

Optical image encryption algorithm based on a new four-dimensional memristive hyperchaotic system and compressed sensing

Yang Du(都洋), Guoqiang Long(隆国强), Donghua Jiang(蒋东华) , Xiuli Chai(柴秀丽), and Junhe Han(韩俊鹤)
Citation: Chin. Phys. B, 2023, 32 (**11**): 114203. DOI: 10.1088/1674-1056/acef08
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Color-image encryption scheme based on channel fusion and spherical diffraction

Jun Wang(王君), Yuan-Xi Zhang(张沅熙), Fan Wang(王凡), Ren-Jie Ni(倪仁杰), and Yu-Heng Hu(胡玉衡)
Chin. Phys. B, 2022, 31 (**3**): 034205. DOI: 10.1088/1674-1056/ac1b87

Computational ghost imaging with deep compressed sensing

Hao Zhang(张浩), Yunjie Xia(夏云杰), and Deyang Duan(段德洋)
Chin. Phys. B, 2021, 30 (**12**): 124209. DOI: 10.1088/1674-1056/ac0042

Compressive imaging based on multi-scale modulation and reconstruction in spatial frequency domain

Fan Liu(刘璠), Xue-Feng Liu(刘雪峰), Ruo-Ming Lan(蓝若明), Xu-Ri Yao(姚旭日), Shen-Cheng Dou(窦申成), Xiao-Qing Wang(王小庆), and Guang-Jie Zhai(翟光杰)
Chin. Phys. B, 2021, 30 (**1**): 014208. DOI: 10.1088/1674-1056/abb3ea

An image compressed sensing algorithm based on adaptive nonlinear network

Yuan Guo(郭媛), Wei Chen(陈炜), Shi-Wei Jing(敬世伟)
Chin. Phys. B, 2020, 29 (**5**): 054203. DOI: 10.1088/1674-1056/ab7b4e

Compressed ghost imaging based on differential speckle patterns

Le Wang(王乐), Shengmei Zhao(赵生妹)
Chin. Phys. B, 2020, 29 (**2**): 024204. DOI: 10.1088/1674-1056/ab671a

---

Optical image encryption algorithm based on a new four-dimensional memristive hyperchaotic system and compressed sensing

Yang Du(都洋)[1], Guoqiang Long(隆国强)[2], Donghua Jiang(蒋东华)[3], Xiuli Chai(柴秀丽)[2,†], and Junhe Han(韩俊鹤)[1,‡]

[1] *Center for Physics of Low-Dimensional Materials, Henan Joint International Research Laboratory of New Energy Materials and Devices, School of Physics and Electronics, Henan University, Kaifeng* 475004, *China*

[2] *School of Artificial Intelligence, Henan Engineering Research Center for Industrial Internet of Things, Henan University, Zhengzhou* 450046, *China*

[3] *School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou* 510006, *China*

(Received 30 May 2023; revised manuscript received 13 July 2023; accepted manuscript online 11 August 2023)

Some existing image encryption schemes use simple low-dimensional chaotic systems, which makes the algorithms insecure and vulnerable to brute force attacks and cracking. Some algorithms have issues such as weak correlation with plaintext images, poor image reconstruction quality, and low efficiency in transmission and storage. To solve these issues, this paper proposes an optical image encryption algorithm based on a new four-dimensional memristive hyperchaotic system (4D MHS) and compressed sensing (CS). Firstly, this paper proposes a new 4D MHS, which has larger key space, richer dynamic behavior, and more complex hyperchaotic characteristics. The introduction of CS can reduce the image size and the transmission burden of hardware devices. The introduction of double random phase encoding (DRPE) enables this algorithm has the ability of parallel data processing and multi-dimensional coding space, and the hyperchaotic characteristics of 4D MHS make up for the nonlinear deficiency of DRPE. Secondly, a construction method of the deterministic chaotic measurement matrix (DCMM) is proposed. Using DCMM can not only save a lot of transmission bandwidth and storage space, but also ensure good quality of reconstructed images. Thirdly, the confusion method and diffusion method proposed are related to plaintext images, which require both four hyperchaotic sequences of 4D MHS and row and column keys based on plaintext images. The generation process of hyperchaotic sequences is closely related to the hash value of plaintext images. Therefore, this algorithm has high sensitivity to plaintext images. The experimental testing and comparative analysis results show that proposed algorithm has good security and effectiveness.

