



# Image encryption algorithm using multi-base diffusion and a new four-dimensional chaotic system

Simiao Wang<sup>1</sup> · Baichao Sun<sup>1</sup> · Yiming Wang<sup>1</sup> · Baoxiang Du<sup>1</sup>

Received: 11 July 2022 / Revised: 1 March 2023 / Accepted: 11 June 2023 /  
Published online: 20 June 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Information security is very important in the era of rapid development of science and technology. People often use multimedia to communicate in their daily life. Image plays an important role in multimedia communication, so it is urgent to protect image information. In order to improve the security of images in transmission, an image encryption algorithm using multi-base diffusion and a new four-dimensional chaotic system is designed in this paper. The algorithm decomposes each pixel value into multi-based. After decomposition, the coefficient matrix and base matrix are scrambled by FYTS using the sequence generated by the new four-dimensional chaotic system, and finally recombined to perform pixel-level and bit-level scrambling respectively. After simulation with MATLAB, it is obvious that the final encrypted image does not have the contour of the original image, which meets the requirements of encryption. Through histogram, entropy analysis, anti-differential attack analysis and other experimental results, it is proved that the proposed algorithm has high security.

**Keywords** Bit plane · Diffusion · 4D chaotic system · Image encryption

## 1 Introduction

In the era of increasingly developed network media tools, people have become accustomed to remote communication in daily life [17, 32]. In multimedia communication, images, as a common transmission carrier, are widely used in various fields, such as the military field, the medical field, and so on. In many cases, images may be stolen and destroyed by people with ulterior motives, resulting in information leakage and losses. Nowadays, text encryption technology is very mature, but if you directly use text encryption to encrypt images, the encryption effect is not ideal. Because the amount of image data is larger than that of text, and its ranks have a certain correlation, image encryption has also become a research hotspot. Apply DES, AES, RSA and other encryption methods to image encryption [12, 13, 25, 38]. However, due to the large amount of image data, the efficiency is not very high.

---

✉ Baoxiang Du  
dubaoxiang@hlju.edu.cn

<sup>1</sup> Electronic Engineering College, Heilongjiang University, Harbin, China

Since Matthews proposed the encryption method based on chaos in 1989 [18], it was found that chaotic encryption is highly sensitive to the initial conditions, and various methods of applying chaos theory to encryption have been proposed.

Initially, scholars used classical chaotic maps to encrypt images. In 2006, Pareek used an 80-bit external key and two logistic maps to encrypt images [21]. [20] proposed an image encryption algorithm based on Arnold transform and hyperchaotic map, decomposed the original image at the bit level, used Arnold transform to scramble the image, and then diffused it by hyperchaotic map to obtain an encrypted image. In 2015, Zhou et al. combined Skew tent map and Line map for L-scrambling and diffusion, and designed an image encryption scheme suitable for all sizes [41]. In 2021, Sang et al. scrambled the plaintext image with Logistic map, and then encoded it with a deep autoencoder [23]. [14] uses Logistic map to generate key matrices for permutation and diffusion, which are resistant to differential attacks.

At present, many scholars have proposed new chaotic systems. Compared with the classical chaotic mapping, the improved chaotic systems have been improved in many aspects. At the same time, these chaotic systems have also been applied to image encryption [5]. [43] proposed an image encryption method based on memristive chaotic system and compressed sensing, which can well resist chosen-plaintext attack. [31] designed a compound one-dimensional two-parameter chaotic system and applied it to scramble in row, column and diagonal directions. [5] proposed a new one-dimensional chaotic map with a larger chaotic range, and combined it with dynamic DNA coding designed a selective image encryption method. To encrypt the image, all of the approaches previously discussed rely on low-dimensional chaotic mapping. Although low-dimensional mapping operates quickly, the key space is limited and has a history of being predictably predictable [6]. In [11], 2D sin-cos chaotic mapping is used in the scrambling part, 2D chaotic mapping is used, and 1D Logical-tent map is used in the diffusion part to generate a chaotic matrix for XOR. In [1], the sequence generated by two-dimensional economic mapping is converted into binary, and XOR is performed with the image after the rows and columns are scrambled. [16] used the chaotic map of the 2-D Baker Map and Logistic map in series to form ciphertext images after scrambling and diffusion.

However, all the methods mentioned above are based on low-dimensional chaotic mapping to encrypt the image. Although low-dimensional mapping has the advantages of fast operation, the key space is small and has been proved to be easy to predict [26]. The high-dimensional chaotic system can effectively enlarge the key space in image encryption. In [28], a 4D mixed mapping is proposed by combining one-dimensional Sine map and 2D Thinker Bell map, and the scrambled matrix is formed to scramble the image. [8] designs a new four-dimensional chaotic system with hidden attractors and used it to generate random numbers for image encryption. [2] extends Arnold mapping to 3D space and proposes 3D modular chaotic map, which improves the speed and key space of color image encryption. A new chaotic system NCCS was proposed in [42] by combining Sine map and Tent map, and the initial value of the system was generated by SHA384 to carry out image scrambling diffusion.

For the scrambling process, there are generally two scrambling methods: pixel-level scrambling and bit-level scrambling. Scrambling on the bit plane can change both the pixel position and the pixel value. [15] used chaotic transformation and row-column scrambling at the pixel level in the scrambling process, which has strong security. [24] uses the cyclic shift method when it is scrambled in the bit plane, and the key is generated by the Henon map. [39] proposed to use a bit-level pixel scrambling strategy to replace the bit planes of the image with each other, which does not require additional

storage space. [30] used L-shaped and Fisher-Yates methods for scrambling, and then used L-shaped to perform bit-level scrambling on pixel values.

Regarding the concept of decomposition, many scholars decompose the pixel value of the image in different ways. For example [36] applies DWT and DCT in Cb and Cr space in YCbCr space, and performs singular value decomposition, the security is improved. [22] used the QR decomposition method to obtain the permutation matrix, generated the ciphertext, and expanded the key space. [4] proposed to decompose the image through singular value first, and then scrambling it multiple times through Arnold transform. [27] encrypts multiple images at the same time, first decomposes the original image into bit planes, and then performs XOR operations with the chaotic matrix generated by the chaotic system. [35] used the direct difference between the image and the composite vector as a key, and the decomposed amplitude and phase were obtained using an optical modulator. [37] proposed to use the M-ary decomposition method to decompose the pixel values of the image in [161, 256], and defined virtual and real bits to improve the visual quality.

By analyzing the above image encryption algorithms, this paper uses a high-dimensional chaotic system to generate chaotic sequences. The pixels are scrambled on the bit plane and diffused by decomposing the pixel values. The following work has been done: 1. A four-dimensional chaotic system has been designed and tested by NIST. 2. A scrambling algorithm called FYTS, which combines Fisher Yates scrambling and Thorp scrambling, is proposed. 3. In the diffusion part, a diffusion method is proposed to decompose and reassemble pixel values. The security of the proposed algorithm can be proved by histogram analysis and differential attack resistance.

The rest of this paper is as follows: Section 2 introduces the new four-dimensional chaotic system, FYTS scrambling method and multi-base diffusion method. Section 3 introduces the image encryption and decryption algorithm. Section 4 carries on the safety analysis, evaluates its safety and reliability. Section 5 is the conclusion.

## 2 Preliminaries

### 2.1 Chaotic system

#### 2.1.1 Proposed four-dimensional chaotic system

In this paper, a novel four-dimensional chaotic system is proposed, which is defined as Eq. (1)

$$\begin{cases} \dot{x} = y - x \\ \dot{y} = -axz - (0.1 + 0.03w^2)x \\ \dot{z} = xy - b \\ \dot{w} = -cx \end{cases} \quad (1)$$

Where  $x, y, z, w$  are all state variables, and  $a, b, c$  are system parameters.

