IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 April 2024, accepted 21 May 2024, date of publication 29 May 2024, date of current version 7 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406766

RESEARCH ARTICLE

A Color Image Encryption Algorithm Based on Compressive Sensing and Block-Based DNA Coding

QIJI HE, PEIYA LI, AND YANYIXIAO WANG
College of Cyber Security, Jinan University, Guangzhou 510632, China

Corresponding author: Peiya Li (lpy0303@jnu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3103500, in part by the National Natural Science Foundation of China under Grant 62302195, and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515011960.

ABSTRACT Image encryption is an effective method to prevent images from being captured or stored illegally. With the increasing demand for image security and transmission efficiency, this paper proposes a color image encryption algorithm based on chaotic systems, which combines block-based *DNA* coding and compressive sensing. Firstly, the plain image is compressed using compressive sensing (*CS*) to obtain three measurement value matrices, which are then quantized into integer matrices and permuted by the method of the Josephus problem. Subsequently, the scrambled measurement value matrices are divided into subblocks. These subblocks are encoded by random *DNA* rules, diffused by chaotic sequences-based *DNA* operations, and dynamically decoded. It is worth noting that the initial values of the chaotic systems used for the *DNA* operations and the generation of the measurement matrices depend on the plain image, which enables our algorithm to withstand well the chosen-plaintext attacks and the known-plaintext attacks. Moreover, we utilize singular value decomposition (*SVD*) to optimize the measurement matrices to enhance the reconstruction quality of the encrypted image. Simulation results and security analysis show that the algorithm has excellent compression and encryption performance and is resistant to various typical attacks.

INDEX TERMS Image encryption, compressive sensing, chaotic system, color image, block-based DNA encoding.

I. INTRODUCTION

Nowadays, with the rapid development of Internet communication and multimedia, digital images have been widely applied in military, medical, and commercial fields. However, images are frequently transmitted in a shared and accessible cyberspace, where their security and transmission efficiency have become a challenge. Image encryption is an effective technology that can protect image information from illegal interception or tampering. Considering the high redundancy of images and the strong correlation of adjacent pixels, traditional text encryption algorithms, such as *AES*, *DES*, and *RSA*, are no longer suitable for image encryption. There are great deals of alternative image encryption algorithms introduced with different technologies, such as chaotic systems [1], [2], [3], *CS* [4], [5], [6], *DNA* encoding [7], [8], [9], [10], quantum computing [11], [12], cellular automata [13], [14] and so on, which turn meaningful plain images into noise-like or texture-like cipher images preventing the attacker from obtaining any valid plain image information.

Since Friedrich et al. [15] proposed a classical permutation and diffusion encryption architecture based on chaotic systems in 1998, image encryption techniques based on chaotic systems have been widely studied in recent years due to their excellent properties consisting of good pseudorandomness, unpredictability, ergodicity, and extreme sensitivity to initial values. One-dimensional($1D$) chaotic systems with simple formats are capable of rapidly generating chaotic sequences. In [16], the logistic map was utilized to encrypt the confused

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

image. Jain et al. [17] generated a mask by using a one-dimensional chaotic map to add with the *DNA*-encoded image. However, Li et al. [18] demonstrated the insecurity of image encryption algorithms based on chaotic logic mapping through a typical example. Wang et al. [19] also successfully conducted a cryptanalysis of an image encryption scheme [20] based on a one-dimensional chaotic system with a small key space via the chosen-plaintext attack. Increasingly, high-dimensional (*HD*) chaotic systems are adopted for image encryption systems because of their more complex dynamic behavior. In reference [21], the authors proposed a color image encryption scheme based on a novel Five Dimensional Sine Chaotic System that produces chaotic data used for encryption. Es-Sabry et al. [22] presented a secure image encryption scheme using a high-dimensional chaotic system and *DNA* coding, which adopts the improved Rossler system as the keystream generator, and the improved Rossler system has better chaotic performance and higher security. Zou et al. [23] developed a new wide-range 2*D*-Logistic-Sine hyperchaotic map, which was proved to possess good ergodicity and unpredictability, and proposed an image encryption algorithm by combining this map with an improved zigzag diffusion method. In addition, Li et al. [24] introduced a new fast image encryption scheme based on Chen's hyper-chaotic system, which generates random sequences to scramble the four low-frequency components of the plain image. Despite the high security of *HD* chaotic systems with complex trajectories and long iteration periods, the time complexity is higher and more time-consuming than 1*D* chaotic systems. Zhu et al. [25] established a fast and flexible novel chaos-based image encryption algorithm that utilizes a 1*D* piece-wise quadratic polynomial chaotic map. Compared with some existing 1*D* chaotic systems, the composite chaotic system has better chaotic properties and chaotic performance and shows robust chaotic behavior over a large range of continuous parameters. In this paper, we combine the advantages of low time complexity of 1*D* chaotic systems and excellent randomness of *HD* chaotic systems by using three composite chaotic systems: Logistic-Tent System (*LTS*), Logistic-Sine System (*LSS*) and Tent-Sine System (*TSS*). The entropy values and the *SHA*-384 hash value of the plain image are utilized to calculate the initial values of the composite chaotic systems, which enables our encryption algorithm to be robust to both known-plaintext attacks and chosen-plaintext attacks.

Due to the high parallelism and large storage capacity of *DNA*, there are a growing number of image encryption schemes that combine *DNA* encoding and chaotic systems have been developed [26], [27], [28]. For example, in [29], Chen et al. proposed a technique for image encryption with the help of self-adaptive permutation-diffusion and *DNA* random coding. Reference [30] presented a color image encryption algorithm based on dynamic *DNA* encryption and a four-wing hyperchaotic system. They adopted the *SHA*-384 hash of the plain image to calculate the initial values of the hyperchaotic system, which resulted in an encryption scheme that is very sensitive to the plain image. However, the current study suffers from two major safety concerns [31]. The first one is that for many image encryption systems, the *DNA* encoding and decoding rules are stationary for all pixels. The adoption of fixed rules not associated with the plain image will significantly degrade the security of the image encryption scheme. The second one is that most *DNA*-based image encryption algorithms employ the *DNA* addition operation, *DNA* subtraction, *DNA XOR* operation, and *DNA XNOR* operation to diffuse the *DNA* matrices. They are all based on binary calculus, so the result of the conversion between the four *DNA* bases is well predicted, which increases the probability that the encryption algorithm is cracked. Liang et al. [32] proposed an image encryption algorithm based on a 1*D* sine-cosine chaotic map and random *DNA* encoding. Although the new chaotic system has good chaotic performance and the *DNA* encoding rules are based on pseudo-random sequences, the *DNA* operations are limited to addition, subtraction, exclusive *OR* and complementary, and the algorithm is only applicable to grayscale images. Combining Newly Designed Chaotic Map and parallel *DNA* coding, Zhu et al. [33] designed a gray-scale image encryption algorithm, their chaotic system has a larger range of chaotic parameters than the traditional 1*D* chaotic system and the encryption efficiency of the algorithm is higher than that of common *DNA* coding-based image encryption algorithms. However, the algorithm has a small number of image subblocks when performing parallel *DNA* operations, which means that the increase in speed is more limited. Moreover, the *DNA* permutation is only based on the chaotic sequence index permutation, and its security is still insufficient. In this paper, composite chaotic systems and block-based *DNA* coding are integrated and a color image encryption algorithm is proposed. The algorithm divides the image into small sub-blocks and then synchronizes *DNA* encoding, *DNA* diffusion and *DNA* decoding of the image sub-blocks, which are all extremely dependent on chaotic sequences. Notably, this algorithm performs two rounds of *DNA* diffusion, which improves the security considerably.

Considering the frequent transmission of images and the large size of images, *CS* is available for image compression to improve transmission efficiency, reduce transmission bandwidth and save storage space. *CS* enables simultaneous non-uniform sampling, compression and encryption of images. Since Donoho proposed compressive sensing [34], numerous *CS*-based image encryption schemes have been introduced. Lu et al. [35] presented an image encryption scheme based on *CS* and a double random-phase encoding technique. Although this algorithm is satisfactory, they consider the entire measurement matrix as a key, thus increasing the transmission burden and storage space of the key. Huang et al [36] constructed an algorithm that enhanced the security of the *CS*-based algorithm by using a chaotic system to permute the compressed image, but the key remains the entire measurement matrix. None of these schemes are resistant to chosen-plaintext attacks. Therefore, most of the studies utilize the information from the plain image to calculate the initial values of the chaotic systems and then iterate the chaotic systems to obtain the chaotic sequences to construct the measurement matrices. In [37], a fast image encryption algorithm based on parallel *CS* and *DNA* sequences is presented. In their scheme, they adopted the *SHA*-256 hash value and information entropy of the plain image to construct the measurement matrix, and thus the compressed and encrypted image was sensitive to the plain image. Reference [38] developed an image encryption scheme that is based on compressive sensing and chaos. The scheme utilizes *LTS* to construct a measurement matrix to measure the sparsified image matrix, this effectively reduces the storage space or transmission bandwidth of the cipher image. Given the limitations of 1*D* compressive sensing for image sampling, several researchers have proposed 2*D* compressive sensing for image encryption schemes. A visually meaningful dual-image encryption scheme using 2*D* compressive sensing and multi-rule *DNA* encoding was proposed by Huo et al. [39]. In this scheme, two plain images are permuted, diffused and 2*D* compressive sensing to obtain two private images, then these two private images are diffused again using *DNA* coding theory to obtain secret images and finally embedded to obtain the visually meaningful encrypted image. The application of 2*D* compressive sensing improves the encryption capability of the system by reducing the amount of data. In this scheme, we adopt a 2*D* compressive sensing, which allows us to efficiently sample the information of the image and significantly reduce the size of the transmitted image.

Given the above analysis, a color image encryption algorithm based on *CS* and blocked *DNA* encoding is introduced in this paper, which satisfies security and achieves effective compression. The contributions of this paper can be summarized in the following aspects:

(1) Combining *CS* and chaotic systems. Both the initial values of the chaotic systems and the construction of the measurement matrices depend on the plain image, which contributes to the well-robustness of our encryption scheme against the known-plaintext attacks and the chosen-plaintext attacks.

(2) A permutation method based on pseudo-random numbers and the Josephus problem is used to reduce the pixel correlation between and within the red, green, and blue components of a color image, which can effectively decrease the risk of statistic attacks and upgrade the security level.

(3) Unlike *DNA* addition, *DNA* subtraction, and *DNA XOR* operations based on binary computation, the *DNA* matrix sub-blocks of the compressed image are diffused by pseudo-random numbers derived from chaotic sequences. Besides, the encoding and decoding rules of the *DNA* sub-blocks of the compressed image are also related to pseudo-random numbers.

(4) *SVD* is applied to optimize the measurement matrix to enhance the quality of the reconstructed image.

The rest of this paper is organized as follows, Section II introduces basic knowledge of chaotic systems, compressive sensing, *DNA* operation, and the Josephus problem. The proposed encryption scheme is introduced in detail in Section III. The simulation results are presented in Section IV and the security performance is analyzed in Section V. And Section VI summarizes this paper.

II. PRELIMINARIES

A. COMPOSITE CHAOTIC SYSTEMS

As a result of the limited chaotic range and the less-than-ideal chaotic behavior of one-dimensional chaotic systems, [40] combined the existing one-dimensional chaotic systems including the Logistic map, Tent map, and Sine map in pairs to obtain the Logistic-Tent system, the Logistic-Sine system and Tent-Sine system, which are composite chaotic systems with a wider chaotic range and better chaotic behaviors. Their mathematical definitions are as follows.

The *LTS* is defined by *Eq.* (1).

$$x_{n+1} = \begin{cases} mod\big((rx_n(1-x_n)+(4-r)x_n/2),1\big) \\ \qquad x_i < 0.5 \\ mod\big((rx_n(1-x_n)+(4-r)(1-x_n)/2),1\big) \\ \qquad x_i \geq 0.5 \end{cases} \tag{1}$$

where r is an adjustable parameter with range of $(0, 4]$, $mod(\cdot)$ is the modulo operation, x_n is input chaotic sequence and x_{n+1} is output sequence, x_n,$x_{n+1} \in [0, 1]$.

The *LSS* is defined as follows.

$$y_{n+1} = \big(uy_n(1-y_n)+(4-u)\sin(\pi y_n)/4\big) \mod 1 \tag{2}$$

where control parameter $u \in (0, 4]$, y_n is input chaotic sequence and y_{n+1} is output sequence, y_n,$y_{n+1} \in [0, 1]$.