Keywords: memristor, hyperchaotic system, compressed sensing, fractional Fourier transform, optical image encryption

PACS: 42.30.–d, 42.30.Kq, 42.30.Va, 42.30.Wb **DOI:** 10.1088/1674-1056/acef08

1. Introduction

With the rapid development of the Internet, artificial intelligence technology is more and more closely connected with people's lives,[1–6] and different types of data can be rapidly disseminated and diffused through the network, in which the issue of image information security has become more and more serious.[7–9] In order to address the above issues, some image encryption schemes have been proposed.

Fridrich[10] first applied chaotic maps for image encryption in 1997. After that, some simple one-dimension (1D) chaotic maps[11–15] were widely proposed by scholars and applied to the field of image encryption. However, the security of 1D chaotic map is low because of its small key space and vulnerability to violent attacks. Therefore, in order to improve the shortcomings of 1D chaotic map, some scholars have provided image encryption algorithms based on high-dimensional chaotic map by improving 1D chaotic map.[16–18] Because of the excellent nonlinear characteristics of memristor, some scholars applied it for chaotic maps to produce systems with stronger chaotic characteristics. Li *et al.*[19] designed an image encryption algorithm combining memristor and chaotic tent map. Wang *et al.*[20] presented a new hyperchaotic system with two memristors as nonlinear terms in the system. Ye *et al.*[21] proposed an image encryption scheme based on seven-order hybrid memristive chaotic circuit. Peng *et al.*[22] designed a new memristor chaotic circuit based on Lorenz-like circuits and applied it to image encryption. Memristor chaotic systems have great application value in the field of image encryption, so it is urgent and necessary to continue to explore and study memristor chaotic systems with more complex dynamic behavior.

Images have the characteristics of large amount of data and high information redundancy. In order to remove the redundant information of images and improve the transmission and storage efficiency of algorithms, scholars applied the theory of compressed sensing[23] to the image encryption process, and proposed some image encryption algorithms based on CS. Xu *et al.*[24] constructed two measurement matrices to

†Corresponding author. E-mail: chaixiuli@henu.edu.cn
‡Corresponding author. E-mail: junhh@henu.edu.cn

© 2023 Chinese Physical Society and IOP Publishing Ltd

measure the image from two directions, so as to obtain better compression performance with higher efficiency. Zhou *et al.*[25] proposed an image encryption algorithm based on CS and fractional Mellin transform. Yang *et al.*[26] constructed the measurement matrix through the fractional hyperchaotic system, and introduced the DNA coding technology into the encryption process, which improved the complexity and security of the algorithm. Zhou *et al.*[27] used two-dimensional (2D) CS to compress and encrypt images and cyclic shift to improve the security of the algorithm. Although CS is introduced into the above scheme, the image reconstruction quality is poor and the robustness of the algorithm cannot be guaranteed. Therefore, the design of a robust CS encryption algorithm has attracted attention.[28,29] Luo *et al.*[30] constructed a measurement matrix by Lissajous map and logistic map, and designed an encryption algorithm with strong robustness. Zhang *et al.*[31] put forward a robust image compression–encryption scheme, which uses 2D projection gradient algorithm for image reconstruction, but the security of the algorithm is lacking. Therefore, it is necessary to design a compression–encryption algorithm with both robustness and security. Based on the previous discussion of image encryption algorithms based on CS, it is worth noting that while some algorithms have used CS to achieve efficient image compression, the trade-off is reduced image quality and potential security vulnerabilities in the encryption process.[32–34] This has led to a growing interest in designing image encryption algorithms that balance robustness and security. One approach that has gained attention is the use of optical encryption methods, such as the double random-phase encoding algorithm.