According to Eq. (2), when  $b=0$ , the system has line equilibrium point  $(0, 0, \eta, l)$ , where  $\eta, l$  is an arbitrary constant. When  $b$  is not equal to 0, there is no equilibrium point in the system, and the attractor is a hidden attractor.

**Fig. 1** Phase diagram of four-dimensional chaotic system. **a** Phase diagram of the x-y plane. **b** Phase diagram of the x-z plane. **c** Phase diagram of the x-w plane. **d** Phase diagram of the y-z plane. **e** Phase diagram of the y-w plane. **f** Phase diagram of the x-y-z plane. **g** Phase diagram of the x-y-w plane. **h** Phase diagram of the y-z-w plane

$$\begin{cases} y - x = 0 \\ -axz - (0.1 + 0.03w^2)x = 0 \\ xy - b = 0 \\ -cx = 0 \end{cases} \tag{2}$$

The Jacobian matrix of the system can be solved in Eq. (3), and then the divergence of the system can be obtained as shown in Eq. (4). It can be seen that the divergence is equal to  $-1$  and less than  $0$ . It shows that the system is a dissipative chaotic system.

$$J = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -az - 0.03w^2 - 0.1 & 0 & -ax & -0.06wx \\ y & x & 0 & 0 \\ -c & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

$$\nabla V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{w}}{\partial w} = -1 \tag{4}$$

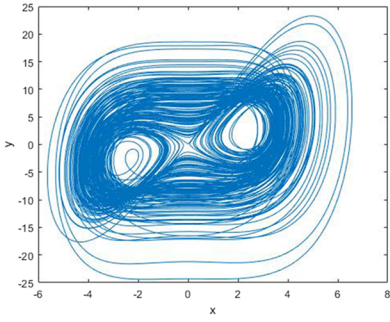
In addition, we can judge the stability of the equilibrium point by calculating the characteristic roots of the system, where  $a=2.5$ ,  $b=0$ , and  $c=5$ . Set  $\eta=0$ ,  $l=0$ . In this case, the equilibrium point of the system is  $s_0=(0,0,0,0)$ . We can get the characteristic equation:  $P_0(\lambda)=\lambda^4+\lambda^3+0.1\lambda^2$ , after solving the equation, we get the characteristic roots:  $\lambda_1=0$ ;  $\lambda_2=-0.8873$ ;  $\lambda_3=-0.1127$ . Therefore, the equilibrium point type is unstable saddle point.

When  $a=2.5$ ,  $b=6$ , and  $c=5$ , the phase diagrams of each plane are shown in Fig. 1. The chaotic behavior of the system can be intuitively understood from the phase diagram. When the value of  $c$  varies between  $[0, 30]$ , the Lyapunov exponent of this system varies as shown in Fig. 2. We know that when at least one Lyapunov exponent is greater than zero, the system is chaotic. When two or more are greater than  $0$ , it is hyperchaotic state. It can be seen from the figure that the system is in a hyperchaotic state when  $c$  is in  $[3.4, 8]$ . The bifurcation diagram of this chaotic system is shown in Fig. 3.

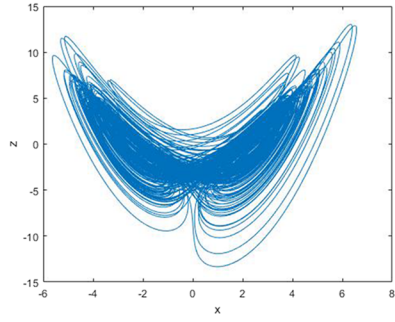
Kaplan-Yorke dimension is an important geometric characteristic quantity to describe dynamic systems. It is usually used to measure the complexity of chaotic systems. The closer it is to the fractal dimension of the system, the more fractal the attractor of the system will be. Its mathematical calculation expression is shown in Eq. (5):

$$D_{KY} = j + \frac{1}{LE_j + 1} \sum_{i=1}^j LE_i \tag{5}$$

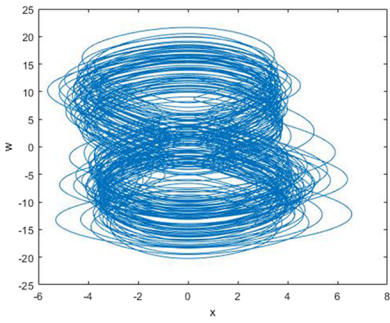
Where  $j$  is the largest integer that makes  $x > 0$ . When  $a=2.5$ ,  $b=6$ ,  $c=5$ ,  $LE1=0.1727$ ,  $LE2=0.0328$ ,  $LE3=-0.0016$ ,  $LE4=-1.2038$ . By substituting the values into Eq. (5), the Kaplan-Yorke dimension of the proposed system is  $3.2042$ .



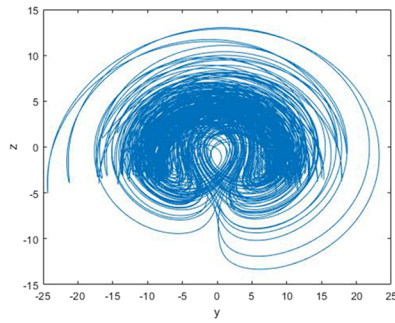
(a)



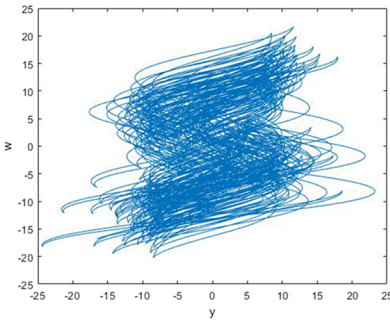
(b)



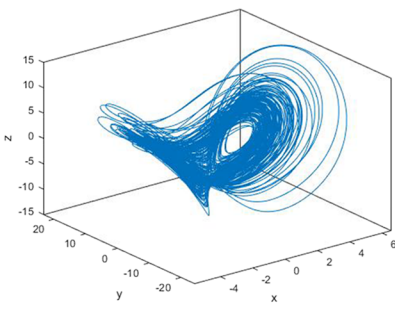
(c)



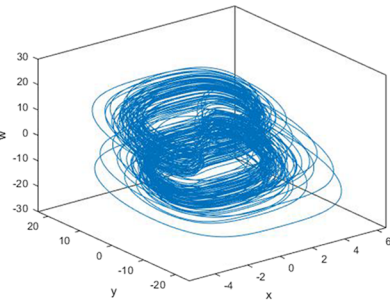
(d)



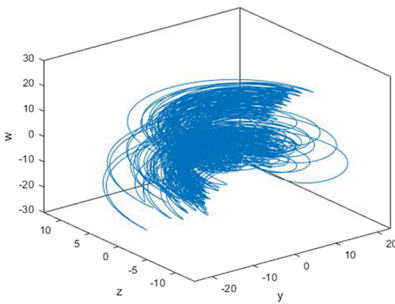
(e)



(f)

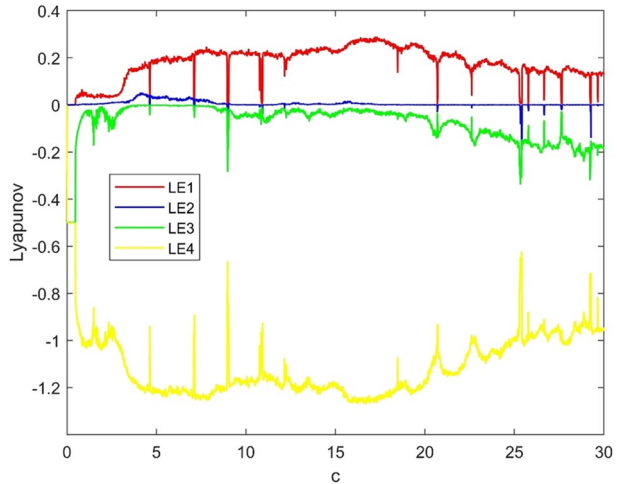


(g)