Eq. (3) is the definition of the *TSS*.

$$z_{n+1} = \begin{cases} \big(az_n/2+(4-a)\sin(\pi z_n)/4\big) \mod 1 \\ \qquad x_i < 0.5 \\ \big(a(1-z_n)/2+(4-a)\sin(\pi z_n)/4\big) \mod 1 \\ \qquad x_i \geq 0.5 \end{cases} \tag{3}$$

where, control parameter $a \in (0, 4]$, z_n is input chaotic sequence and z_{n+1} is output sequence, z_n,$z_{n+1} \in [0, 1]$.

B. COMPRESSIVE SENSING

Compressive sensing is a sampling technique that compresses data during sampling. It breaks the sampling theory of *Nyquist*, which was first introduced by Donoho et al. [34]. This allows us to reconstruct the original signal from the

TABLE 1. DNA encoding rules.

Rules	1	2	3	4	5	6	7	8
A	00	00	01	01	10	10	11	11
T	11	11	10	10	01	01	00	00
C	01	10	00	11	00	11	01	10
G	10	01	11	00	11	00	10	01

TABLE 2. Addition of DNA sequence.

+	A	T	G	C
A	C	G	T	A
T	G	C	A	T
G	T	A	C	G
C	A	T	G	C

TABLE 3. Diffusion rules of DNA sequence.

diffusion	A	C	T	G
1	A	C	T	G
2	C	G	A	T
3	G	T	C	A
4	T	A	G	C

TABLE 4. Inverse diffusion rules of DNA sequence.

Inverse diffusion	A	C	T	G
1	A	C	T	G
2	T	A	G	C
3	G	T	C	A
4	C	G	A	T

sparse signal with high accuracy using a reconstruction algorithm. Suppose X is a sparse signal of $N \times 1$, and the one-dimensional CS ($1DCS$) can be represented as the $Eq.$ (4).

$$Y = \Phi X = \Phi \Psi s \tag{4}$$

where Φ is a measurement matrix of size $M \times N (M<N)$, Ψ is sparse basis matrix of dimensions $N \times N$, s is the coefficient vector of signal X with only $K(K \ll N)$ non-zero elements, and Y is the compressed signal of size $M \times 1$. The popular sparsifying bases include Fourier, DCT, and wavelets. More generally, $1DCS$ is applied to compress $1D$ signals. In this case, in order for $1DCS$ to be able to compress a $2D$ image, it is necessary to convert the $2D$ image into a long vector and then sample this long vector [41]. It is represented by

$$y = \Phi vec(X) = \Phi x \tag{5}$$

where $\Phi \in \mathbb{R}^{M^{\Theta} \times N^{\Theta}}$ is a measurement matrix, $vec(\cdot)$ is a vectorized function that arranges $2D$ image in columns, X is $2D$ image of size $N \times N$, x is a long vector of size $N^2 \times 1$, and $y \in \mathbb{R}^{M^2 \times 1}$ is the compressed image. However, this faces two main challenges: high computational time complexity and large storage capacity of the measurement matrix. To make up for these shortcomings, in this paper, two-dimensional CS ($2DCS$) is used to compress the plain image. It is denoted by $Eq.$ (6).

$$Y = \Phi_{\Delta} X \Phi_{\Theta}^{T} \tag{6}$$

where Φ_{Δ} and Φ_{Θ} are measurement matrices of dimension $M \times N$, $X \in \mathbb{R}^{N \times N}$ and $Y \in \mathbb{R}^{M \times M}$ are the original $2D$ image and the compressed $2D$ image, respectively. In general, common measurement matrices are random *Gaussian* matrices, random *Bernoulli* matrices, partial *Hadamard* matrices, and circular measurement matrices. Due to their high storage space, many schemes [42], [43], [44] have proposed measurement matrices constructed based on chaotic systems. When using $2DCS$ sampling $2D$ image, the compression ratio (CR) is calculated by the following formula (7).

$$CR = \frac{M^2}{N^2} \tag{7}$$

where N and M are the image sizes before and after the image is compressed. Exact reconstruction of X from Y has to be done with the help of some reconstruction algorithms, such as match tracking (MP) [45], orthogonal matching pursuit (OMP) [46], smooth l_0 norm (Sl_0) [47]. Inspired by the project of [48], an iterative gradient projection reconstruction algorithm is adopted to reconstruct the plain image in this paper.

C. DNA OPERATION

1) DNA ENCODING AND DNA DECODING RULES

The four nucleic acid bases adenine (A), guanine (G), cytosine (C), and thymine (T) can construct a unique DNA sequence. There are total $4! = 24$ kinds of coding to encode a decimal integer in the range 0 to 255 into a DNA sequence [7]. However, since nucleic acid bases are complementary, that is, A with T and C with G are two complementary base pairs, respectively, only 8 coding rules are compatible with complementarity, as shown in Table. 1.

Based on the DNA encoding rules, the image can be converted into a DNA matrix. For example, assuming that the pixel value in the image is 45, converting it to an 8-bit binary sequence [00101101] can be encoded as a DNA sequence [$TGAC$] according to rule 7. DNA decoding is the reverse process of DNA encoding, that is, the DNA sequence can be decoded to the decimal number. The DNA sequence [$TGAC$] is decoded according to rule 7 to obtain the binary sequence [00101101], and its corresponding decimal number is 45. Therefore, the DNA matrix can be recovered as an image matrix. Frequently we use random number-based encoding and decoding rules to scramble the image, which can be effective anti-statistic attacks. Moreover, the addition operation of DNA is defined in Table. 2.

2) DNA DIFFUSION

Diffusion of DNA matrices is performed employing traditional binary-based operations of DNA addition, DNA

subtraction, and *DNA XOR*, which are inclined to be predictable and may consume excessive time. This paper adopts a *DNA* matrix diffusion mechanism based on chaotic systems [30]. We first iterate the *LTS* to obtain a sequence T, convert the sequence values to integers from 1 to 4, and then diffuse the *DNA* coding-based image sub-blocks based on this integer sequence. Since the diffusion process of the *DNA* coding-based image sub-blocks is completely randomized, this guarantees that the diffusion process is unpredictable to crack. The diffusion rules for *DNA* are listed in Table. 3 and Table. 4. For example, suppose there is a *DNA* sequence $[CGATGAAC]$ and an equal-length chaotic sequence [13241143], which can be converted into another *DNA* sequence $[CACGGATT]$ according to the diffusion rule. In the same way, *DNA* sequences can be recovered by *DNA* inverse diffusion rules.

D. JOSEPHUS PROBLEM

The Josephus problem is a problem that arises in computer science and mathematics, and we can describe it this way: given the number of people in a circle, a starting point, a direction, as well as a number to be skipped, and then sequentially choose the people in a certain position in the circle. To understand it better, an example is given: Suppose there is a six-person circle [125,241,69,97,223,168], the starting position is 6, and the skipped number is 3, a new Josephus can be generated clockwise sequence [241,223,69,125,97,168]. Fig. 1 shows the generation process of the Josephus sequence.

III. THE PROPOSED IMAGE ENCRYPTION AND DECRYPTION ALGORITHM

A. GENERATE INITIAL VALUES FOR CHAOTIC SYSTEMS AND JOSEPHUS PROBLEM

To enhance the dependency between the encryption scheme and the plain image, the *SHA*-384 hash value and entropy values of the plain image are utilized to calculate the initial values of the composite chaotic systems. The specific calculation steps are as follows:

Initially, we calculate the 384-bit hash value K of the plain image and convert K into 48 decimal numbers, K can be expressed as $K = K_1, K_2, \ldots, K_{48}$. Then calculate the entropy values s_1, s_2, s_3 of the three components of the color image R, G, and B. The entropy value calculation formula is as follows:

$$s = \sum_{i=0}^{2^n-1} P(m_i) \log_2 \frac{1}{P(m_i)} \tag{8}$$

where $P(m_i)$ is the probability that m_i occurs and $n = 8$ for an image with 256 gray levels.

The two sets of initial values x_1, r_1, x_2, r_2 for *LTS*, the two sets of initial values y_1, u_1, y_2, u_2 for *LSS*, the initial values z_1, a_1 for *TSS* and the initial values x_0, y_0 for the Josephus problem are obtained by *Eq.* (9).

$$\begin{cases} x_1 = \left(\dfrac{(k_1 + k_2 + k_3 + k_4 + k_5 + k_6)}{256} + t_1\right) mod \quad 1 \\ r_1 = \left(\dfrac{(k_7 + k_8 + k_9 + k_{10} + k_{11} + k_{12})}{256} + t_2\right) mod \quad 4 \\ x_2 = \left(x_1 + (s_1 + s_2)/2\right) mod \quad 1 \\ r_2 = \left(r_1 + (s_1 + s_3)/2\right) mod \quad 4 \\ y_1 = \left((k_{13} \oplus k_{14} \oplus \ldots \oplus k_{18})/256 + t_3\right) mod \quad 1 \\ u_1 = \left((k_{19} \oplus k_{20} \oplus \ldots \oplus k_{24})/256 + t_4\right) mod \quad 4 \\ y_2 = (y_1 + s_1 + s_2 + s_3) mod \quad 1 \\ u_2 = (u_1 + s_1 + s_2 + s_3) mod \quad 4 \\ z_1 = \left(\dfrac{(k_{25} + k_{26} + k_{27}) \oplus (k_{28} + k_{29} + k_{30})}{256} + t_5\right) mod \quad 1 \\ a_1 = \left(\dfrac{(k_{31} + k_{32} + k_{33}) \oplus (k_{34} + k_{35} + k_{36})}{256} + t_6\right) mod \quad 4 \\ x_0 = \dfrac{k_{37} + k_{38} + \ldots + k_{42}}{6} \\ y_0 = \dfrac{(k_{43} + k_{44} + k_{45}) \oplus (k_{46} + k_{47} + k_{48})}{3} \end{cases} \tag{9}$$

where t_1, t_2, t_3, t_4, t_5 and t_6 are variable extra control parameters that take values between 1 and 4 and are used to calculate the initial value of the composite chaotic systems. When encrypting the same plain image, even with the same *SHA* value, differences in these parameters result in a different encrypted image, which effectively improves security.

B. CONSTRUCTION OF THE MEASUREMENT MATRIX

In this paper, $2DCS$ is adopted, hence two measurement matrices Φ_Δ and Φ_Θ are required to be constructed to sample the plain image. Suppose the plain image is of size $N \times N$, the construction process is as follows:

Step 1. Calculate the initial values x_1, r_1, x_2, r_2 of the *LTS* by *Eq.* (9).

Step 2. Iterate *LTS* $MN + N_0$ times ($N_0 \geq 1000$) with x_1 and r_1, x_2 and r_2 respectively. To avoid harmful effects, the first N_0 values are discarded and obtaining two chaotic sequences of length MN, $P_1 = \{p_1, p_2, \ldots, p_{MN}\}$ and $Q_1 = \{q_1, q_2, \ldots, q_{MN}\}$. And $M = \sqrt{CR \times N^2}$, CR is the compression ratio of the plain image.

Step 3. Reshape the chaotic sequences P_1 and Q_1 in a column-major manner to obtain matrices P_2 and Q_2, which can be denoted via *Eq.* (10).

$$\begin{cases} P_2 = \begin{pmatrix} p_1 & p_{M+1} & \cdots & p_{M(N-1)+1} \\ p_2 & p_{M+2} & \cdots & p_{M(N-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ p_M & P_{2M} & \cdots & p_{MN} \end{pmatrix} \\ Q_2 = \begin{pmatrix} q_1 & q_{M+1} & \cdots & q_{M(N-1)+1} \\ q_2 & q_{M+2} & \cdots & q_{M(N-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ q_M & q_{2M} & \cdots & q_{MN} \end{pmatrix} \end{cases} \tag{10}$$

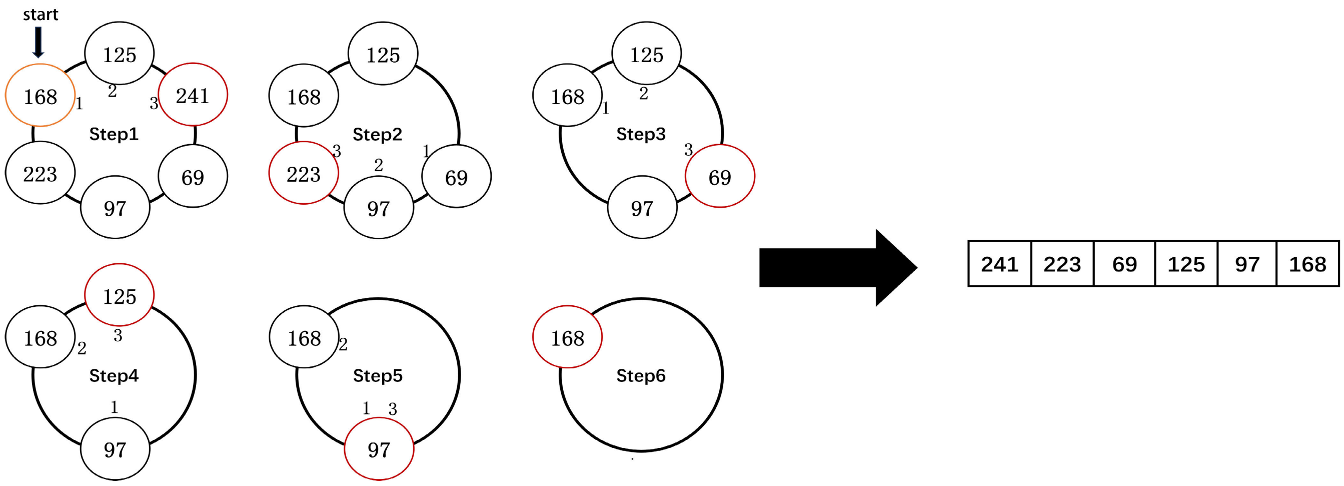


FIGURE 1. A example of Josephus problem and its resulting vector.

