
<td>28.4798</td>
<td>29.1910</td>
<td>29.9173</td>
</tr>
</tbody>
</table>
**B. KEY SPACE ANALYSIS**

For a secure image encryption algorithm, its key space is at least $2^{100}$ to resist brute-force attacks. In the proposed scheme, the secret key consists of two parts as follows: (1) the initial values $x_1(0), x_3(0), x_4(0), x_4(0)$ of the chaotic systems. (2) The SHA 256 hash function of the plain image. According to the IEEE floating-point standard, the precision of the double-precision value is approximately $10^{-15}$. The SHA-256 key space can reach $2^{128}$. Therefore, the total key space is $10^{15} \times 10^{15} \times 10^{15} \times 2^{128} \approx 2^{300}$, which is greater than $2^{100}$. It means that our image compression-encryption algorithm meets the security demand of key space. Table 2 gives the key space of different schemes.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Proposed algorithm</th>
<th>Algorithm in [37]</th>
<th>Algorithm in [38]</th>
<th>Algorithm in [39]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key space</td>
<td>$2^{300}$</td>
<td>$2^{187}$</td>
<td>$2^{276}$</td>
<td>$2^{249}$</td>
</tr>
</tbody>
</table>

**C. KEY SENSITIVITY ANALYSIS**

The more sensitive the algorithm is, the more secure the algorithm is. The algorithm is sensitive to the key, which means that when decrypting, if the wrong decryption key is input which is slightly different from the correct key, no useful information of the plaintext image can be obtained from the decryption result. Take the gray image ‘Cameraman’ as an example to test the key sensitivity of the proposed algorithm. The sensitivity of the chaotic system to the initial condition determines the sensitivity of the encryption system to the keys. Experiments show that the accuracy of the algorithm can reach $10^{-15}$ for $x_1(0), x_3(0), x_4(0), x_4(0)$, that is, the attacker’s attempt to crack the original image is futile even if the key value has a slight deviation of $10^{-15}$. Fig. 3 (a) ~ (f) (CR=0.65) are the images decrypted by different error keys. Fig. 3(a) ~ (c) are the results of decryption when the initial values of the chaotic system are incorrect and Fig. 3(d) ~ (f) are the results of decryption when the hash function of the plaintext image is incorrect. It can be seen that no information of the original image can be obtained in the decrypted image. The 256-bit hash value is changed from $K$ to $K_0$, $K_1$ and $K_2$ with a bit modified, and they are given below.

$K = \text{d6f5e24bf170a68a37c9b8bfdec91dc39777d7a98e67d45}$

$K_0 = \text{c6f35e24bf170a68a37c9b8bfdec91dc39777d7a98e67d45}$

$K_1 = \text{d6f35e24bf170a68a37c9b8bfdec91dc39777d7a98e67d45}$

$K_2 = \text{d6f35e24bf170a68a37c9b8bfdec91dc39777d7a98e67d45}$
FIGURE 3. The results of decryption with the error keys. 
(a) Decrypted image with $x_1(0) + 10^{-15}$. 
(b) Decrypted image with $x_2(0) + 10^{-15}$. 
(c) Decrypted image with $x_3(0) + 10^{-15}$. 
(d) Decrypted image with $x_4(0) + 10^{-15}$. 
(e) Decrypted image with $K_0$. 
(f) Decrypted image with $K_1$. 
(g) Decrypted image with $K_2$.

D. HISTOGRAM ANALYSIS

Histogram is a basic attribute of digital image. It reflects the distribution of pixel gray value in the image. The more uniform the distribution of histogram of ciphertext image, the better the encryption effect [40]. In addition, if the histogram of different ciphertext images is similar, the encryption effect is good [41-43]. Figs. 4 (a), (c) and (e) are the histograms of three original images with different size: “Cameraman”, “Rice”, and “Peppers”, respectively. Figs. 4 (b), (d) and (f) are the histograms of their corresponding encrypted images. Obviously, the histograms of the three plaintext images are different, but surprisingly, the histograms of the corresponding ciphertext are similar. So, encrypted images can resist statistical attacks and have good cryptographic properties.
E. ANTI ACTIVE ATTACK ANALYSIS

The previous security analysis is mainly for passive attacks, which mainly collect information rather than access, and the legitimate users of the data are not aware of such activities. Another attack method is active attack, in which the active attacker will delete, change information illegally, and threaten the integrity and effectiveness of information. Therefore, active attack is more harmful than passive attack, so it is necessary to test the anti-active attack of encryption system. In the active anti attack test, we mainly test the ability of ciphertext images to resist shear and noise pollution.

1) Anti cutting test

From the ciphertext and the ciphertext image histogram, we can infer that the pixel value of ciphertext image is mostly 120. We add the pixel value of the cut part to 120, then the image can be restored, as shown in Fig. 5(d). It can be seen that the decrypted image has a slight noise, but this does not affect the visual effect of the image.

2) Test of anti noise pollution attack

In the ciphertext image, all kinds of noises are added artificially to test the ability of resisting noise of encrypted images. Fig. 6 is the test result of the ciphertext image polluted by the salt and pepper noise with the intensity of 0.001 and Fig. 7 is the test result of the ciphertext image polluted by Gauss noise with the intensity of 0.00001. Because the intensity of salt and pepper noise is large, but the noise distribution is loose; the intensity of Gaussian noise is small, but the distribution is dense, so compared with Fig.s 6 and 7, the ciphertext image can resist salt and pepper noise pollution more powerful.