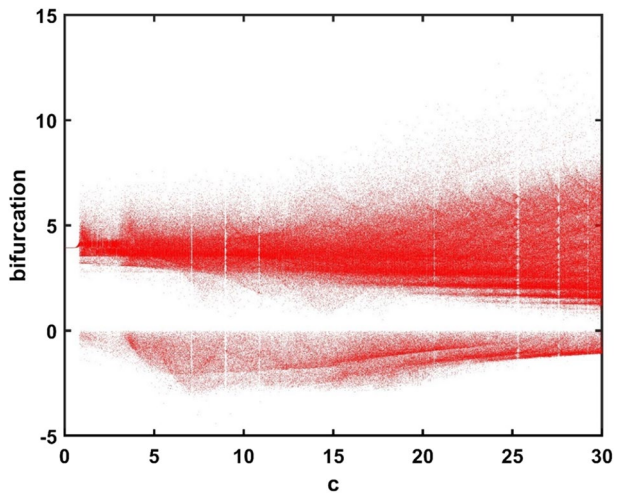


(h)

**Fig. 2** Lyapunov exponent of the proposed system



**Fig. 3** Bifurcation diagram of the proposed system



After applying the NIST test to this system, the data in Table 1 was obtained. It can be found that all indicators meet the standard requirements. Through this test, we can also judge that the chaotic sequence generated by the four-dimensional chaotic system has high randomness.

### 2.1.2 Sine-logistic map

To improve the chaotic complexity of 1D map, [9] introduced a sine chaotification model. The resulting chaotic map has better chaotic complexity and larger chaotic range. Compared with the classical Logistic map, the obtained Sine-Logistic map overcomes the shortcoming of its small initial value range. Its formula is shown in Eq. (6).

$$d_{k+1} = \sin(\pi \mu d_k (1 - d_k)), d_k \in (0, 1) \tag{6}$$

**Table 1** NIST test of proposed four-dimensional chaotic system

Test	P value(X)	P value(Y)	P value(Z)	P value(W)	State
Frequency	0.6811	0.0674	0.3914	0.8942	PASS
BlockFrequency	0.7635	0.9331	0.0652	0.2196	PASS
CumulativeSums	0.6062	0.8281	1.0000	0.5936	PASS
Runs	0.0685	0.2174	0.0448	0.3143	PASS
LongestRun	0.5480	0.4422	0.1755	0.9436	PASS
NonOverlappingTemplate	0.1468	0.2955	0.6672	0.8906	PASS
Serial	0.2634	0.2634	0.2634	0.2634	PASS
LinearComplexity	0.4279	0.4279	0.4279	0.4279	PASS
RandomExcursions	0.6198	0.6198	0.6198	0.6198	PASS
RandomExcursionsVariant	0.3241	0.3241	0.3241	0.3241	PASS
ApproximateEntropy	0.9630	0.2032	0.8019	0.9184	PASS
Universal	0.5324	0.6554	0.8875	0.6546	PASS
FFT	0.0519	0.4506	0.6014	0.5423	PASS
Rank	0.3936	0.0774	0.0774	0.0192	PASS
OverlappingTemplate	0.9925	0.9925	0.9925	0.9925	PASS

## 2.2 FYTS scrambling

### 2.2.1 Fisher-yates shuffle

The Fisher-Yates shuffle algorithm was first proposed by Ronald Fisher and Frank Yates. It was later modernized by Knuth in *The Art of Computer Programming*. Thus this algorithm is also called Knuth random scrambling method. This method is suitable for computer calculations, which can randomly arrange a finite set. For a sequence  $a$  with  $n$  elements, when randomizing from the  $n$ -1th element to the first element, select a random integer  $j$  and swap elements  $a[j]$  and  $a[i]$ , where the range of  $j$  is  $[0, i]$ .

### 2.2.2 Thorp shuffle

In 1973, Edward Thorp proposed a model for shuffling cards: Suppose there are  $2N$  cards, and divide the cards into two equal piles of  $N$  cards each. Then randomly flip a coin, if the result is heads, take the  $N$ th card from the left cards, and if the result is tails, take the  $N$ th card from the right cards. Thorp shuffle is named after Edward Thorp and is mainly used for small-space encryption. [19] proposed an encryption scheme in small domains based on Thorp shuffle and block encryption and proved its security. The steps to shuffle the  $A$  sequence of length  $2N$  into the  $B$  sequence using thorp shuffle are as follows:

- Step 1: Divide  $A$  into two sequences of length  $N$  called  $A_1$  and  $A_2$  respectively.
- Step 2: Use a decision sequence  $C$  with all elements in the range  $[0, 1]$ , if  $C(i)$  is less than 0.5, then take elements from  $A_1$ , if  $C(i)$  is greater than 0.5, then take elements from  $A_2$ .
- Step 3: Put the elements taken out in a certain order into sequence  $B$ , and the scrambled sequence is obtained.



### 2.2.3 Enhanced fisher-yates thorp shuffle

This paper combines the above two scrambling methods and proposes the Enhanced Fisher-Yates Thorp shuffle (FYTS). Assuming that the size of image I is  $M \times N$ , the specific steps for scrambling this image by FYTS are as follows, and the flow chart is shown in Fig. 4.

Step 1: Generate a binary sequence A with a length of  $M \times N$ .

Step 2: Convert the  $M \times N$  image I into a one-dimensional sequence O with a length of  $M \times N$ .

Step 3: When the k-th element A(k) in A is 1, swap O(i) and O(j). where the value of j is as shown in Eq. (7):

$$j = \text{floor}(\text{mod}(A(i) \times 10^4, n) + 1) \tag{7}$$

Step 4: When the k-th element A(k) in A is 0, swap O(i), O(j) and O(w) respectively. Where the value of w is shown in Eq. (8):

$$w = \text{mod}(i + j + \text{floor}(n/2), n) + 1 \tag{8}$$

Step 5: Then convert the O after the element is replaced into the size of  $M \times N$ , which is the final image obtained after FYTS.

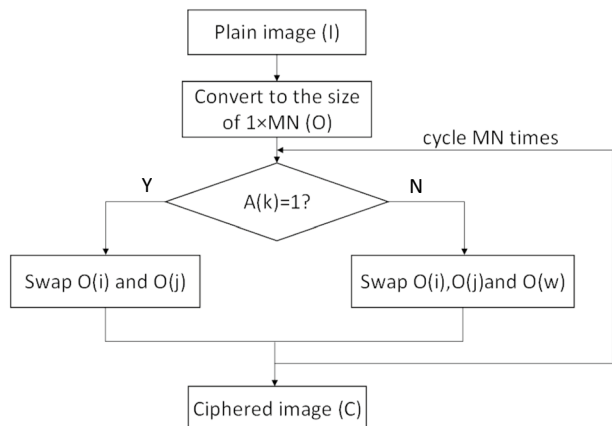
### 2.3 Multi-base diffusion method of pixels

In general, computers often convert numerical values to binary, octal, hexadecimal, etc. A value x can be converted to base B by Eq. (9):

$$x = A_n \cdot B^n + A_{n-1} \cdot B^{n-1} + \dots + A_1 \cdot B^1 + A_0 \cdot B^0 \tag{9}$$

[37] proposed an M-ary decomposition method to decompose positive integers in different bases. When converting a value to binary, the value is usually divided by 2. And when the multi-binary factorization method is used to factorize integers, the divisor changes. Taking B base as an example, to decompose an integer x in B base is to divide x by B continuously. Based on this M-ary decomposition method, this paper proposes a multi-base diffusion method for pixels. The method decomposes the pixel value of each pixel in the image,

Fig. 4 Flowchart of Enhanced Fisher-Yates Thorp shuffle





and generates a coefficient matrix and a base matrix after decomposing and then operates. Taking a 4×4 matrix as an example to decompose the pixels as shown in Fig. 5. Specific steps are as follows:

Step 1: Read the image I with the image size M\*N, and generate the chaotic sequence S with the length M\*N.