Step 4. The measurement matrices Φ_Δ and Φ_Θ are obtained by optimizing P_2 and Q_2 by means of *SVD*. Taking P_2 as an example, the optimization process is as follows [49]:

Step 4.1. The *SVD* is conducted on the matrix P_2 with $P_2 = U\Sigma V^T$, where U is the orthogonal matrix of $M \times M$, Σ is the diagonal matrix of $M \times N$, and V^T is the orthogonal matrix of $N \times N$, $(\cdot)^T$ denotes the matrix transpose.

Step 4.2. The mean *var* of the diagonal matrix Σ is calculated to obtain the number of elements f in the diagonal matrix greater than or equal to *var*.

Step 4.3. Generate a matrix H of size $M \times N$ whose element values are all equal to 1. Then multiply its first f columns by a weighting factor t. The ideal value of t is 5 [50].

Step 4.4. The measurement matrix P_2 is dot-multiplied by matrix H to obtain P_2'.

Step 4.5. Manipulate *SVD* on matrix P_2', that is, $P_2' = U_1\Sigma_\Delta V_\Delta^T$. Then set all elements of Σ_Δ to 1 to gain Σ_Δ' [50].

Step 4.6. The optimized measurement matrix Φ_Δ is obtained by calculating $\Phi_\Delta = U_\Delta \Sigma_\Delta' V_\Delta^T$. Likewise, measurement matrix Φ_Θ can be obtained by optimizing Q_2.

C. COMPLETE ENCRYPTION ALGORITHM

The flowchart of the proposed encryption scheme is presented in Fig. 2, if the color plain image I is of size $N \times N$, and the detailed encryption steps are given below.

Step 1. Compute the *SHA*-384 hash value K of I.

Step 2. I is decomposed into red, green, and blue components with information entropy values s_1, s_2, and s_3 by *Eq.* (8) are calculated. These three components are denoted as R, G, and B, respectively.

Step 3. K, s_1, s_2 and s_3 are combined with external parameters by *Eq.* (9) to compute the initial values of the chaotic systems and the Josephus problem.

Step 4. Generate measurement matrices Φ_Δ and Φ_Θ of size $M \times N$, and the detailed generation process is in Section III-B.

Step 5. Sampling R, G, and B to obtain R_1, G_1, and B_1 of size $M \times M$. The measurement process is as *Eq.* (11).

$$\begin{cases} R_1 = \Phi_1 R \Phi_2^T \\ G_1 = \Phi_1 G \Phi_2^T \\ B_1 = \Phi_1 B \Phi_2^T \end{cases} \tag{11}$$

Step 6. The integer matrices R_2, G_2, and B_2 are acquired by quantizing R_1, G_1, and B_1. As an example, the quantization of R_1 is performed by *Eq.* (12).

$$R_2 = round(\frac{255 \times (R_1 - min)}{max - min}) \in [0, 255] \tag{12}$$

where *min* and *max* are the minimum and maximum values of measurement value matrix R_1.

Step 7. To mutually permute the components, the Josephus problem is used to displace R_2, G_2, and B_2, the principle of which has been presented in Section II-D. R_3, G_3, and B_3 are obtained by the following substitution operations.

Step 7.1. Combining R_2, G_2 and B_2 yields $S = [R_2,G_2,B_2]$ with dimension $M \times 3M$.

Step 7.2. A Josephus sequence J of length $3M$ is generated using x_0 as the starting position and y_0 as the skipping number, and then manipulate the following process to permute S.

$$S(:, i) = S(:, J(i)), \quad i = 1, 2, \ldots, 3M \tag{13}$$

Step 7.3. Decomposing S yields R_3, G_3, and B_3, whose size is $M \times M$, the process is as below.

$$\begin{cases} R_3 = S(:, 1 : M) \\ G_3 = S(:, M + 1 : 2M) \\ B_3 = S(:, 2M + 1 : 3M) \end{cases} \tag{14}$$

Step 8. Iterate *LSS* $L + N_0$ times by utilizing y_1, u_1 and y_2, u_2 respectively. To avoid the transient effect, remove the former N_0 numbers to obtain sequences $E = \{e_1, e_2, \ldots, e_L\}$ and $D = \{d_1, d_2, \ldots, d_L\}$. $L = (M/n)^2$ is the number of blocks and n is defined as the sub-block size.

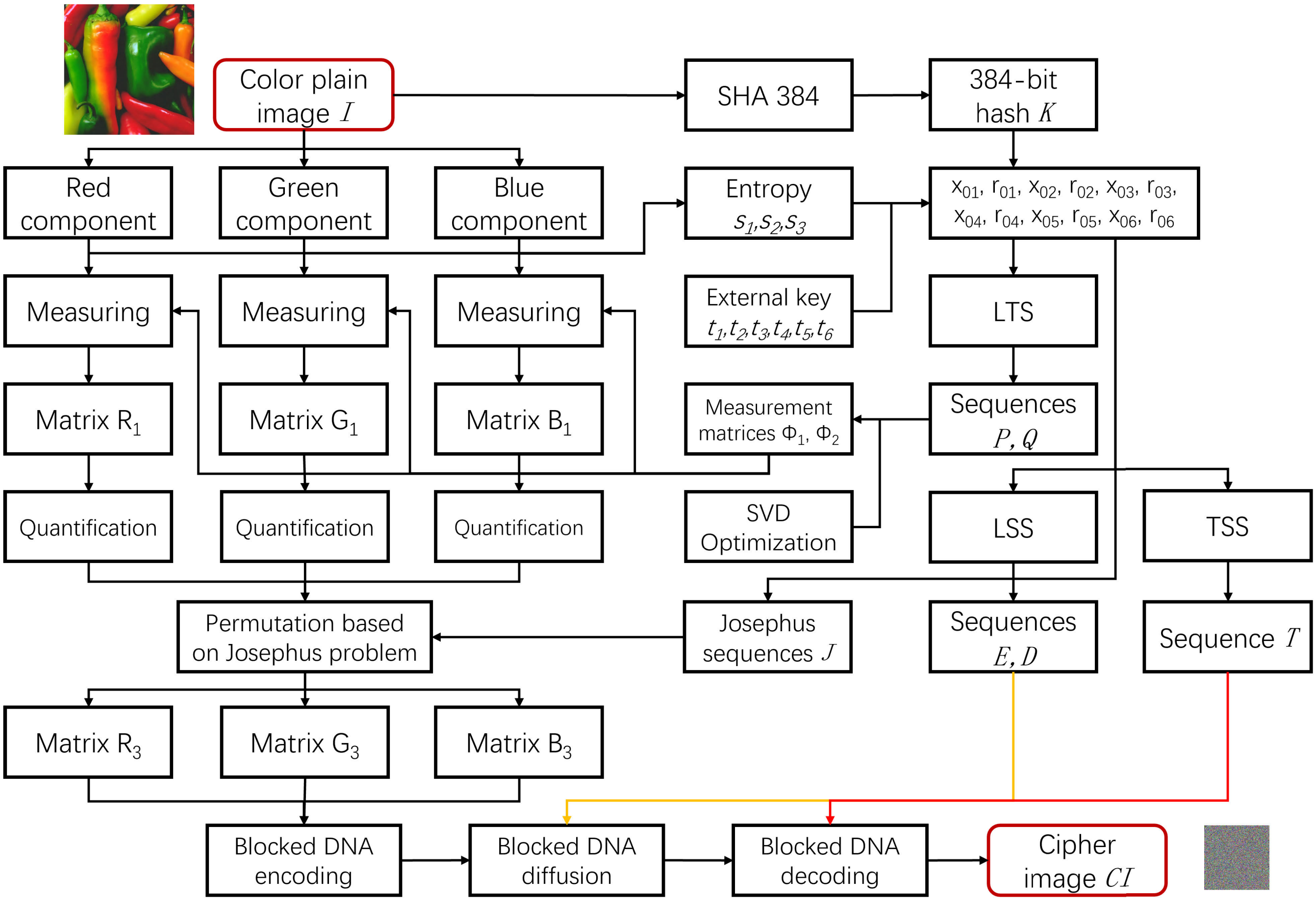


FIGURE 2. **The flow diagram of the proposed encryption scheme.**

Step 9. According to *Eq.* (15), the elements in sequences $\{E\}$ and $\{D\}$ are converted to integers in the range 1 to 8. They are then applied to sub-block *DNA* encoding and *DNA* decoding.

$$E_i{}' = mod(round(E_i \times 1000), 8) + 1 \tag{15}$$

Step 10. The initial values z_1 and a_1 are applied to iterate *TSS* $L + N_0$ times, and the former N_0 values are discarded to obtain the sequence $T = \{t_1, t_2, \ldots, t_L\}$. Then its elements are transformed via *Eq.* (16).

$$T_i{}' = mod(round(T_i \times 1000), 4) + 1 \tag{16}$$

where the value range of the sequence $\{T\}$ obtained again is between 1 and 4.

Step 11. *DNA* permutation and diffusion were performed on R_3, G_3, B_3 obtained in Step 7. Take R_3 as an instance and proceed as follows:

Step 11.1. Divide R_3 into L sub-blocks of size $n \times n$ in the order from top to bottom and left to right.

Step 11.2. All sub-blocks of R_3 are operated as follows:

Step 11.2.1. The sub-block is encoded with the sequence $\{E\}$. The values of $\{E\}$ range from 1 to 8, corresponding to the 8 *DNA* encoding rules. *Eq.* (17) represents the *DNA* coding process for each sub-block.

$$R_4(i) = Encode(R_3(i), E(i)), \quad i = 1, 2, \ldots, L \tag{17}$$

where $R_3(i)$ is the *i*-th sub-block of the image matrix to be encoded, R_4 denotes the image matrix after *DNA* encoding, $R_4(i)$ represents the *i*-th coding sub-block of the *DNA* matrix.

Step 11.2.2. First-round DNA diffusion. The current sub-block *DNA* matrix and the previous sub-block *DNA* matrix are added according to *Eq.* (18). DNA addition operations follow Table. 2. $R_4(i)$ and $R_4(i-1)$ denote the current sub-block *DNA* matrix and the previous sub-block *DNA* matrix.

$$\begin{cases} R_4(i) = R_4(i) + R_4(L) & (i = 1) \\ R_4(i) = R_4(i) + R_4(i-1) & (i > 1) \quad i = 1, 2, \ldots, L \end{cases} \tag{18}$$

Step 11.2.3. Second-round DNA diffusion. *DNA* diffusion is performed for each sub-block *DNA* matrix of $R_4(i)$. The diffusion process is described by *Eq.* (19).