In 1995, Refregier *et al.*[35] first proposed an optical image encryption algorithm based on double random-phase encoding (DRPE). Optical transformation has its unique high-speed parallel data processing ability and multi-dimensional coding space.[36] Optical image encryption algorithms based on DRPE and different optical transformations have been widely proposed. The authors in Refs. [37–40] applied different optical transformations to realize image encryption in DRPE, such as fractional wavelet transform, Hartley transform, fractional Fourier transform, fractional Mellin transform, and so on. However, since DRPE is a linear system, the corresponding image encryption is vulnerable to known-plaintext and chosen-plaintext attacks. Fortunately, the nonlinearity of chaotic systems can compensate for the lack of nonlinearity of DRPE. Some algorithms[41–43] combined chaotic system with DRPE to improve the security of the algorithm, but these algorithms generally used a single integer-order chaotic system, and the key space was small. The authors in Refs. [44,45] proposed some optical encryption algorithms combining DRPE with CS. However, these algorithms have not been comprehensively analyzed, so their security needs to be verified.

Recently, Jiang *et al.*[2] proposed a cross-ring neural network model based on the Hopfield neural network to reduce bandwidth saturation in communication channels and securing the transferred medical data. Yu *et al.*[3] constructed a 4D memristive Hopfield neural network (MHNN) which can perform complex dynamics and performs well in image encryption applications for the significant complexity of multi-scroll. Su *et al.*[4] proposed a multi-image visual encryption algorithm based on region of interest (ROI) to ensure the security of multiple image regions of interest. This design scheme can not only reconstruct the significant part without damage, but also reduce the storage space.

Generally speaking, despite being powerful already, the current image encryption schemes suffer from the following problems:

(i) Most of image encryption schemes adopt simple low-dimensional chaotic systems, which have small key space and chaotic interval, and thus they are vulnerable to violent attacks and have insufficient security.

(ii) In some CS-based schemes, the measurement matrix is a random Gaussian matrix, and the whole measurement matrix needs to be transmitted and stored as a key when decrypting, which takes up a lot of transmission bandwidth and storage space.

(iii) Although some schemes use chaotic sequences to construct deterministic measurement matrices, most of them use a single chaotic system. The parameters and initial values of chaotic systems are simple and fixed, and the performance of the constructed measurement matrices is poor, the quality of image reconstruction is not high, and the robustness of the algorithm is unsatisfactory.

(iv) Some schemes are not related to plaintext images, and are hence vulnerable to known-plaintext and chosen-plaintext attacks.

In order to solve the problems in the above schemes, this paper proposes a new image encryption scheme based on 4D MHS, CS, and DRPE. The main contribution of the proposed scheme is as follows:

(I) A new 4D MHS is designed, which generate four hyperchaotic sequences simultaneously. The structure of 4D MHS is simple and easy to implement. The results show that 4D MHS has a larger key space, richer dynamical behavior, and more complex hyperchaotic properties.

(II) A method of constructing deterministic chaotic measurement matrix (DCMM) is proposed by combining 4D MHS and 1D CCS. The system parameters and initial values of 1D CCS are determined from four hyperchaotic sequences of 4D MHS, in which the system parameters u_i and v_i are no longer fixed values and will change in each iteration, which improves the randomness of the DCMM. Compared with other com-

monly used random measurement matrices, using the DCMM can not only save a lot of transmission bandwidth and storage space, but also ensure the quality of image reconstruction.

(III) The proposed scheme combines 4D MHS, CS, and DRPE. The introduction of CS reduces the amount of image data and reduces the transmission burden of hardware devices. DRPE has unique high-speed parallel data processing ability and multi-dimensional coding space ability, and the excellent hyperchaotic characteristics of 4D MHS can make up for the lack of nonlinearity of DRPE. Among them, the optical transformation adopts fractional Fourier transform, which increases the key space of the algorithm and improves the security of the algorithm.

(IV) The confusion method and the diffusion method are proposed. The confusion and diffusion process needs row key k_r and column key k_c, which are closely related to plaintext images. Four hyperchaotic sequences are used in the construction process of the DCMM, DRPE, confusion and diffusion process, and the generation of hyperchaotic sequences is determined by the hash value of plaintext images and external keys. The proposed algorithm is sensitive to plaintext images and can effectively resist known-plaintext and chosen-plaintext attacks.