Step 2: Convert the chaotic sequence S to a number in the range [0,17] with Eq. (10), and add 4 to the value less than 4 in the converted sequence, so that the range of the sequence S is in [6, 25].

$$X = \text{floor}(\text{mod}(X^* 10000, 17)) \tag{10}$$

Step 3: Read the i-th pixel value in the image. Eq. (11)- Eq. (14) are used to calculate the four coefficients a, b, c, d after decomposing the pixel values.

$$a = \text{floor}(x/s(1)^3) \tag{11}$$

$$b = \text{floor}((x - a \cdot s(1)^3)/s(2)^2) \tag{12}$$

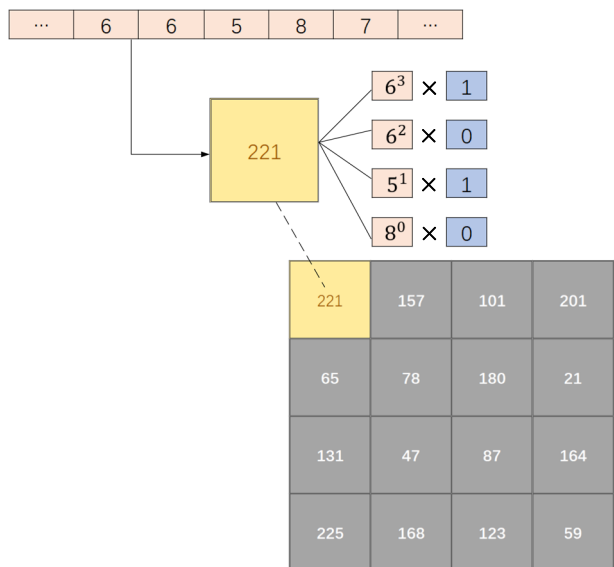
$$c = \text{floor}((x - a \cdot s(1)^3 - b \cdot s(2)^2)/s(3)) \tag{13}$$

$$d = \text{floor}((x - a \cdot s(1)^3 - b \cdot s(2)^2 - c \cdot s(3))) \tag{14}$$

Among them, x is the pixel value of the i-th pixel, and s is a matrix of length 3 obtained by intercepting the chaotic sequence S through Eq. (15).

$$s = S(i : i + 2) \tag{15}$$

Fig. 5 Multiple base diffusion method of pixels



- Step 4: Store the coefficient obtained by the decomposition of the  $i$ -th pixel in the matrix  $Y$ , and store the base number in the matrix  $Z$ .
- Step 5: Scramble the matrices  $X$  and  $Z$  with the FYTS scrambling method, and the scrambled matrices are  $X_1$  and  $Z_1$ .
- Step 6: Recombine the matrices  $X_1$  and  $Z_1$  according to Eq. (16), and the obtained matrix  $Q$  is the ciphertext image after MBD decomposition and scramble.

$$Q(i) = X_1(1, i) * Z_1(1, i)^3 + X_1(2, i) * Z_1(2, i)^2 + X_1(3, i) * Z_1(3, i) + X_1(4, i); \quad (16)$$

### 3 Proposed method

#### 3.1 Encryption algorithm

When the original image  $I$  of size  $M \times N$  is encrypted, the encryption steps are as follows. The encryption flow chart is shown in Fig. 6.

- Step 1: Read the original image  $I$ , the size is  $M \times N$ .
- Step 2: Select the initial value of the four-dimensional chaotic system according to the original image  $I$ , and the formula is shown in Eq. (17). The final initial value  $X_0, Y_0, Z_0, H_0$  is obtained after four decimal places are retained. Four chaotic sequences  $S_1, S_2, S_3, S_4$  are generated after setting four initial values.

$$\begin{aligned}
 X_0 &= \text{sum}(\text{sum}(\text{bitand}(B1, 136))) / (136 * \text{SUM}) \\
 Y_0 &= \text{sum}(\text{sum}(\text{bitand}(B1, 68))) / (68 * \text{SUM}) \\
 Z_0 &= \text{sum}(\text{sum}(\text{bitand}(B1, 34))) / (34 * \text{SUM}) \\
 H_0 &= \text{sum}(\text{sum}(\text{bitand}(B1, 17))) / (17 * \text{SUM})
 \end{aligned} \quad (17)$$

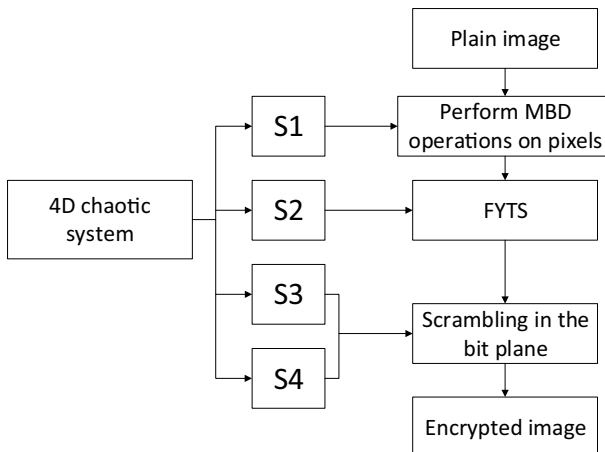


Fig. 6 Flowchart of the proposed encryption image algorithm

Step 3: Use sequence  $S_1$  to decompose plain image  $I$  with MBD method. The decomposed image is  $B$ .

Step 4: First convert the decomposed image  $B$  into a one-dimensional sequence, and then use the sequence  $S_2$  of length  $M \times N$  to perform FYTS scrambling on it. When scrambling,  $S_2$  is used to generate a binary sequence  $C$ . When the element in  $S_2$  is greater than the average value  $a$  of the whole sequence,  $C(i) = 1$ ; otherwise,  $C(i) = 0$ . After scrambling, it is transformed into a matrix  $P$  of  $M \times N$  size.

Step 5: All pixel values in  $P$  are converted to octet binary with a size of  $8 M \times N$ . Eq. (18) is used to intercept  $8 \times M$  elements of sequence  $S_3$ , and Eq. (19) is used to intercept  $N$  elements of sequence  $S_4$ . Scramble columns of  $P$  with  $y_1$  by sorting  $y_1$  upward and rearranging columns of  $P$  by index. In the same way, scramble rows of  $P$  with  $y_2$  to get the scrambled binary matrix. Finally, convert the binary matrix to decimal.

$$y_1 = S_3(3001 : 8m + 3000) \quad (18)$$

$$y_2 = S_4(3001 : n + 3000) \quad (19)$$

Step 6: FYTS scrambling was performed on the bit plane of the image obtained in step5 using the sequence with length  $M \times N$  generated by Sine-logistic map. Finally, the encrypted image  $C$  is obtained.

### 3.2 Decryption algorithm

The decryption process is the reverse of the encryption process, as follows:

Step 1: Read the encrypted image  $C$  with a size of  $M \times N$ .