$$R_5(i) = Diffusion(R_4(i), T(i)), \quad i = 1, 2, \ldots, L \tag{19}$$

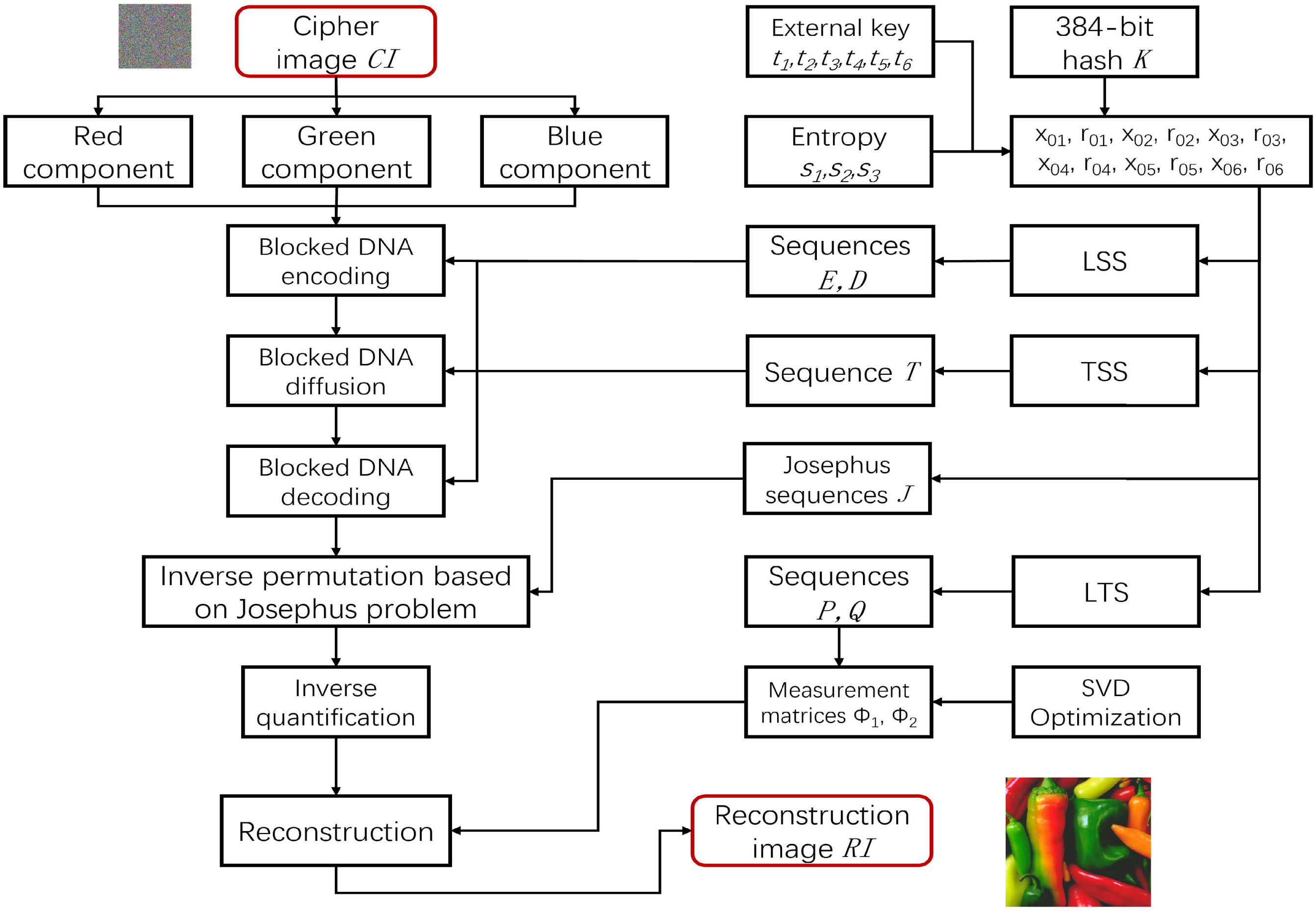


FIGURE 3. The flow chart of the proposed decryption scheme.

where $R_5(i)$ is the i-th sub-block of the DNA matrix after DNA diffusion. The detailed DNA diffusion scheme has been described in section II-C2.

Step 11.2.4. Decode the sub-block DNA matrix $R_5(i)$ using the quantized chaotic sequence $\{D\}$ and convert it to sub-block decimal matrix $R_6(i)$, which can be described as *Eq*. (20).

$$R_6(i) = Decode(R_5(i), D(i)), \quad i = 1, 2, \ldots, L \tag{20}$$

Similarly, G_6 and B_6 can also be obtained through the above encryption process.

Step 12. Integrating R_6, G_6, and B_6 to generate the final cipher image CI.

D. COMPLETE DECRYPTION ALGORITHM

The decryption process is illustrated in Fig. 3. It is an inverse process of the encryption process, and the detailed decryption algorithm is described below:

Step 1. The initial values of chaotic systems and the Josephus problem are calculated with the keys K, s_1, s_2, and s_3 and extra parameters t_1, t_2, t_3, t_4, t_5, t_6.

Step 2. Iterate LSS with initial values y_1 with u_1 and y_2 with u_2. Chaotic sequences $\{E\}$ and $\{D\}$ of length L are obtained for determining DNA encoding rules and DNA decoding rules. The chaotic sequence $\{T\}$ of length L is also obtained by iterating over the TSS with initial values z_1 and a_1 for the DNA diffusion.

Step 3. Cipher image CI of dimension $M \times M$ is decomposed into three components R, G, and B.

Step 4. The inverse DNA substitution and inverse DNA diffusion processes of R, G, and B are carried out according to the following steps. An example is taken for R.

Step 4.1. Divide R into L sub-blocks of size $n \times n$ and arrange them in order from top to bottom and from left to right.

Step 4.2.The block-based DNA encoding is executed, and this process is expressed as *Eq*. (21). where $R_1(i)$ denotes the i-th sub-block of the DNA matrix.

$$R_1(i) = Encode(R(i), D(i)), \quad i = L, L-1, \ldots, 1 \tag{21}$$

Step 4.3. According to *Eq*. (22), the inverse DNA diffusion is implemented on R_1 by means of sequence $\{T\}$.

$$R_2(i) = INdif(R_1(i), T(i)), \quad i = L, L-1, \ldots, 1 \tag{22}$$

TABLE 5. Mean time for DNA diffusion for different block sizes.

n	8	16	32	64
Mean time(s)	0.0199	0.0802	0.3151	1.4440

TABLE 6. PSNR of the reconstructed images.

CR	PSNR			
	Peppers	Peppers2	Airplane	Tiffany
0.25	37.1197	31.3243	32.4741	34.8178
0.5	40.2718	33.8798	37.6650	39.5580
0.75	42.994	37.2358	41.9846	43.7637

where $INdif\ (\cdot)$ is the inverse DNA diffusion described in Section II-C2 and R_2 is the resulting DNA matrix after inverse diffusion.

Step 4.4. The secondary inverse diffusion of the DNA matrix R_2 is conducted based on $Eq.$ (23).

$$\begin{cases} R_3(i) = R_2(i) + R_2(L) & i = 1 \\ R_3(i) = R_2(i) + R_2(i-1) & i > 1 \quad i = L, L-1, \ldots, 1 \end{cases} \tag{23}$$

where $R_3(i)$ represents the i-th sub-block DNA matrix after re-inverse diffusion.

Step 4.5. The decimal matrix R_4 of size $M \times M$ is obtained by decoding R_3, the decoding rule follows the pseudo-random sequence $\{E\}$. It is expressed as $Eq.$ (24).

$$R_4(i) = Decode(R_3(i), E(i)) \quad i = L, L-1, \ldots, 1 \tag{24}$$

Step 5. With step 4, the inverse scrambling and diffusion matrices G_4 and B_4 of the green and blue components are also obtained. Then combine R_4, G_4, and B_4 into a matrix SC of size $M \times 3M$.

Step 6. Generate a Josephus sequence J of length $3M$ with x_0 and y_0 as initial values and then perform an inverse permutation of SC based on this Josephus sequence, which can be described as

$$SC(:, J(i)) = SC(:, i) \quad i = 1, 2, \ldots, 3M \tag{25}$$

where $SC(:, i)$ denotes the i-th column of SC.

Step 7. Three matrices R_5, G_5, B_5 are obtained by decomposing SC with the following formula.

$$\begin{cases} R_5 = SC(:, 1 : M) \\ G_5 = SC(:, M + 1 : 2M) \\ B_5 = SC(:, 2M + 1 : 3M) \end{cases} \tag{26}$$

Step 8. The initial values x_1 with $r_1 and x_2$ with r_2 are used to iterate the LTS, respectively, generating two measurement matrices Φ_Δ and Φ_Θ. The construction of the measurement matrices is illustrated in Section III-B.

Step 9. R_6, G_6, and B_6 are obtained by performing inverse quantization on R_5, G_5, and B_5 according to the following $Eq.$ (27).

$$R_6 = \frac{R_5 \times (max - min)}{255} + min \tag{27}$$

TABLE 7. Compression performance of different algorithms.

CR	Ref. [51]	Ref. [52]	Ref. [53]	Ours
0.25	<30	32.7130	14.0119	37.1197
0.5	≈35	35.1721	26.4965	40.2718
0.75	<40	38.4215	31.7121	42.9942

TABLE 8. The SSIMs between plain images and reconstructed images.

Image	R component	G component	B component	SSIM(%)
Peppers	99.98	99.90	99.99	
Sailboat	99.89	99.96	99.98	
Tiffany	99.91	99.93	99.96	
Peppers2	99.92	99.92	99.94	
Baboon	99.97	99.90	99.92	

Step 10. Inspired by [45], a two-dimensional reconstruction algorithm is employed to reconstruct R_6, G_6, and B_6 to obtain R_7, G_7, and B_7 of size $N \times N$.

Step 11. The reconstructed color image RI is obtained by combining R_7, G_7, and B_7.

IV. EXPERIMENTAL SIMULATION RESULTS

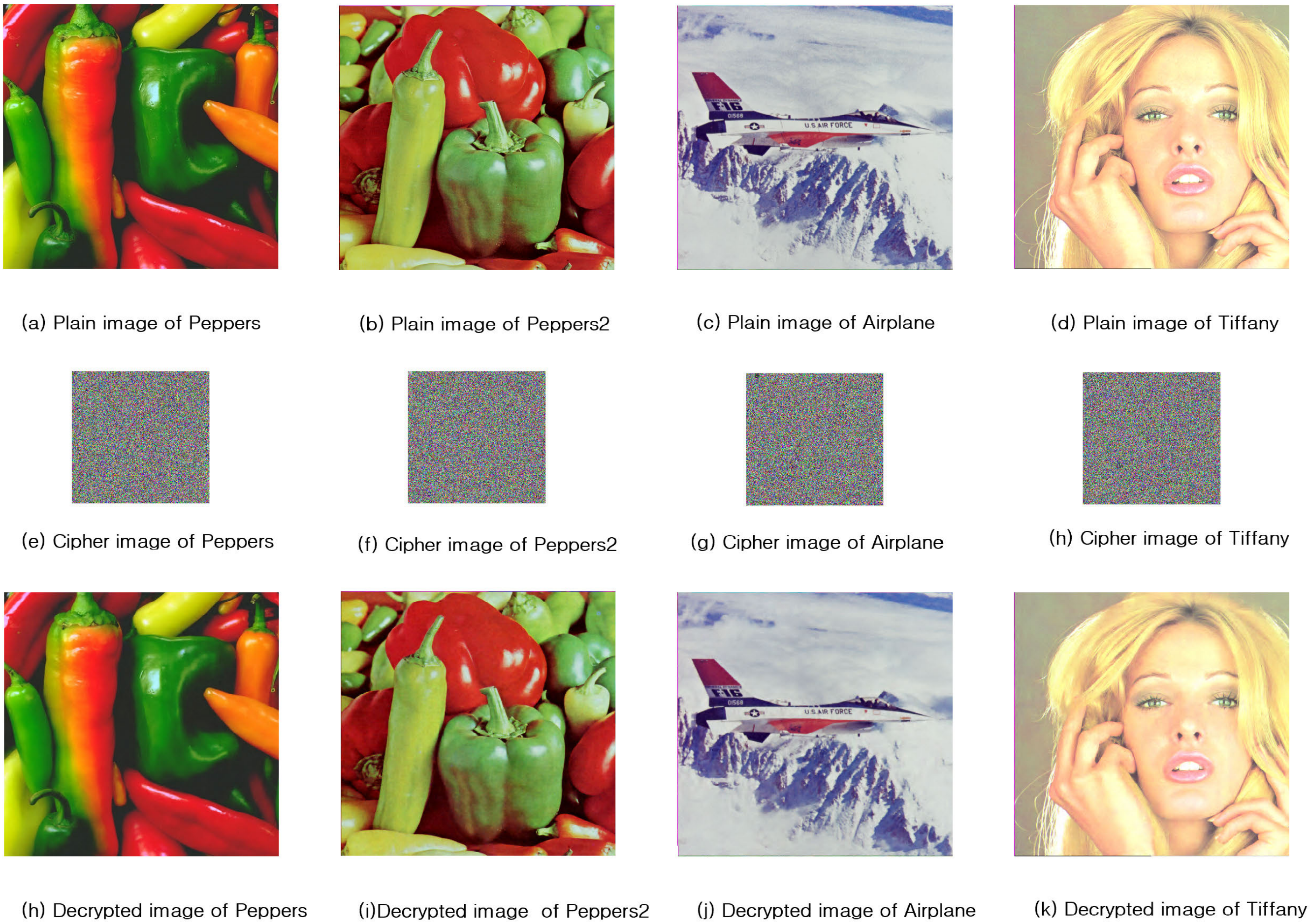
This section presents the simulation results of the proposed algorithm, and all experiments are performed on a personal computer environment: Windows 11, 3.2Ghz CPU, 16GB RAM. The compilation software is MATLAB R2021a.