The rest of this paper is organized as follows: In Section 2, the preliminaries of the work are described. In Section 3, the specific process of image encryption and decryption in this scheme is introduced in detail. In Section 4, simulation results and performance analyses are provided. In Section 5, our work is summarized.

2. Preliminaries

2.1. One-dimensional Chebyshev–Chebyshev system (1D CCS)

The one-dimensional CCS[13] is a new chaotic map constructed by the difference between the output sequences of two 1D Chebyshev chaotic maps. Compared with 1D Chebyshev chaotic map, the 1D CCS has better chaotic characteristics and larger chaotic interval. Its mathematical expression is

$$\begin{aligned} x_{n+1} &= \cos((u+1)\times \arccos(x_n))\times 2^k \\ &\quad - \text{floor}(\cos(u+1)\times \arccos(x_n)\times 2^k), \end{aligned} \tag{1}$$

where the system parameter $u \in (0,10]$, $v \in [8,20]$, and the initial value of the system is s_0.

2.2. Compressed sensing (CS)

Donoho *et al.*[24] put forward compressed sensing theory in 2006. The CS pointed out that if the signal is sparse or sparse in a transform domain, the high-dimensional signal can be projected into the low-dimensional space through the measurement matrix independent of sparse basis, and then the original signal can be recovered more accurately by solving the convex optimization problem.

Assume that the signal $x_{N\times 1}$ is K-sparse and transforms it under the standard orthogonal base Ψ.

$$x = \Psi \cdot s. \tag{2}$$

There are only K ($K \ll N$) non-zero coefficients in the vector s. The original signal is projected by using the measurement matrix $\Phi_{M\times N}$ independent of Ψ, which is expressed as

$$y = \Phi \cdot x \tag{3}$$

M ($M \ll N$) linear measurements y can be obtained, which already contain enough information for reconstructing the signal x, and y can be expressed as

$$y = \Phi \cdot x = \Phi \cdot \Psi \cdot s = \Theta \cdot s, \tag{4}$$

where Θ is sensing matrix, x is reconstructed by solving the optimal problem, and the convex optimization problem is solved by using the minimum l_p ($0 < p < 1$) norm, which is expressed as

$$\min ||s||_p \text{s.t.y} = \Theta \cdot s, \tag{5}$$

where $||s||_p$ is the l_p norm of the vector s and the measurement matrix $\Phi_{M\times N}$ has to satisfy the restricted isometry property (RIP). The original signal x can be recovered from the dimensionality reduction process. At present, there are many reconstruction algorithms, such as basis pursuit (BP), orthogonal matching pursuit (OMP), smoothed l_0 norm (SL_0), and so on.

2.3. Fractional Fourier transform (FrFT)

In 1980, Namias first put forward the definition and properties of fractional Fourier transform.[46,47] In 1987, Mcbride and others perfected the concept of Namias, and they gave a strict mathematical definition of the fractional Fourier transform.[48] The integral form of the fractional Fourier transform of the function $f(x)$ is

$$F^{\alpha}\{f(x)\}(u) = \int_{-\infty}^{+\infty} k_{\alpha}(x,u) f(x)\,\mathrm{d}x, \tag{6}$$

where $k_{\alpha}(x,u)$ is its kernel function and its state equation is

$$k_{\alpha}(x,u) = \begin{cases} A_{\alpha}\exp[\mathrm{i}\pi(u^2\cot\theta_{\alpha} - 2xu\csc\theta_{\alpha} + x^2\cot\theta_{\alpha})], & \alpha \neq 2n, \\ \delta(x+u), & \alpha = 4n+2, \\ \delta(x-u), & \alpha = 4n. \end{cases} \tag{7}$$

Here $A_\alpha = \sqrt{1-\mathrm{i}\cot\theta_\alpha}$, $\theta_\alpha = \alpha\pi/2$.