Step 2: Select the same initial values and parameters as in encryption, and use the proposed 4D chaotic system to generate four-dimensional chaotic sequences.

Step 3: Convert the encrypted image  $C$  to binary, and perform FYTS inverse scrambling on the bit plane. The same binary sequence is used to determine the swapped element position.

Step 4: Reverse the row and column on the bit plane with  $y_1$  and  $y_2$ , where  $y_1$  and  $y_2$  are the same as in encryption. The resulting matrix in decimal is  $P$ .

Step 5: Generate a decision sequence  $C$  using sequence  $S_2$  with a length of  $M \times N$ . The sequence  $C$  is used to transform  $P$  into a one-dimensional sequence for FYTS inverse scrambling.

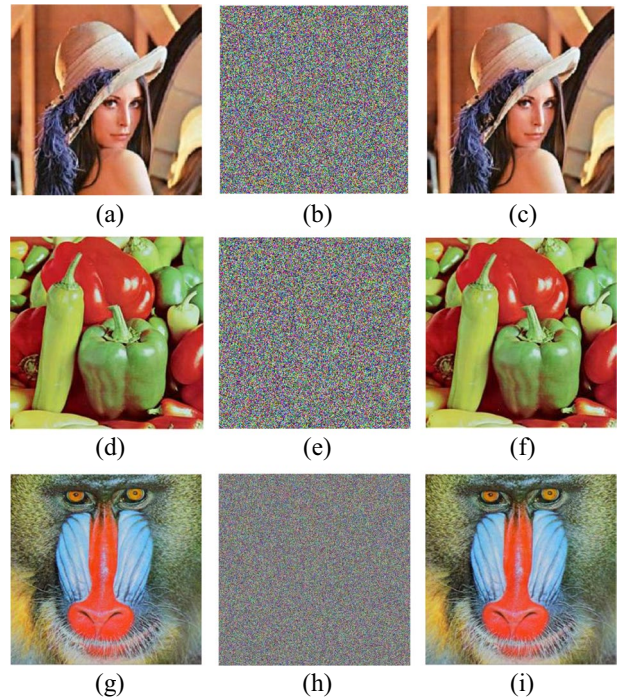
Step 6: Read the coefficient matrix of MBD decomposition. Use the sequence  $S_1$  to obtain the matrix of base numbers. The two matrices are combined after FYTS inverse scrambling. The reconstructed matrix is the decrypted image.

## 4 Security analysis

### 4.1 Encryption and decryption Results

We use MATLAB 2019B to simulate the algorithm. After the classical image Lena, Baboon and Pepper are encrypted with the proposed image encryption algorithm, the results are as shown in Fig. 7. The size of Baboon are  $512 \times 512$ , and the size of others are

**Fig. 7** Comparison of encryption and decryption results (a) Plaintext image of Lena (b) Ciphertext image of Lena (c) Decrypted image of Lena (d) Plaintext image of Baboon (e) Ciphertext image of Baboon (f) Decrypted image of Baboon (g) Plaintext image of Pepper (h) Ciphertext image of Pepper (i) Decrypted image of Pepper



$256 \times 256$ . It can be seen that the encrypted image is like noise, from which the information of the original image cannot be obtained.

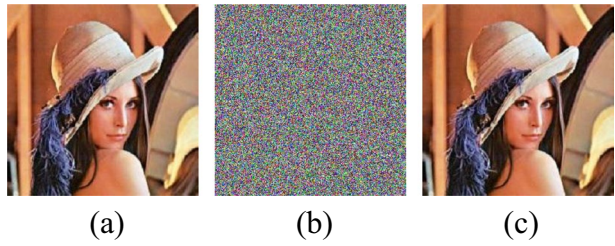
## 4.2 Key space

A strong encryption algorithm should have a large key space. For high security, the key space should be larger than  $2^{100}$ , which can resist brute force attacks. The chaotic system proposed in this paper has four initial values and three parameters, and the Sine-Logistic map has one initial value and one parameter. So when the computer precision is  $10^{-15}$ , the key space should be  $10^{135}$ . It can be seen that the key space is much larger than  $2^{100}$ .

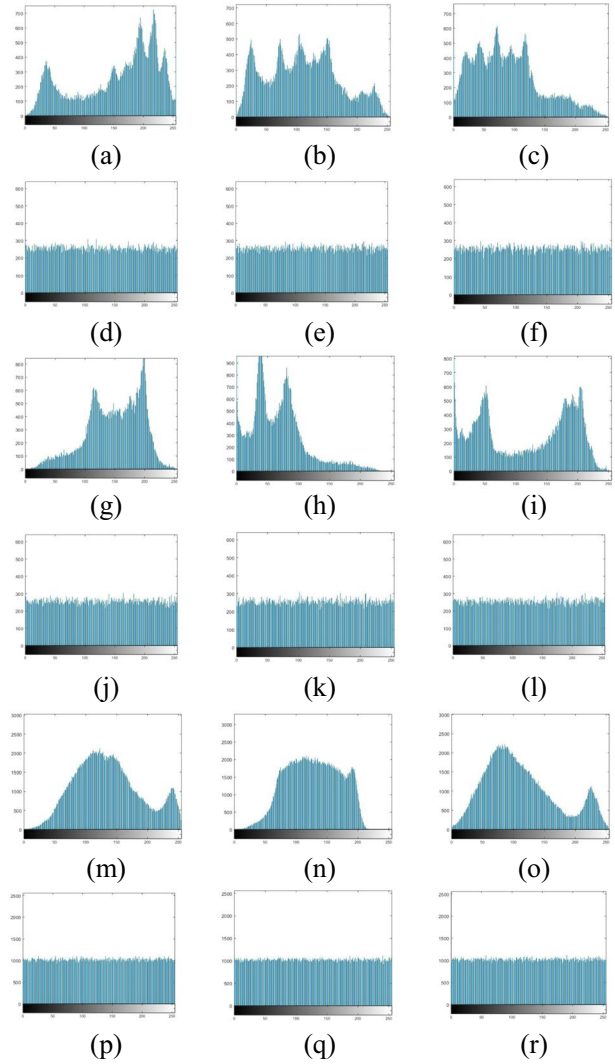
## 4.3 Key sensitivity

Key sensitivity means that the original image cannot be decrypted even if there is a slight change in the key, so as to prevent others from stealing information. When decrypting, we change the initial value of the proposed four-dimensional chaotic system by  $10^{-10}$ , and the decrypted Lena image is shown in Fig. 8. It can be seen that even if the key changes by  $10^{-10}$ , the original image cannot be decrypted correctly.

**Fig. 8** Key sensitivity results of Lena image (a) Plaintext image of Lena (b) Decrypted image after changing  $10^{-10}$  (c) Correctly decrypted image



**Fig. 9** Histogram analysis. (a)-(c) Histogram of the R,G,B channel of the original Lena image (d)-(f) Histogram of R,G,B channel of Lena encrypted image (g)-(i) Histogram of the R,G,B channel of the original Pepper image (j)-(l) Histogram of R,G,B channel of Pepper encrypted image (m)-(o) Histogram of the R,G,B channel of the original Baboon image (p)-(r) Histogram of R,G,B channel of Baboon encrypted image



**Table 2** Results of the  $\chi^2$  test

Image	$\chi^2$ test		
	R	G	B
Lena256	282.4375	249.1172	242.9531
Pepper256	252.0391	258.9297	241.1016
Baboon512	282.5801	214.1563	264.3887

**Table 3** Information entropy of the images

Images	Plain image			Encrypted image		
	R	G	B	R	G	B
Lena256	7.7250	7.7745	7.6504	7.9974	7.9973	7.9977
Pepper256	7.4230	7.6226	7.1979	7.9972	7.9972	7.9973
Barbara256	7.6362	7.4904	7.5965	7.9968	7.9973	7.9969
Lena512	7.7322	7.7792	7.6544	7.9992	7.9993	7.9993
Baboon512	7.6583	7.3558	7.6990	7.9992	7.9994	7.9993

#### 4.4 Histogram

The histogram can intuitively reflect the pixel distribution of an image. The flatter the histogram, the more even the pixel distribution, which proves that the probability of each pixel appearing is closer, and the more difficult it is to obtain meaningful information from the image. Lena, Pepper and Baboon are encrypted with the proposed algorithm, the histograms before and after encryption are shown in Fig. 9. It can be found from the histogram that the histogram of the encrypted image is flatter than that of plain image, which can prevent useful information from being obtained from the histogram.