A. ENCRYPTION AND DECRYPTION RESULTS

In this section, four color images Peppers (512 × 512), Peppers2 (512 × 512), Airplane (512 × 512), and Tiffany (512 × 512), are used as plain images. The additional self-defined parameters t_1, t_2, t_3, t_4, t_5, and t_6 used to compute the initial values of the chaotic systems in Section III-A are randomly determined to be 0.25, 2.78, 0.43, 3.72, 0.39, and 1.90, respectively. As in Table 5, the mean time of the diffusion process for each sub-block for different blocking cases is given, which determines the image sub-block size of 8 for this simulation experiment(i.e., n = 8). And their simulation results are presented in Fig. 4. Since $2DCS$ is adopted in this image encryption scheme, the size of the encrypted image is one-fourth of the plain image, which significantly saves the bandwidth in the transmission of the cipher image. From Fig. 4, it is observed that the cipher images are noisy, meaning that it is almost impossible for an attacker to obtain any valuable information from the cipher image. The peak signal-to-noise ratio ($PSNR$) is an objective criterion used to evaluate image quality and is defined as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}$$
$$MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} ||I_1(i,j) - I_2(i,j)||^2 \tag{28}$$

where MSE is the mean square error between the plain image $I_1(i,j)$ and the reconstructed image $I_2(i,j)$, and m and



(a) Plain image of Peppers
(b) Plain image of Peppers2
(c) Plain image of Airplane
(d) Plain image of Tiffany
(e) Cipher image of Peppers
(f) Cipher image of Peppers2
(g) Cipher image of Airplane
(h) Cipher image of Tiffany
(h) Decrypted image of Peppers
(i)Decrypted image of Peppers2
(j) Decrypted image of Airplane
(k) Decrypted image of Tiffany

FIGURE 4. **Simulation results.**

TABLE 9. **Comparison of key spaces.**

Algorithm	Ref. [37]	Ref. [49]	Ref. [55]	Ref. [56]	Ours
Key space	2^{372}	2^{215}	2^{288}	2^{256}	2^{465}

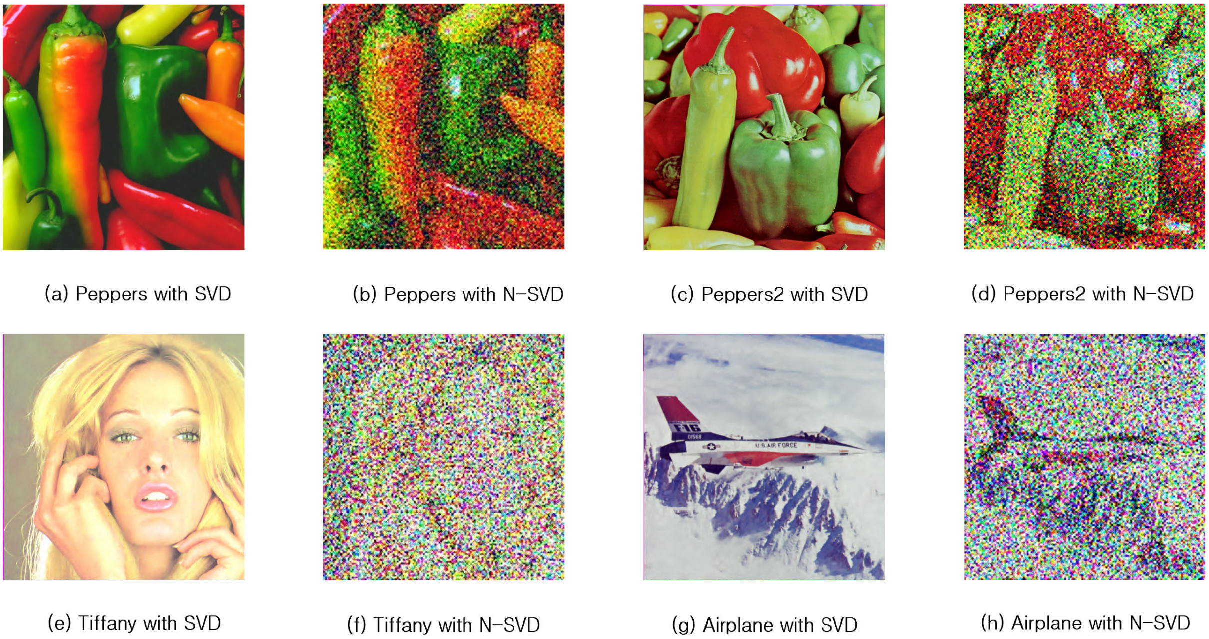
n represent the height and width of the image. As shown in Table. 6, the *PSNR* values of the reconstructed images increase continuously as *CR* increases from 0.25, 0.5 to 0.75 and are all greater than 31. This means that the reconstructed image is extremely similar to the plain image. Table. 7 presents the *PSNR* results of the color Peppers image under different *CR*, as well as compares them with the existing algorithms [51], [52], [53]. From Table. 7, we can see that the quality of our reconstructed images is better than other algorithms.

To further demonstrate the effectiveness of the present algorithm, the Structural Similarity Index (*SSIM*) is used to evaluate the similarity between the three component channels of the color plain image and the reconstructed image. The *SSIM* takes into account the luminance, contrast, and structural information of the image, and therefore provides a more comprehensive evaluation. The value domain of the *SSIM* is usually between $[-1, 1]$, where 1 means that the two images are identical, while 0 means that the two images do not have any similarity. In the case of image encryption, ideally, the *SSIM* value of the encrypted image and the original image is close to 1, indicating that the encryption algorithm has not significantly altered the structural information of the image and proving the effectiveness of the algorithm. *SSIM* is calculated as in *Eq.* (29).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{29}$$

where x and y represent the plain and encrypted images, respectively, μ_x and μ_y are the mean values of the pixels of x and y, respectively, as well as σ_x^2 and σ_y^2 denote the variances of x and y, respectively, and σ_{xy} is the covariance between x and y, C_1 and C_2 are stabilization constants to prevent division by zero. Ordinarily, $C_1 = (K_1L)^2$, $C_2 = (K_2L)^2$, K_1 and K_2 are default values set to 0.01 and 0.03. They control the visibility of the first and second order statistics in the *SSIM* index. L is the dynamic range of the pixel values (usually 255 for an 8-bit image).

The *SSIM* values of the plain images and the reconstructed images are tested and displayed in Table. 8, it can be



(a) Peppers with SVD (b) Peppers with N–SVD (c) Peppers2 with SVD (d) Peppers2 with N–SVD

(e) Tiffany with SVD (f) Tiffany with N–SVD (g) Airplane with SVD (h) Airplane with N–SVD

FIGURE 5. The effect of the SVD-based measurement matrix on the reconstructed image.

TABLE 10. NPCR of the cipher images.

Image	Changed key	NPCR(%)
Fig. 6(a)	–	0
Fig. 6(b)	$t_1 + 10^{-14}$	99.61
Fig. 6(c)	$t_3 + 10^{-14}$	99.47
Fig. 6(d)	$t_5 + 10^{-14}$	96.95
Fig. 6(e)	–	0
Fig. 6(f)	$t_1 + 10^{-14}$	99.62
Fig. 6(g)	$t_3 + 10^{-14}$	99.62
Fig. 6(h)	$t_5 + 10^{-14}$	96.96

TABLE 11. NPCR of the reconstructed images.

Image	Changed key	NPCR(%)
Fig. 7(a)	–	0
Fig. 7(b)	k_1	99.61
Fig. 7(c)	$t_6 + 10^{-14}$	99.53
Fig. 7(d)	–	0
Fig. 7(e)	k_1	99.15
Fig. 7(f)	$t_6 + 10^{-14}$	99.27

observed that the *SSIM* values of the three components of the plain images and the corresponding three components of the reconstructed images are very close to 1, which proves the effectiveness of the proposed algorithm.

B. INFLUENCE OF SVD OPTIMIZATION ON RECONSTRUCTED IMAGES

Furthermore, Fig. 5 illustrates the impact of *SVD*-based measurement matrix optimization on image reconstruction. Fig. 5(a), (c), (e), and (g) are reconstructed images with *SVD*-based measurement matrices and they are almost identical to their plain images. Fig. 5(b), (d), (f), and (h) are the reconstructed images with non-*SVD*-based (N-SVD) measurement matrices, and their sharpness is not sufficient to recognize the details of their plain images. It is observed that the optimized measurement matrices significantly improve the quality of the reconstructed images.

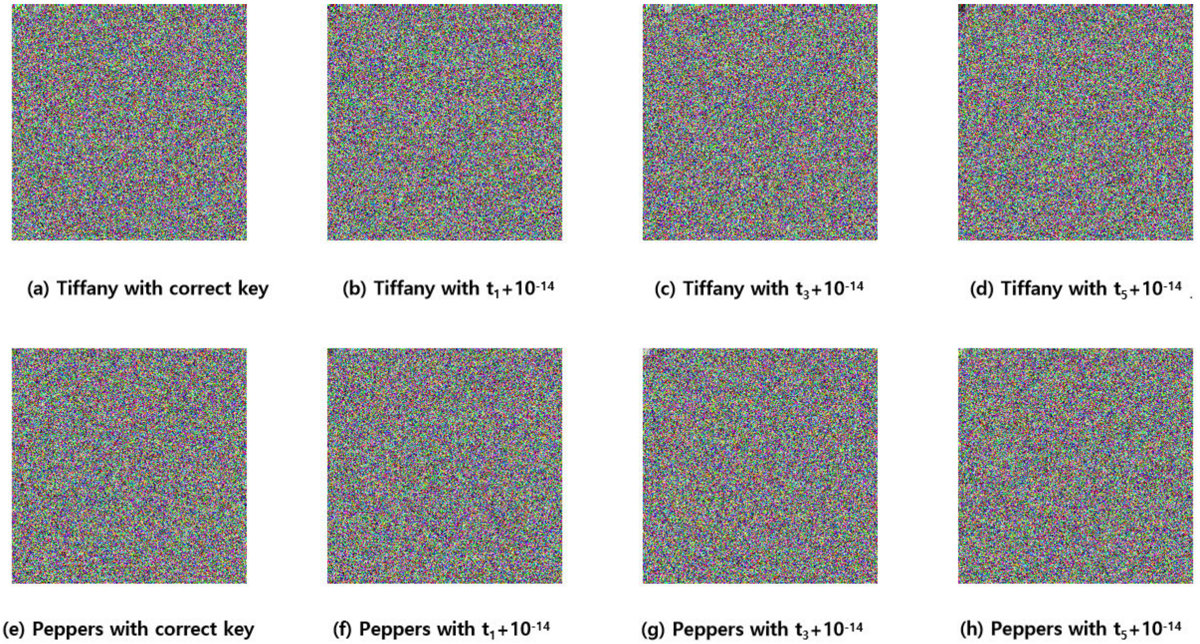
V. PERFORMANCE ANALYSIS

A. KEY SPACE ANALYSIS

In cryptography, key space is an important performance indicator of an encryption algorithm. A cryptosystem with a large key space tends to resist brute force very well. Ideally, the keys pace should be more than 2^{100}. The secret keys in this scheme mainly include (1) The 384-bit hash value K of the plain image. (2) the given parameters t_1, t_2, t_3, t_4, t_5, and t_6. (3) the information entropy s_1, s_2, s_3 of R, G, B. Besides, the *max* and *min* of the quantization process can also be used as secret keys. We set the initial values of the Josephus problem as the public keys. If the precision of the computer is 10^{14}, then the key space of our algorithm is about $(10^{14})^{10} \approx 2^{465} \gg 2^{100}$. When considering only the 384-bit hash as a key, our key space is already much larger than 2^{100}. Therefore, the key space of our encryption scheme can completely disable brute force attacks. Additionally, as shown in Table. 9, a key space comparison between our scheme and existing encryption algorithms is performed.