The fractional Fourier transform of the two-dimensional function $f(x, y)$ is given by

$$F^y F^x\{f(x,y)\}(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} k_{y,x}(x,y,u,v) f(x,y)\,\mathrm{d}x\mathrm{d}y, \quad (8)$$

where $k_{y,x}(x, y, u, v)$ its kernel function

$$\begin{aligned} k_{y,x}(x,y,u,v) = {} & \sqrt{1-\mathrm{i}\cot\alpha}\sqrt{1-\mathrm{i}\cot\beta} \\ & \times \exp[\mathrm{i}\pi((x^2+u^2)\cot\alpha) - xu\csc\alpha] \\ & \times \exp[\mathrm{i}\pi((y^2+v^2)\cot\beta) - yv\csc\beta]. \quad (9) \end{aligned}$$

2.4. Latin square

Latin square[49] is an $n \times n$ square matrix. In the $n \times n$ square matrix, there are exactly n different elements, and each different element only appears once in each row or column. Figure 1 gives four examples of Latin square matrix with different orders. The pseudo-code for generating the Latin square matrix[50] is shown in Algorithm 1.

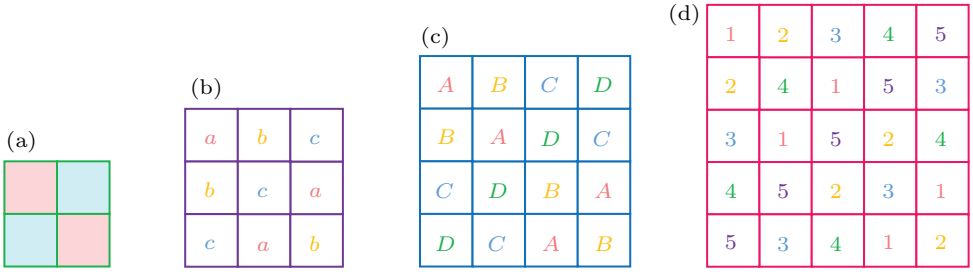


Fig. 1. Latin square example. (a) 2×2, (b) 3×3, (c) 4×4, (d) 5×5.

Algorithm 1: A Latin square matrix generation method

Input: Q_1 and Q_2 are two sequences of equal length (N)
Output: L is a Latin square of order N

```
Q_seed = sort(Q_1);
Q_shift = sort(Q_2);
for i = 1 : N do
    L(i,:) = circshift (Q_seed, Q_shift(i));
end for
```

Here sort(x) is a sort function, which sorts the sequence x in ascending order. The circshift (x, y) means to cyclically move the sequence x from left to right by y positions.

3. The proposed image encryption and decryption scheme

3.1. The new 4D memristive hyperchaotic system (4D MHS)

In this section, a new 4D memristive hyperchaotic system is proposed. By introducing a quadratic memristor model.[51] into a three-dimensional (3D) Lorenz-like chaotic system, a 4D chaotic system fed back by the memristor is obtained. After that, the performance analysis of the new system proves that it has excellent hyperchaotic characteristics. The memristor model is shown below:

$$\begin{cases} q(\varphi) = -\alpha\varphi + 0.5\beta\varphi|\varphi|, \\ W(\varphi) = dq(\varphi)/d\varphi = -\alpha + \beta|\varphi|, \end{cases} \quad (10)$$

where q is the charge, φ is the magnetic flux, α, β are positive real numbers, and $W(\varphi)$ is the memristor function of the quadratic magnetron memristor model.

3.1.1. Mathematical model

The mathematical model of the 4D MHS can be defined by

$$\begin{cases} \dot{x} = a(y-x), \\ \dot{y} = cy - xz + kW(w)x, \\ \dot{z} = y^2 - bz, \\ \dot{w} = x, \end{cases} \quad (11)$$

where a, b, c are the control parameters of the system, x, y, z, w are the state variables of the system, k is the coefficient associated with the memristor.

3.1.2. Phase–orbit diagram

The phase–orbit diagram of chaotic attractor is the motion trajectory of the system, which reflects the change in the motion state of the system. When the chaotic attractor appears in the phase space, it can be roughly considered that the system has chaotic behavior.