#### 4.5 $\chi^2$ test

The more evenly distributed the pixels and the flatter the histogram, the better the encryption should be. And  $\chi^2$  test represents how flat the histogram is. The formula for  $\chi^2$  test is shown in Eq. (20).

$$\left\{ \begin{array}{l} \chi^2 = \sum_{i=0}^{255} \frac{(v_i - v)^2}{v} \\ v = \frac{M \times N}{256} \end{array} \right. \quad (20)$$

where  $v_i$  represents the number of occurrences of pixel value  $i$ . For example, the pixel value 200 appears 89 times in the image, then  $i=200$ ,  $v_i=89$ . Table 2 presents the  $\chi^2$ -values of the three images.

**Table 4** Comparison of information entropy

Algorithm	Information entropy			
	R	G	B	Mean
The proposed algorithm	7.9974	7.9973	7.9977	7.9975
Ref. [11]	7.9972	7.9972	7.9974	7.9973
Ref. [40]	7.9917	7.9912	7.9918	7.9916
Ref. [3]	7.9973	7.9969	7.9971	7.9971
Ref. [7]	7.9974	7.9969	7.9979	7.9974

#### 4.6 Information entropy

Information entropy is an important parameter reflecting the randomness of information. It can reflect the uncertainty of image information and gray distribution in the image. For any image, the information entropy can be calculated by Eq. (21).

$$H(s) = \sum_{i=1}^n p(s_i) \log \frac{1}{p(s_i)} \quad (21)$$

where,  $p(s_i)$  represents the probability of the occurrence of pixels $s_i$ , and  $n$  is the gray level of the pixel. For an 8-bit image, its gray level is  $2^8=256$ , and  $n$  is 256 [33]. Therefore, the closer the information entropy of an image is to 8, the better its encryption effect is, and the more secure the encrypted image is. The data in Table 3 is obtained by calculating the information entropy of the three images. Table 4 is the result of comparison with other literatures. It can be seen that the proposed encryption algorithm has high security, and encrypted images have a more random distribution of pixels.

#### 4.7 Correlation of adjacent pixels

Image encryption differs from text encryption in that adjacent pixels of an image are strongly correlated with each other. If the correlation remains at a high level, the attacker may perform a statistical attack to analyze the original image. Consequently, for picture encryption techniques, lowering the correlation between pixels is crucial. The calculation formula of correlation is shown in Eq. (22).

$$\gamma_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad (22)$$

where:



**Fig. 10** Distribution of adjacent pixels. **(a)-(c)** Correlation of R,G,B components of the original image in the horizontal direction. **(d)-(f)** Correlation of R,G,B components of the cipher image in the horizontal direction. **(g)-(i)** Correlation of R,G,B components of the plain image in the vertical direction. **(j)-(l)** Correlation of R,G,B components of the cipher image in the vertical direction. **(m)-(o)** Correlation of R,G,B components of the plain image in the diagonal direction. **(p)-(r)** Correlation of R,G,B components of the cipher image in the diagonal direction

$$\begin{aligned} \text{cov}(x, y) &= \frac{1}{N} \sum_{i=1}^N (x_i - E(x_i))(y_i - E(y_i)) \\ D(x) &= \frac{1}{N} \sum_{i=1}^N (x_i - E(x_i))^2 \\ E(x) &= \frac{1}{N} \sum_{i=1}^N x_i \end{aligned}$$

In Fig. 10, the correlation before and after the encryption of the  $256 \times 256$  Lena image can be seen intuitively. A comparison of the correlation values before and after encryption can be seen in Table 5. The correlation with other literatures is compared in Table 6. It can be seen that the correlation is greatly reduced after encryption by the proposed encryption algorithm. It can effectively resist statistical attacks.

#### 4.8 Differential Attack Analysis

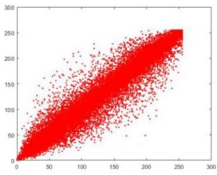
Differential attack is a type of chosen-plaintext attack. It is mainly reflected by two values, one is the number of pixel change rate (NPCR) and the other is the unified average change of intensity (UACI). The original image is encrypted by changing one pixel, and the obtained encrypted image is compared with the unaltered encrypted image. When these two values are close to the ideal values of 99.6094% and 33.4635%, respectively, it shows that the algorithm can resist differential attacks well. Their formulas are shown in Eq. (23) and Eq. (24).

$$\text{NPCR} = \frac{\sum_{i=1}^M \sum_{j=1}^N D(i, j)}{M \times N} \times 100\% \quad (23)$$

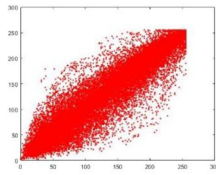
$$\text{UACI} = \frac{\sum_{i=1}^M \sum_{j=1}^N \left| P_1(i, j) - P_2(i, j) \right|}{255 \times M \times N} \times 100\% \quad (24)$$

$$\text{where : } D(i, j) = \begin{cases} 0 & P_1(i, j) = P_2(i, j) \\ 1 & P_1(i, j) \neq P_2(i, j) \end{cases}$$

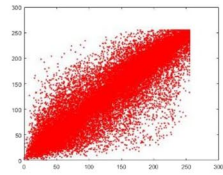
M and N represent the size of the image.  $P_1$  and  $P_2$  represent the original encrypted image and the encrypted image changed by one pixel, respectively. The NPCR and UACI values of the encryption algorithm proposed in this paper are calculated by the formula as shown in Table 7. The results compared with other literatures are shown in Table 8.



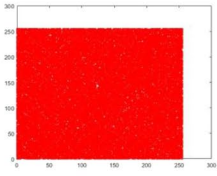
(a)



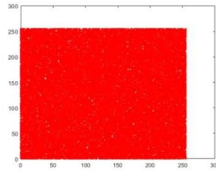
(b)



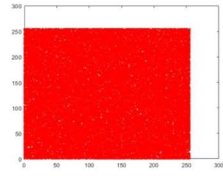
(c)



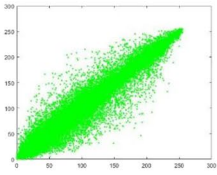
(d)



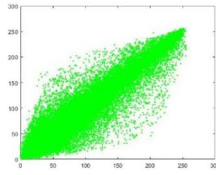
(e)



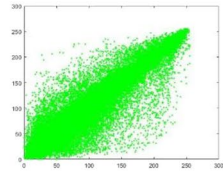
(f)



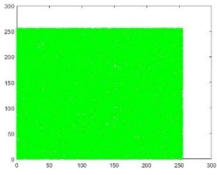
(g)



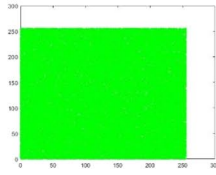
(h)



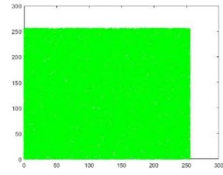
(i)