B. KEY SENSITIVITY ANALYSIS

A secure encryption algorithm should be sensitive to the key. Even a tiny change in the secret key can significantly change the final encryption or decryption result. In this section, we evaluate the key sensitivity of our scheme by making slight modifications to the secret keys: the hash value K of the plain image, and additional self-defined parameters



(a) Tiffany with correct key

(b) Tiffany with t_1+10^{-14}

(c) Tiffany with t_3+10^{-14}

(d) Tiffany with t_5+10^{-14}

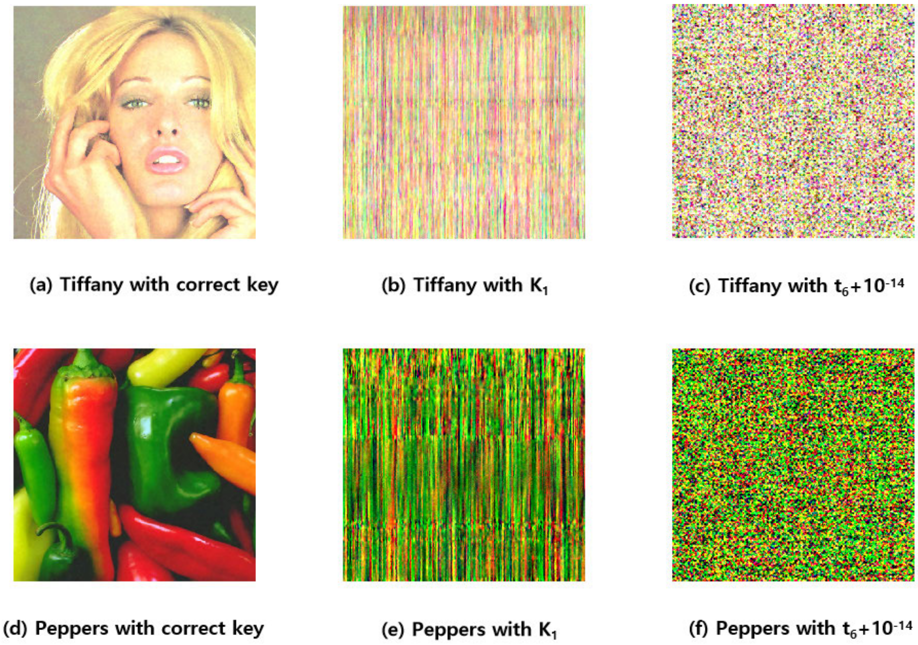
(e) Peppers with correct key

(f) Peppers with t_1+10^{-14}

(g) Peppers with t_3+10^{-14}

(h) Peppers with t_5+10^{-14}

FIGURE 6. Key sensitivity results in encryption process .



(a) Tiffany with correct key

(b) Tiffany with K_1

(c) Tiffany with t_6+10^{-14}

(d) Peppers with correct key

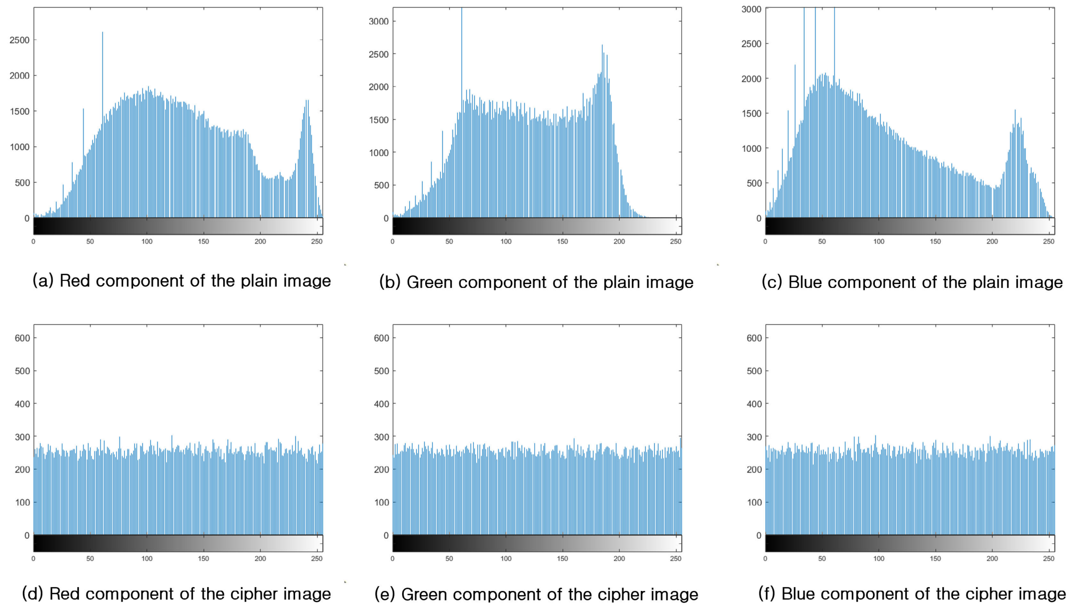
(e) Peppers with K_1

(f) Peppers with t_6+10^{-14}

FIGURE 7. Key sensitivity results in decryption process.

t_1, t_3, t_5, t_6 of the initial values of the chaotic systems. The test color images are Tiffany (512 × 512) and Peppers (512 × 512).

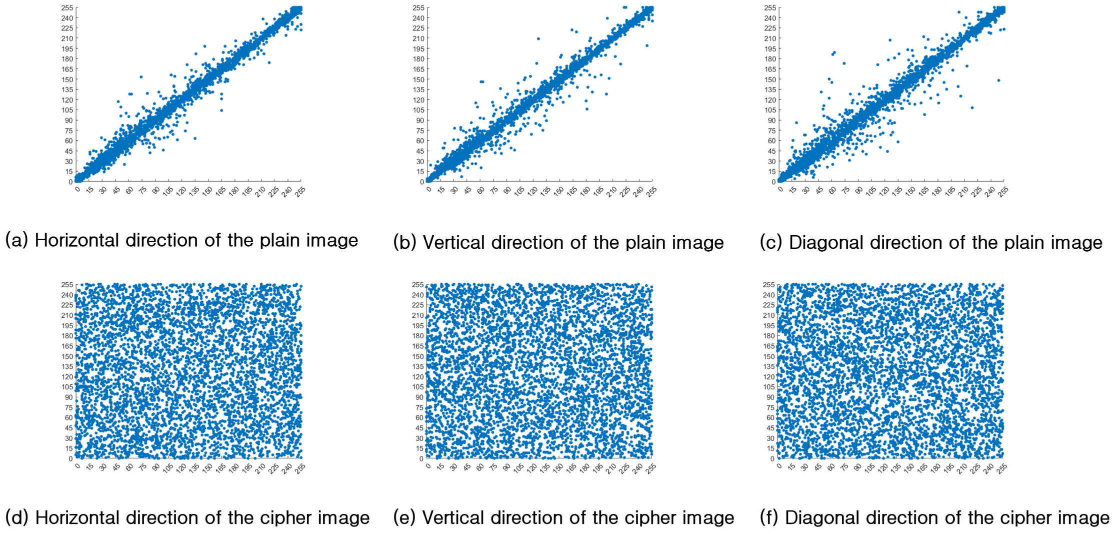
We change t_1, t_3, t_5, t_6 by adding 10^{-14} to gain t'_1, t'_3, t'_5, t'_6. K_1 is obtained by changing only one bit of K. These changed keys are used in the encryption and decryption phases of the test images. As shown in Fig. 6, (a) and (e) are cipher images with the correct key, and (b)-(d) with (f)-(h) are the respective cipher images obtained with t'_1, t'_3, and t'_5. K_1 and t'_6 are used in the decryption process to acquire different decrypted images, as shown in Fig. 7. As seen from Fig. 6 and Fig. 7, our encryption scheme is highly sensitive to keys. The encrypted images with the slightly modified keys are completely different from the encrypted images with the correct keys. Likewise, in the decryption stage, the decrypted images with the changed key are almost distinct



(a) Red component of the plain image (b) Green component of the plain image (c) Blue component of the plain image

(d) Red component of the cipher image (e) Green component of the cipher image (f) Blue component of the cipher image

FIGURE 8. The Histograms of plain image Baboon and its cipher image.



(a) Horizontal direction of the plain image (b) Vertical direction of the plain image (c) Diagonal direction of the plain image

(d) Horizontal direction of the cipher image (e) Vertical direction of the cipher image (f) Diagonal direction of the cipher image

FIGURE 9. Distribution of adjacent pixels of the plain image Peppers and its cipher images.

from Fig. 7(a) and (d), which are decrypted by the correct keys.

In addition, the number of pixel change rate (*NPCR*) is the percentage of different pixel numbers between two images and is to measure the difference between two images. The *NPCR* is defined by *Eq*. (30).

$$NPCR = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j) \times 100\%$$

$$D(i,j) = \begin{cases} 0 & C_1(i,j) = C_2(i,j) \\ 1 & C_1(i,j) \neq C_2(i,j) \end{cases} \tag{30}$$

where $C_1(i,j)$ and $C_2(i,j)$ denote pixels at the same position in two different images, and m, n are the width and height of the image respectively. The closer the *NPCR* value is to 1 means that the two images are not identical. The *NPCR* values of Fig. 6(b)-(d) and Fig. 6(a), Fig. 6(f)-(h) and Fig. 6(e), Fig. 7(b)-(c) and Fig. 7(a), Fig. 7(e)-(f) and Fig. 7(d) are calculated in Table. 10 and Table. 11. Their *NPCR* values are almost close to 1, indicating the high-level key sensitivity of the encryption and decryption processes.

C. HISTOGRAM ANALYSIS

The histogram is an essential statistical feature for evaluating an image encryption scheme. The histogram of an encrypted image should be uniformly distributed for a secure encryption scheme. This section analyzes the histogram of the color

TABLE 12. Correlation coefficients of plain images and cipher images.

Image	Direction	Correlation coefficients					
		Red	Green	Blue	Red	Green	Blue
		Plain image			Cipher image		
Peppers	Horizontal	0.9967	0.9943	0.9661	0.0197	-0.0087	0.0142
	Vertical	0.9964	0.9940	0.9724	0.0027	0.0069	0.0145
	Diagonal	0.9944	0.9900	0.9429	-0.0066	0.0029	0.0158
Peppers2	Horizontal	0.9698	0.9847	0.9669	-0.0079	0.0132	-0.0361
	Vertical	0.9655	0.9840	0.9692	0.0117	0.0106	0.0058
	Diagonal	0.9623	0.89751	0.9482	0.0039	-0.0054	-0.0134
Airplane	Horizontal	0.9572	0.9654	0.9370	-0.0197	0.0080	0.0108
	Vertical	0.9737	0.9651	0.9622	0.0099	-0.0041	-0.0092
	Diagonal	0.9356	0.9369	0.9151	0.0053	-0.0052	0.0239
Tiffany	Horizontal	0.9325	0.9339	0.9302	-0.0102	-0.0064	-0.0179
	Vertical	0.9449	0.8781	0.8948	0.0113	0.0061	0.0080
	Diagonal	0.8929	0.8423	0.8405	0.0027	-0.0220	-0.0114

TABLE 13. Comparison with other studies on correlation coefficient.

Algorithms	Ref [37]	Ref [51]	Ref [52]	Ref [57]	Ours
Horizontal	0.0082	0.0004	0.0017	0.0082	-0.0017
Vertical	0.0006	0.0001	0.0008	0.0034	0.0004
Diagonal	0.0035	0.0019	0.0014	0.0041	0.0064

Baboon image, Fig. 8(a)-(c) are the histograms of the three channels of the plain image and Fig. 8(d)-(f) are the histograms of the three channels of the cipher image. The histograms of the three components of the plain image are non-uniform and different from each other. On the contrary, the histograms of the cipher image are smooth and uniform. Therefore, our algorithm turns out to be robust to histogram attacks.

D. CORRELATION COEFFICIENT ANALYSIS

An image typically has a high correlation between adjacent pixels, which an attacker may exploit to analyze the entire image to crack the encryption algorithm. Consequently, an encrypted image with low adjacent pixel correlation is also a requirement for a secure image encryption algorithm. Mathematically defined, the correlation coefficient r can be calculated as

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{31}$$

where $cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$, $D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$, $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$. $cov(\cdot)$ denotes covariance, $D(\cdot)$ is variance, $E(\cdot)$ is the mean value, and N is the total number of pixels.

In this section, 5000 pairs of pixel values are randomly selected in each of the horizontal, vertical, and diagonal directions of the plain and cipher images for experimentation. Fig. 9 shows the distribution of adjacent pixels of the plain image peppers and its cipher image. The adjacent pixels of the plain image shows a linear distribution with a high correlation. However, the adjacent pixels of the cipher image are randomly and uniformly distributed, indicating almost no correlation. Table. 12 lists the adjacent correlation coefficients of the four images Peppers (512 × 512), Peppers2 (512 × 512), Tiffany (512 × 512), Airplane (512 × 512), and their encrypted images. The correlation coefficients of the red, green, and blue components of these plain images are all close to 1. After image encryption, the correlation coefficients are almost close to 0, meaning that the correlation between adjacent pixels is essentially eliminated. Meanwhile, Table. 13 gives the comparison results of the adjacent correlation coefficients of the encrypted peppers image. It shows that our scheme is slightly better than those references [37], [51], [52], [57], which means that our encryption algorithm has excellent security.

E. INFORMATION ENTROPY ANALYSIS

Information entropy is used to measure the randomness of information, and it can be calculated by *Eq*. (8). For a cipher image with uniformly distributed pixel values, the information entropy is close to ideal value 8, implying resistance to statistic attacks. As shown in Table. 14, the information entropy values of the encrypted images are close to 8. In Table. 15, for the cipher image Peppers512, the information entropy of our algorithm is slightly higher than those of the references [5], [58], [59], [60]. This indicates that these encrypted images have high randomness, and our proposed algorithm has better robustness against entropy attacks.