FIGURE 4. Histogram analysis results: (a) the histogram of image cameraman. (b) The histogram of cipher image for (a). (c) The histogram of plain image rice; (d) The histogram of cipher image rice. (e) The histogram of Plain image Peppers. (f) The histogram of cipher image Peppers.

FIGURE 5. Anti cutting test: (a) The clipped ciphertext image; (b) images restored from (a).

FIGURE 6. Test of salt and pepper noise pollution: (a) Ciphertext image after the intensity of 0.0001 salt and pepper noise pollution; (b) Decryption of the image.
According to the Kerchoff principle, the security of the encryption system depends on the key, not on the encryption algorithm itself. Chosen plaintext attack [44-46] means that an attacker temporarily gains the right to use the encryption machine, so he can encrypt arbitrary plaintext and obtain the corresponding ciphertext to decipher all or part of the plaintext and key. Chosen-plaintext attack is the strongest attack, if a cryptosystem can resist this attack, it must be able to resist other attacks. In the paper, SHA 256 hash function of the original image is computed as part of the initial values of a chaotic systems. Thus, when different plain images are encrypted, the corresponding Gauss random matrix $\phi$, the random scrambling matrix $R$ and scrambling sequence $T$ are changed too, so the hackers cannot get useful information by encrypting some special images. Therefore, the proposed scheme can well withstand the chosen-plaintext attacks.

G. COMPUTATIONAL COMPLEXITY ANALYSIS

The difference between the algorithm and the traditional compression sensing algorithm lies in the generation of two matrices: one is random scrambling matrix and the other is the Gauss measurement matrix. In the image restoration stage, the algorithm is exactly the same as the traditional compressed sensing image restoration algorithm. We analyze the encryption and decryption time of different size images at different compression ratios (CR), and the results are listed in Tables 3 and 4.

![Image](image_url)

**TABLE 3. Encryption time (second).**

<table>
<thead>
<tr>
<th>Images size</th>
<th>Rice</th>
<th>cameraman</th>
<th>Peppers</th>
</tr>
</thead>
<tbody>
<tr>
<td>128×128</td>
<td>0.329325</td>
<td>0.475708</td>
<td>0.817562</td>
</tr>
<tr>
<td>256×256</td>
<td>0.379671</td>
<td>0.498010</td>
<td>0.938217</td>
</tr>
<tr>
<td>512×512</td>
<td>0.378972</td>
<td>0.536409</td>
<td>0.974393</td>
</tr>
</tbody>
</table>

Table 5 is a comparison of the encryption time between our algorithm and other algorithms under the condition of compression ratio CR = 0.55. It can be seen that this algorithm is faster than that in Ref.s [47, 48], but slightly slower than that in Ref. [6] when encrypting the same size image. Compared with the pure image encryption algorithm based on SHA-256 [49], the speed of the image compression-encryption algorithm in this paper is slightly slower.

**TABLE 4. Decryption time (second).**

<table>
<thead>
<tr>
<th>Images size</th>
<th>Rice</th>
<th>cameraman</th>
<th>Peppers</th>
</tr>
</thead>
<tbody>
<tr>
<td>128×128</td>
<td>1.235198</td>
<td>7.352746</td>
<td>20.718625</td>
</tr>
<tr>
<td>256×256</td>
<td>1.821770</td>
<td>8.800748</td>
<td>25.126654</td>
</tr>
<tr>
<td>512×512</td>
<td>2.181208</td>
<td>9.058219</td>
<td>33.558026</td>
</tr>
</tbody>
</table>

**TABLE 5. The encryption time comparison results with other algorithms (second).**

<table>
<thead>
<tr>
<th>Images size</th>
<th>Rice</th>
<th>cameraman</th>
<th>Peppers</th>
</tr>
</thead>
<tbody>
<tr>
<td>128×128</td>
<td>0.379602</td>
<td>0.498021</td>
<td>0.938217</td>
</tr>
<tr>
<td>256×256</td>
<td>0.239800</td>
<td>0.308500</td>
<td>0.536800</td>
</tr>
<tr>
<td>512×512</td>
<td>0.479811</td>
<td>0.713404</td>
<td>0.998765</td>
</tr>
<tr>
<td>256×256</td>
<td>3.524321</td>
<td>5.556790</td>
<td>8.974393</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

In this paper, we design a compression scheme for ciphertext images. It is dependent on the Gauss random matrix, random scrambling matrix and the scrambling sequence $T$, which mainly consists of three steps: Alice encrypts images with scrambling technology. Charlie encode the ciphertext images using the newly generated Gauss matrix. Finally, when decrypting, the joint decryption and decompression will restore the plaintext image by Bob. This scheme makes full use of the principle that the scrambling Gauss matrix is still the Gauss matrix. Experimental results show that the scheme has strong robustness against noise.

As for image encryption, some future studies is worth considering, such as efficient image encryption technology in resource-constrained mobile social network [50], sensor network communication environment [51]. And searchable encryption [52], which is a very promising direction in the field of cloud computing.

REFERENCES