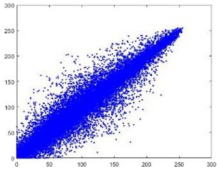
(j)



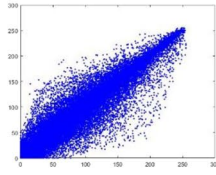
(k)



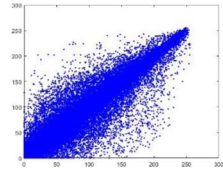
(l)



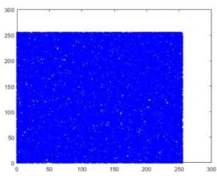
(m)



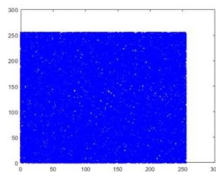
(n)



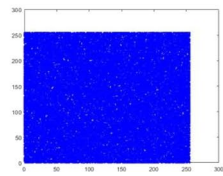
(o)



(p)



(q)



(r)

**Table 5** Correlation coefficients

	Correlation coefficients								
	Horizontal			Vertical			Diagonal		
	R	G	B	R	G	B	R	G	B
Original Image	0.9854	0.9814	0.9783	0.9688	0.9611	0.9560	0.9517	0.9415	0.9348
Ciphered Image	0.0021	0.0019	-0.0003	0.0042	-0.0042	0.0008	0.0019	-0.0019	-0.0034

**Table 6** Comparison of correlation coefficients with other literature

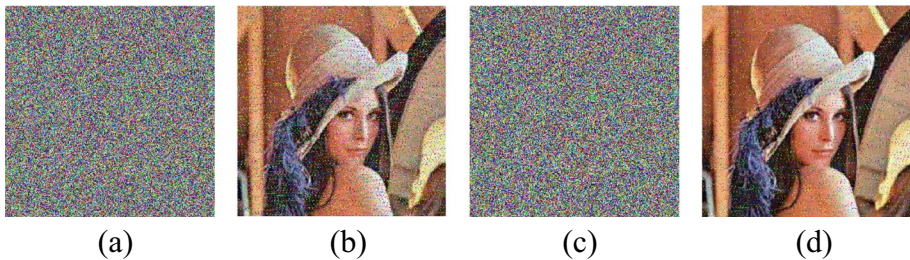
Correlation	Proposed Algorithm(256×256)	Ref. [7]	Ref. [29]
Red Channel			
Horizontal	0.0021	-0.0073	-0.0206
Vertical	0.0042	0.0010	-0.0116
Diagonal	0.0019	-0.0013	0.0097
Green Channel			
Horizontal	0.0019	0.0011	-0.0005
Vertical	-0.0042	-0.0020	0.0002
Diagonal	-0.0019	0.0078	0.0189
Blue Channel			
Horizontal	-0.0003	-0.0061	0.0016
Vertical	0.0008	0.0058	0.0133
Diagonal	-0.0034	-0.0003	-0.0123

**Table 7** NPCR and UACI performance

Image	Position of the changed pixel	NPCR(%)			UACI(%)		
		R	G	B	R	G	B
Lena256	(30,128)	99.5972	99.6078	99.6094	33.5419	33.4133	33.4754
	(205,36)	99.5956	99.6201	99.6155	33.4273	33.4365	33.4554
	(203,246)	99.6033	99.6033	99.5834	33.4449	33.4405	33.4779
	(243,9)	99.6063	99.6017	99.5972	33.5082	33.3667	33.4980
	(158,121)	99.6048	99.5911	99.5941	33.5205	33.4487	33.4749
	(182,193)	99.6033	99.6292	99.6185	33.3169	33.5032	33.4398
	(112,98)	99.6155	99.6033	99.6170	33.4034	33.4688	33.4390
	Mean	99.6037	99.6081	99.6050	33.4519	33.4397	33.4658
Lena512	(71,140)	99.6056	99.6033	99.6113	33.4754	33.4761	33.4632
	(245,247)	99.6094	99.6120	99.6189	33.5298	33.4610	33.4835
	(48,125)	99.6021	99.6017	99.6159	33.4680	33.4718	33.4524
	(30,128)	99.5934	99.6128	99.6159	33.4671	33.4254	33.4915
	Mean	99.6026	99.6075	99.6155	33.4851	33.4586	33.4727

**Table 8** Comparison results of NPCR for Lena color image

Encryption Algorithm	Red Channel		Green Channel		Blue Channel	
	NPCR(%)	UACI(%)	NPCR(%)	UACI(%)	NPCR(%)	UACI(%)
Proposed Algorithm(256×256)	99.6037	33.4519	99.6081	33.4397	99.6050	33.4658
Ref. [34]	99.6300	33.3200	99.7300	33.4500	99.7100	33.4200
Proposed Algorithm(512×512)	99.6026	33.4851	99.6075	33.4586	99.6195	33.4727
Ref. [10]	99.6103	33.4655	99.6098	33.4652	99.6089	33.4591
Ref. [40]	99.6243	33.4224	99.6185	33.4361	99.6280	33.4603



**Fig. 11** Encrypted and decrypted images with added noise. (a) Encrypted image with salt-and-pepper noise with density of 0.05 (b) Decrypted image with salt-and-pepper noise with density of 0.05 added (c) Encrypted image with salt-and-pepper noise with density of 0.1 (d) Decrypted image with salt-and-pepper noise with density of 0.1 added

#### 4.9 Noise attack

The transmission process in the real world will be accompanied by various disturbances. An image encryption algorithm must not only be able to resist attacks, but also be able to resist interference such as noise. Figure 11 is the result of adding salt and pepper noise with density 0.05 and 0.1 to the encrypted image, respectively. As can be seen from Fig. 11, the decrypted image after adding noise can basically see the information of the original image.

#### 4.10 Classic types of attacks

There are four classic types of attacks:

(1) Ciphertext-only attack

The attacker can only analyze the ciphertext to get the plaintext or key.

(2) Plaintext-known attack

Plaintext-known attack is that the attacker can obtain a set of plaintext and its ciphertext.

### (3) Chosen-plaintext attack

Chosen-plaintext attack means that the attacker can choose a set of plaintext and corresponding ciphertext.

### (4) Chosen-ciphertext attack

Chosen-ciphertext attack means that the attacker can choose some ciphertext and get the corresponding plaintext.

It can be seen that the chosen-plaintext attack is the strongest attack among the four attacks. So if the algorithm can resist the chosen-plaintext attack, then the other three attacks can be resisted by the algorithm. From the analysis of NPCR and UACI, it can be seen that when the initial value or parameters of the proposed algorithm change slightly, the encrypted image is completely different. The proposed algorithm is therefore resistant to four classical types of attacks.

## 4.11 Running time

The efficiency of the algorithm can be evaluated by the running time. The test environment of the proposed image encryption algorithm is MATLAB2019b (processor: Intel CORE i9-13900H RAM:16.00G), and the running time is 9.34 s after the encryption with  $256 \times 256$  Lena image.

## 5 Conclusion

This paper proposes an image encryption algorithm that diffuses pixels according to multi-based decomposition. In this algorithm, the proposed four-dimensional chaotic system is used to generate chaotic sequences, which are decomposed into multiple digits and reassembled, and the final encrypted images are obtained by pixel-level FYTS scrambling and other operations. After testing the security with standardized images, the entropy of information is close to the ideal value, the correlation is low, and the algorithm can resist differential attacks, violent attacks, etc. In addition, we also compare the proposed algorithm with other algorithms, which shows that the image encryption algorithm has good anti-attack ability and high security. There are limitations, though: the algorithm is relatively slow. The running time may be improved in the future, but it still has high security and can be applied to the field of image encryption.