F. NOISE AND OCCLUSION ATTACK ANALYSIS

When an encrypted image is transmitted over a channel, it will inevitably be subject to noise interference and occlusion attacks. Therefore, an effective encryption algorithm should be robust against them. The color image Peppers in this section is used as a test image. The decrypted images after adding Salt & Pepper noise(SPN), Speckle noise(SN), and Gaussian noise(GN) to the cipher images of Peppers are shown in Fig. 10. Fig. 11 shows the cipher images with different sizes of occlusion and their corresponding decrypted images. As seen in Fig. 10 and Fig. 11, We can still recognize the information of the plain image from these decrypted images. Therefore, our proposed algorithm is highly resistant to noise or occlusion attacks. The receiver can still effectively decrypt the image to obtain the information of the plain

TABLE 14. Entropy of plain image and cipher image.

Image	Entropy					
	Red	Green	Blue	Red	Green	Blue
	Plain image			Cipher image		
Peppers	7.3409	6.9593	5.6373	7.9969	7.9973	7.9970
Airplane	6.7113	6.7853	6.2128	7.9971	7.9971	7.9973
Tiffany	4.3372	6.6643	6.4288	7.9973	7.9971	7.9969
Sailboat	7.2995	7.5464	7.1074	7.9976	7.9973	7.9969
Peppers2	7.3255	7.3912	6.9169	7.9972	7.9971	7.9970
Baboon	7.6836	7.4466	7.6874	7.9972	7.9973	7.9970

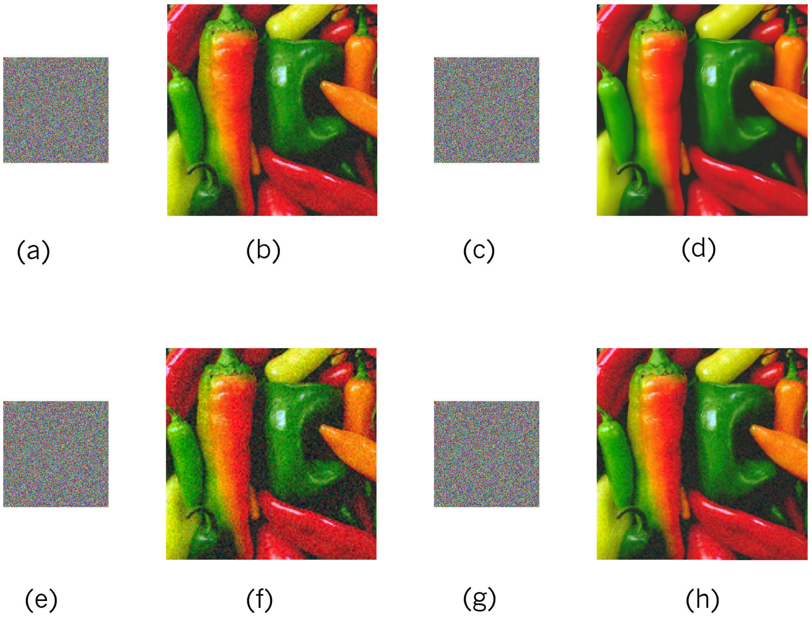


FIGURE 10. Decrypted image for noise attack (a) Cipher image by SPN with density = 0.001. (b) Decrypted image of (a). (c) Cipher image by SN with density = 0.000001. (d) Decrypted image of (c). (e) Cipher image by SN with density = 0.000005. (f) Decrypted image of (e). (g) Cipher image by GN with density = 0.000001. (h) Decrypted image of (g).

TABLE 15. Comparison with other studies on entropy.

Algorithms	Ref [5]	Ref [52]	Ref [58]	Ref [59]	Ours
R Component	7.9975	7.9985	7.9974	7.9972	7.9974
G Component	7.9975	7.9987	7.9972	7.9973	7.9971
B Component	7.9969	7.9986	7.9971	7.9975	7.9970

image when it receives a cipher image that suffers from noise contamination or occlusion attack.

G. DIFFERENTIAL ATTACK ANALYSIS

Differential attack testing is an important method for evaluating the sensitivity of an image encryption algorithm to small input changes to the plain image. Commonly, two almost identical images are selected, one of which is slightly modified, and then encrypted separately. By comparing the differences in the encrypted images, it is possible to determine whether the algorithm is resistant to differential attacks. The core of the difference attack is to verify whether the encryption algorithm can maintain the randomness of the ciphertext when the input is slightly changed, and to prevent the attacker from obtaining useful tips from the differences. To this end, two key metrics are often used: the *NPCR* and the uniform average change rate (*UACI*). The *NPCR* measures the proportion of different pixels between cipher images, which ideally should be close to 100%, while the *UACI* evaluates the average strength of the pixel change values, which ideally is close to 33% for image encryption algorithms. By accurately calculating and analyzing these metrics, we are able to comprehensively evaluate the performance of encryption algorithms under differential attacks, providing a strong guarantee for their security and reliability. The formula for calculating *UACI* is as follows:

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{C_1(i,j) - C_2(i,j)}{255} \times 100\% \quad (32)$$

where M and N refer to the length and width of the image, and C_1 and C_2 denote two different images.

In this section, by subtracting the first pixel of the plain test images by 1 and encrypting them using the proposed

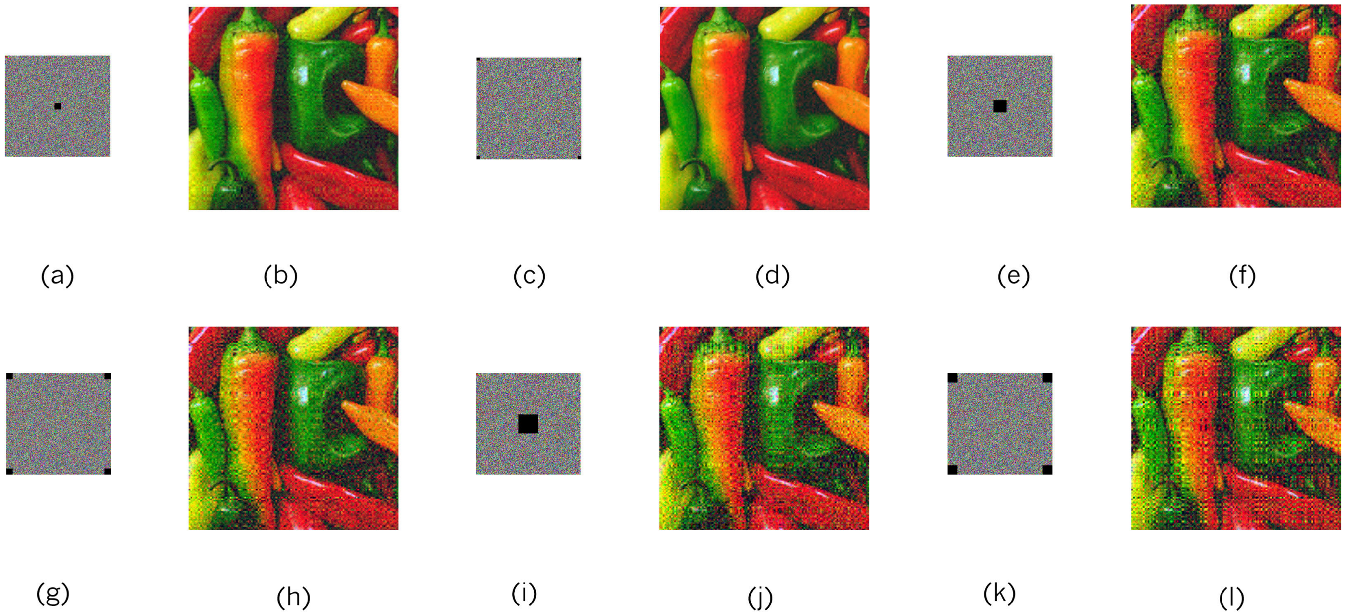


FIGURE 11. Decrypted image for occlusion attack (a) Cipher image by 16 × 16 occlusion at the center. (b) Decrypted image of (a). (c) Cipher image by 16 × 16 occlusion at the four corners. (d) Decrypted image of (c). (e) Cipher image by 32 × 32 occlusion at the center. (f) Decrypted image of (e). (g) Cipher image by 32 × 32 occlusion at the four corners. (h) Decrypted image of (g). (i) Cipher image by 48 × 48 occlusion at the center. (j) Decrypted image of (i). (k) Cipher image by 48 × 48 occlusion at the four corners. (l) Decrypted image of (k).

TABLE 16. UACI and NPCR of encryption results for plain images and micro-altered plain images.

Image	NPCR(%)	UACI(%)
Peppers	99.62	33.51
Sailboat	99.63	33.36
Tiffany	99.62	33.60
Peppers2	99.62	33.48
Baboon	99.61	33.52

encryption algorithm, the encryption results are compared with the encrypted image obtained by encrypting the original plain image. The results of the *NPCR* and *UACI* tests are presented in Table. 16. The *NPCR* and *UACI* obtained from the test results almost converge to the ideal values, indicating that the proposed algorithm is extremely sensitive to the plain image and thus can adequately resist differential attacks.

H. ANALYSIS OF KNOWN-PLAINTEXT ATTACK AND CHOSEN-PLAIN-TEXT ATTACK

The known-plaintext attacks and chosen-plaintext attacks are a means of analyzing cryptosystems to extract information from the original image. Several image encryption schemes have been successfully breached using known-plaintext attacks and chosen-plaintext attacks [60], [61], [62]. In order to resist known-plaintext attacks and chosen-plaintext attacks, our scheme employs some effective measures. The *SHA*-384 hash and information entropy of the plain image are computed and used to calculate the initial values of the composite chaotic systems and Josephus problem. Chaotic sequences generated by *LTS* are used for constructing the measurement matrices. Chaotic sequences produced by *LSS* are utilized to encode and decode the compressed *DNA*-level image. A chaotic sequence derived from *TSS* is employed to diffuse the encoded *DNA*-level image and the Josephus problem is exploited to confuse the measurements of the three components *R*, *G*, and *B* with each other. Therefore, different plain images correspond to their respective key streams and will be encrypted into completely distinct cipher images. It is nearly impossible for an attacker to break the image encryption system by analyzing a part of the encrypted image, so our encryption algorithm is highly robust to known-plaintext attacks and chosen-plaintext attacks.

VI. CONCLUSION

In this paper, we propose a new color image encryption algorithm based on compressive sensing and block *DNA* encoding. First, this paper adopts three composite chaotic systems in the encryption process, which have low time complexity and good dynamic characteristics. And the information entropy and *SHA*-384 hash value of the plain image are applied to calculate the initial values of these chaotic systems. Next, the plain image is measured and quantified. Notably, the scheme uses two-dimensional compressed sensing, which samples the image information more efficiently. The measurement matrices are optimized with *SVD*, significantly improving the quality of the reconstructed image. Then, the three compressed and quantified component matrices are scrambled and diffused through the permutation method based on the Josephus problem and the block-based *DNA* operation, and the three confused and diffused matrices are combined to obtain the encrypted image.

Since the encryption process depends on the plain image, this research scheme can effectively resist known-plaintext attacks and chosen-plaintext attacks. Color images from the USC-SIPI image database or commonly used images such as Peppers are used as test images in this scheme, the simulation experiments and performance analysis show that our proposed scheme has great advantages in terms of security performance and robustness. In the case of encrypting color images or grayscale images with equal length and width, this scheme is well worth considering, but the reconstruction speed and resistance to occlusion attacks of this scheme are still not ideal. At present, the transmission of digital images is increasingly required to guarantee timeliness and robustness, the design of an encryption algorithm that can fulfill the different sizes of color or grayscale images with high encryption and decryption efficiency and security needs to be continuously expanded and researched.

In the future, we will develop other encryption algorithms incorporating compressive sensing and study more efficient measurement matrices and $2D$ reconstruction methods. Moreover, new technologies such as deep learning or mathematical models will be used in image encryption algorithms to improve the security of image encryption and meet high efficiency.

REFERENCES

[1] 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.

[2] Z. Gan, X. Chai, D. Han, and Y. Chen, "A chaotic image encryption algorithm based on 3-D bit-plane permutation," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7111–7130, 2019.

[3] J. Deng, M. Zhou, C. Wang, S. Wang, and C. Xu, "Image segmentation encryption algorithm with chaotic sequence generation participated by cipher and multi-feedback loops," *Multimedia Tools Appl.*, vol. 80, no. 9, pp. 13821–13840, Apr. 2021.

[4] 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.