### Declarations

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

## References

1. Askar SS, Karawia AA, Al-Khedhairi A, Al-Amr FS (2019) An algorithm of image encryption using logistic and two-dimensional chaotic economic maps. *Entropy* 21(1):44
2. Broumandnia A (2019) The 3D modular chaotic map to digital color image encryption. *Futur Gener Comput Syst* 99:489–499
3. Chai X, Fu X, Gan Z, Lu Y, Chen Y (2019) A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Process* 155:44–62
4. Chen L, Zhao D, Ge F (2013) Image encryption based on singular value decomposition and Arnold transform in fractional domain. *Opt Commun* 291:98–103
5. Cun Q, Tong X, Wang Z, Zhang M (2021) Selective image encryption method based on dynamic DNA coding and new chaotic map. *Optik* 243:167286
6. Gao X (2021) Image encryption algorithm based on 2D hyperchaotic map. *Opt Laser Technol* 142:107252
7. Girdhar A, Kumar V (2018) A RGB image encryption technique using Lorenz and Rossler chaotic system on DNA sequences. *Multimed Tools Appl* 77:27017–27039
8. Gong LH, Luo HX, Wu RQ, Zhou NR (2022) New 4D chaotic system with hidden attractors and self-excited attractors and its application in image encryption based on RNG. *Phys A: Stat Mech Appl* 591:126793
9. Hua Z, Zhou B, Zhou Y (2018) Sine chaotification model for enhancing chaos and its hardware implementation. *IEEE Trans Ind Electron* 66(2):1273–1284
10. Huang L, Cai S, Xiong X, Xiao M (2019) On symmetric color image encryption system with permutation-diffusion simultaneous operation. *Opt Lasers Eng* 115:7–20
11. Khalil N, Sarhan A, Alshewimy MA (2021) An efficient color/gray-scale image encryption scheme based on hybrid chaotic maps. *Opt Laser Technol* 143:107326
12. Kovalchuk A, Izonin I, Riznyk O (2019) An efficient image encryption scheme using projective transformations. *Procedia Comput Sci* 160:584–589
13. Kovalchuk A, Izonin I, Kustra N (2019) Information protection service using topological image coverage. *Procedia Comput Sci* 160:503–508
14. Kumar CM, Vidhya R, Brindha M (2022) An efficient chaos based image encryption algorithm using enhanced Thorp shuffle and chaotic convolution function. *Appl Intell* 52(3):2556–2585
15. Liu W, Sun K, Zhu C (2016) A fast image encryption algorithm based on chaotic map. *Opt Lasers Eng* 84:26–36
16. Luo Y, Yu J, Lai W, Liu L (2019) A novel chaotic image encryption algorithm based on improved baker map and logistic map. *Multimed Tools Appl* 78:22023–22043
17. Malik DS, Shah T (2020) Color multiple image encryption scheme based on 3D-chaotic maps. *Math Comput Simul* 178:646–666
18. Matthews R (1989) On the derivation of a “chaotic” encryption algorithm. *Cryptologia* 13(1):29–42
19. Morris B, Rogaway P, Stegers T (2009) How to encipher messages on a small domain: deterministic encryption and the Thorp shuffle. In *Advances in Cryptology-CRYPTO 2009: 29th Annual International Cryptology Conference*, Santa Barbara, CA, USA, August 16–20, 2009. Proceedings, Springer, Berlin Heidelberg, pp. 286–302
20. Ni Z, Kang X, Wang L (2016, August) A novel image encryption algorithm based on bit-level improved Arnold transform and hyper chaotic map. In *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*. IEEE. pp. 156–160
21. Pareek NK, Patidar V, Sud KK (2006) Image encryption using chaotic logistic map. *Image Vis Comput* 24(9):926–934
22. Rakheja P, Singh P, Vig R (2020) An asymmetric image encryption mechanism using QR decomposition in hybrid multi-resolution wavelet domain. *Opt Lasers Eng* 134:106177
23. Sang Y, Sang J, Alam MS (2022) Image encryption based on logistic chaotic systems and deep autoencoder. *Pattern Recogn Lett* 153:59–66
24. Shahna KU, Mohamed A (2020) A novel image encryption scheme using both pixel level and bit level permutation with chaotic map. *Appl Soft Comput* 90:106162
25. Shifa A, Afgan MS, Asghar MN, Fleury M, Memon I, Abdullah S, Rasheed N (2018) Joint crypto-stego scheme for enhanced image protection with nearest-centroid clustering. *IEEE Access* 6:16189–16206
26. Talhaoui MZ, Wang X (2021) A new fractional one dimensional chaotic map and its application in high-speed image encryption. *Inf Sci* 550:13–26
27. Tang Z, Song J, Zhang X, Sun R (2016) Multiple-image encryption with bit-plane decomposition and chaotic maps. *Opt Lasers Eng* 80:1–11

28. ul Haq T, Shah T (2021) 4D mixed chaotic system and its application to RGB image encryption using substitution-diffusion. *J Inf Secur Appl* 61:102931
29. ur Rehman A, Liao X, Ashraf R, Ullah S, Wang H (2018) A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2. *Optik* 159:348–367
30. Wang X, Chen Y (2021) A new chaotic image encryption algorithm based on L-shaped method of dynamic block. *Sens Imaging* 22:1–30
31. Wang X, Zhang M (2021) An image encryption algorithm based on new chaos and diffusion values of a truth table. *Inf Sci* 579:128–149
32. Wang X, Guan N, Liu P (2022) A selective image encryption algorithm based on a chaotic model using modular sine arithmetic. *Optik* 258:168955
33. Wang S, Peng Q, Du B (2022) Chaotic color image encryption based on 4D chaotic maps and DNA sequence. *Opt Laser Technol* 148:107753
34. Wu X, Zhu B, Hu Y, Ran Y (2017) A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps. *IEEE Access* 5:6429–6436
35. Xiong Y, Quan C, Tay CJ (2018) Multiple image encryption scheme based on pixel exchange operation and vector decomposition. *Opt Lasers Eng* 101:113–121
36. Yang YG, Zou L, Zhou YH, Shi WM (2020) Visually meaningful encryption for color images by using Qi hyper-chaotic system and singular value decomposition in YCbCr color space. *Optik* 213:164422
37. Yang YG, Wang BP, Pei SK, Zhou YH, Shi WM, Liao X (2021) Using M-ary decomposition and virtual bits for visually meaningful image encryption. *Inf Sci* 580:174–201
38. Yun-Peng Z, Wei L, Shui-Ping C, Zheng-Jun Z, Xuan N, Wei-di D (2009, October) Digital image encryption algorithm based on chaos and improved DES. In 2009 IEEE international conference on systems, man and cybernetics, IEEE, pp. 474–479
39. Zhang YQ, Wang XY (2015) A new image encryption algorithm based on non-adjacent coupled map lattices. *Appl Soft Comput* 26:10–20
40. Zhang YQ, He Y, Li P, Wang XY (2020) A new color image encryption scheme based on 2DNLCML system and genetic operations. *Opt Lasers Eng* 128:106040
41. Zhou G, Zhang D, Liu Y, Yuan Y, Liu Q (2015) A novel image encryption algorithm based on chaos and Line map. *Neurocomputing* 169:150–157
42. Zhou W, Wang X, Wang M, Li D (2022) A new combination chaotic system and its application in a new Bit-level image encryption scheme. *Opt Lasers Eng* 149:106782
43. Zhu H, Ge J, Qi W, Zhang X, Lu X (2022) Dynamic analysis and image encryption application of a sinusoidal-polynomial composite chaotic system. *Math Comput Simul* 198:188–210

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.