[5] X. Chai, J. Fu, Z. Gan, Y. Lu, and Y. Zhang, "An image encryption scheme based on multi-objective optimization and block compressed sensing," *Nonlinear Dynamics*, vol. 108, no. 3, pp. 2671–2704, 2022.

[6] L. Zhu, H. Song, X. Zhang, M. Yan, T. Zhang, X. Wang, and J. Xu, "A robust meaningful image encryption scheme based on block compressive sensing and SVD embedding," *Signal Process.*, vol. 175, Oct. 2020, Art. no. 107629.

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

[8] K. C. Jithin and S. Sankar, "Colour image encryption algorithm combining Arnold map, DNA sequence operation, and a Mandelbrot set," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102428.

[9] S. Zhou, P. He, and N. Kasabov, "A dynamic DNA color image encryption method based on SHA-512," *Entropy*, vol. 22, no. 10, p. 1091, 2020.

[10] W. Bao and C. Zhu, "A secure and robust image encryption algorithm based on compressive sensing and DNA coding," *Multimedia Tools Appl.*, vol. 81, no. 11, pp. 15977–15996, 2022.

[11] N. Zhou, Y. Hu, L. Gong, and G. Li, "Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations," *Quantum Inf. Process.*, vol. 16, pp. 1–23, Jun. 2017.

[12] A. A. Abd El-Latif, B. Abd-El-Atty, and M. Talha, "Robust encryption of quantum medical images," *IEEE Access*, vol. 6, pp. 1073–1081, 2018.

[13] A. L. A. Dalhoum, B. A. Mahafzah, A. A. Awwad, I. Aldhamari, A. Ortega, and M. Alfonseca, "Digital image scrambling using 2D cellular automata," *IEEE MultimediaMag.*, vol. 19, no. 4, pp. 28–36, Oct. 2012.

[14] Z. Gan, X. Chai, J. Zhang, Y. Zhang, and Y. Chen, "An effective image compression–encryption scheme based on compressive sensing (CS) and game of life (GOL)," *Neural Comput. Appl.*, vol. 32, no. 17, pp. 14113–14141, Sep. 2020.

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

[16] M. Ahmad and M. S. Alam, "A new algorithm of encryption and decryption of images using chaotic mapping," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 1, pp. 46–50, 2009.

[17] A. Jain and N. Rajpal, "A robust image encryption algorithm resistant to attacks using DNA and chaotic logistic maps," *Multimedia Tools Appl.*, vol. 75, pp. 5455–5472, May 2016.

[18] C. Li, T. Xie, Q. Liu, and G. Cheng, "Cryptanalyzing image encryption using chaotic logistic map," *Nonlinear Dynamics*, vol. 78, pp. 1545–1551, Oct. 2014.

[19] 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.

[20] C. Pak and L. Huang, "A new color image encryption using combination of the 1D chaotic map," *Signal Process.*, vol. 138, pp. 129–137, Sep. 2017.

[21] B. Ahuja and R. Doriya, "A secure algorithm using high-dimensional sine map for color image encryption," *Int. J. Inf. Technol.*, vol. 15, no. 3, pp. 1535–1543, 2023.

[22] M. Es-Sabry et al., "Securing images using high dimensional chaotic maps and DNA encoding techniques," *IEEE Access*, vol. 11, pp. 100856–100878, 2023, doi: 10.1109/ACCESS.2023.3315658.

[23] D. Zou, T. Pei, G. Xi, and L. Wang, "Image encryption based on hyper-chaotic system and improved zigzag diffusion method," *IEEE Access*, vol. 11, pp. 95396–95409, 2023, doi: 10.1109/ACCESS.2023.3311038.

[24] Y.-M. Li, Y. Deng, M. Jiang, and D. Wei, "Fast encryption algorithm based on chaotic system and cyclic shift in integer wavelet domain," *Fractal Fractional*, vol. 8, no. 2, p. 75, Jan. 2024.

[25] S. Zhu, X. Deng, W. Zhang, and C. Zhu, "Secure image encryption scheme based on a new robust chaotic map and strong S-box," *Math. Comput. Simul.*, vol. 207, pp. 322–346, May 2023.

[26] L. M. Zhang, K. H. Sun, W. H. Liu, and S. B. He, "A novel color image encryption scheme using fractional-order hyperchaotic system and DNA sequence operations," *Chin. Phys. B*, vol. 26, no. 10, 2017, Art. no. 100504.

[27] X. Wang et al., "Color image encryption algorithm based on Fisher-Yates scrambling and DNA subsequence operation," *Vis. Comput.*, vol. 39, pp. 43–58, 2023, doi: 10.1007/s00371-021-02311-2.

[28] S. Y. D. Nezhad, N. Safdarian, and S. A. H. Zadeh, "New method for fingerprint images encryption using DNA sequence and chaotic tent map," *Optik*, vol. 224, Dec. 2020, Art. no. 165661.

[29] J. Chen, Z. Zhu, L. Zhang, Y. Zhang, and B. Yang, "Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption," *Signal Process.*, vol. 142, pp. 340–353, Jan. 2018.

[30] X. Chai, X. Fu, Z. Gan, Y. Lu, and Y. Chen, "A color image cryptosystem based on dynamic DNA encryption and chaos," *Signal Process.*, vol. 155, pp. 44–62, Feb. 2019.

[31] C. Zhu, Z. Gan, Y. Lu, and X. Chai, "An image encryption algorithm based on 3-D DNA level permutation and substitution scheme," *Multimedia Tools Appl.*, vol. 79, pp. 7227–7258, Mar. 2020.

[32] Q. Liang and C. Zhu, "A new one-dimensional chaotic map for image encryption scheme based on random DNA coding," *Opt. Laser Technol.*, vol. 160, May 2023, Art. no. 109033.

[33] S. Zhu, X. Deng, W. Zhang, and C. Zhu, "Image encryption scheme based on newly designed chaotic map and parallel DNA coding," *Mathematics*, vol. 11, no. 1, p. 231, 2023.

[34] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006, doi: 10.1109/TIT.2006.871582.

[35] P. Lu, Z. Xu, X. Lu, and X. Liu, "Digital image information encryption based on compressive sensing and double random-phase encoding technique," *Optik*, vol. 124, no. 16, pp. 2514–2518, 2013.

[36] R. Huang and K. Sakurai, "A robust and compression-combined digital image encryption method based on compressive sensing," in *Proc. 7th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Oct. 2011, pp. 105–108.

[37] D. Wei and M. Jiang, "A fast image encryption algorithm based on parallel compressive sensing and DNA sequence," *Optik*, vol. 238, Jul. 2021, Art. no. 166748.

[38] J. Cai, S. Xie, and J. Zhang, "Image compression-encryption algorithm based on chaos and compressive sensing," *Multimedia Tools Appl.*, vol. 82, no. 14, pp. 22189–22212, 2023.

[39] D. Huo, Y. Qiu, C. Han, L. Wei, Y. Hong, Z. Zhu, and X. Zhou, "A visually meaningful double-image encryption scheme using 2D compressive sensing and multi-rule DNA encoding," *Complex Intell. Syst.*, vol. 9, no. 5, pp. 4783–4803, 2023.

[40] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Process.*, vol. 97, pp. 172–182, Apr. 2014.

[41] B. Zhang, D. Xiao, and Y. Xiang, "Robust coding of encrypted images via 2D compressed sensing," *IEEE Trans. Multimedia*, vol. 23, pp. 2656–2671, 2021.

[42] W. Huang, D. Jiang, Y. An, L. Liu, and X. Wang, "A novel double-image encryption algorithm based on Rossler hyperchaotic system and compressive sensing," *IEEE Access*, vol. 9, pp. 41704–41716, 2021.

[43] W. Xiao-Qing, Z. Hao, S. Yu-Jie, and W. Xing-Yuan, "A plaintext-related image encryption algorithm based on compressive sensing and a novel hyperchaotic system," *Int. J. Bifurcation Chaos*, vol. 31, no. 2, 2021, Art. no. 2150021.

[44] Q. Xu, K. Sun, C. Cao, and C. Zhu, "A fast image encryption algorithm based on compressive sensing and hyperchaotic map," *Opt. Lasers Eng.*, vol. 121, pp. 203–214, Oct. 2019.

[45] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993, doi: 10.1109/78.258082.

[46] S. K. Sahoo and A. Makur, "Signal recovery from random measurements via extended orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2572–2581, May 2015.

[47] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ^0 norm," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 289–301, Jan. 2009, doi: 10.1109/TSP.2008.2007606.

[48] G. Chen, D. Li, and J. Zhang, "Iterative gradient projection algorithm for two-dimensional compressive sensing sparse image reconstruction," *Signal Process.*, vol. 104, pp. 15–26, Nov. 2014.

[49] X. Chai, H. Wu, Z. Gan, Y. Zhang, and Y. Chen, "Hiding cipher-images generated by 2-D compressive sensing with a multi-embedding strategy," *Signal Process.*, vol. 171, Jun. 2020, Art. no. 107525.

[50] X. L. Tian and Z. H. Xi, "An optimization algorithm for measurement matrix in compressed sensing," *Electron. Sci. Technol.*, vol. 8, no. 28, pp. 102–111, 2015.

[51] X. Chai, J. Bi, Z. Gan, X. Liu, Y. Zhang, and Y. Chen, "Color image compression and encryption scheme based on compressive sensing and double random encryption strategy," *Signal Process.*, vol. 176, Nov. 2020, Art. no. 107684.

[52] J. Wei, M. Zhang, and X. Tong, "Multi-image compression-encryption algorithm based on compressed sensing and optical encryption," *Entropy*, vol. 24, no. 6, p. 784, 2022.

[53] X. Jiang, Y. Xiao, Y. Xie, B. Liu, Y. Ye, T. Song, J. Chai, and Y. Liu, "Exploiting optical chaos for double images encryption with compressive sensing and double random phase encoding," *Optics Commun.*, vol. 484, Apr. 2021, Art. no. 126683.

[54] S. Nan, X. Feng, Y. Wu, and H. Zhang, "Remote sensing image compression and encryption based on block compressive sensing and 2D-LCCCM," *Nonlinear Dyn.*, vol. 108, no. 3, pp. 2705–2729, 2022.

[55] A. Roy, A. P. Misra, and S. Banerjee, "Chaos-based image encryption using vertical-cavity surface-emitting lasers," *Optik*, vol. 176, pp. 119–131, Jan. 2019.

[56] H. Liu, Y. Xu, and C. Ma, "Chaos-based image hybrid encryption algorithm using key stretching and hash feedback," *Optik*, vol. 216, Aug. 2020, Art. no. 164925.

[57] Z. Gan, X. Chai, J. Bi, and X. Chen, "Content-adaptive image compression and encryption via optimized compressive sensing with double random phase encoding driven by chaos," *Complex Intell. Syst.*, vol. 8, no. 3, pp. 2291–2309, Jun. 2022.

[58] H. Dong, E. Bai, X.-Q. Jiang, and Y. Wu, "Color image compression-encryption using fractional-order hyperchaotic system and DNA coding," *IEEE Access*, vol. 8, pp. 163524–163540, 2020.

[59] J. Yu, W. Xie, Z. Zhong, and H. Wang, "Image encryption algorithm based on hyperchaotic system and a new DNA sequence operation," *Chaos, Solitons Fractals*, vol. 162, Sep. 2022, Art. no. 112456.

[60] C. Li, D. Lin, J. Lü, and F. Hao, "Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography," *IEEE Multimedia Mag.*, vol. 25, no. 4, pp. 46–56, Oct. 2018.

[61] C. Li, D. Lin, B. Feng, J. Lü, and F. Hao, "Cryptanalysis of a chaotic image encryption algorithm based on information entropy," *IEEE Access*, vol. 6, pp. 75834–75842, 2018.

[62] C. Li, Y. Zhang, and E. Y. Xie, "When an attacker meets a cipher-image in 2018: A year in review," *J. Inf. Secur. Appl.*, vol. 48, Oct. 2018, Art. no. 102361.

QIJI HE was born in Hunan, China, in 1999. He received the B.S. degree from Jiangxi Normal University of Science and Technology, China, in 2017. He is currently pursuing the master's degree in cyberspace security with Jinan University, China. His main research interests include image encryption algorithms and compressive sensing.

PEIYA LI received the B.S. degree from Anhui Normal University, China, in 2012, the M.E. degree from Zhejiang University, China, in 2014, and the Ph.D. degree in engineering from The Hong Kong Polytechnic University, Hong Kong, in 2018. She is currently a Lecturer with the College of Cyber Security, Jinan University, Guangzhou, China. Her research interests include multimedia encryption, image coding, and image retrieval.

YANYIXIAO WANG received the B.S. degree from Jiangxi University of Science and Technology, China, in 2021. She is currently pursuing the M.E. degree with Jinan University, Guangzhou, China. Her research interests include multimedia security and information hiding.