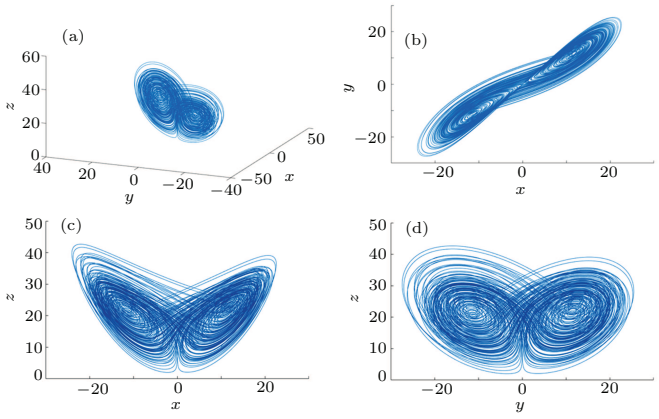


Fig. 2. Phase–orbit diagrams of 4D MHS: (a) x–y–z space, (b) x–y plane, (c) x–z plane, (d) y–z plane.

When the system parameters are set to $a = 35$, $b = 6$, $c = 20$, $k = 6$, $\alpha = 0.1$, $\beta = 0.02$, and the initial values of the system are set to $x_0 = 10$, $y_0 = 10$, $z_0 = 10$, $w_0 = 10$, the phase–orbit diagram of the system is shown in Fig. 2. It can be seen that there are two chaotic attractors in the phase space, which have complex folding and stretching trajectories, ergodicity and boundedness in a specific attraction domain, and the dynamic behavior is complex. So the system can be considered to be in a chaotic state.

3.1.3. Analysis of system dissipation

The trajectory of a chaotic system has the characteristics of contraction and divergence at the same time. The contraction of the trajectory makes the system dissipative, while the local instability of the trajectory makes the system divergent. When the system is dissipative, it can converge to a finite set, which indicates that the system is bounded.

The dissipative analysis of the system is carried out to judge whether the system converges, and the certainty and global stability of the system can be analyzed theoretically. The divergence expression of the system can be derived from Eq. (11):

$$\nabla \cdot V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{w}}{\partial w} = -a + c - b. \tag{12}$$

Under the conditions of system parameters and initial values set in Subsection 3.1.2, the divergence of the system can be calculated by Eq. (12) as $\nabla \cdot V = -a + c - b = -35 + 20 - 6 = -21$. Because $\nabla \cdot V = -21 < 0$, the system is dissipative and converges at e^{-21t} speed.

3.1.4. Analysis of system equilibrium points and stability

It is generally believed that chaotic systems need at least one unstable equilibrium point. In order to solve the equilibrium point of the system, let the right side of the system equations defined by Eq. (11) be equal to zero, and under the above parameters, the equilibrium point of the system is point set $P = \{x = y = z = 0, w = C\}$, where C is any real number. It can be seen that there are infinitely many equilibrium points in the system at this time.

Taking the equilibrium point $O = \{x = y = z = w = 0\}$ as an example to analyze whether the system is stable at this equilibrium point, the Jacobian matrix of the system at the equilibrium point O is

$$J = \begin{bmatrix} -a & a & 0 & 0 \\ kW - z & c & -x & k\beta x \\ 0 & 2y & -b & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -35 & 35 & 0 & 0 \\ -0.6 & 20 & 0 & 0 \\ 0 & 0 & -6 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{13}$$

Assuming that λ is the eigenvalue and I is a fourth-order identity matrix, four eigenvalues $\lambda_1 = 19.6155$, $\lambda_2 = 0$, $\lambda_3 = -6$, and $\lambda_4 = -34.6155$ of the system can be obtained by solving determinant $|\lambda I - J| = 0$.

Because the eigenvalue of the system is a positive real root, a zero root, and two negative real roots, according to the Routh–Hurwitz stability criterion, the system is unstable at the equilibrium point O. Therefore, the system has at least one unstable equilibrium point.

3.1.5. Analysis of system Lyapunov exponent and dimension

Lyapunov exponent is an important quantitative index to measure the chaotic characteristics of the system, which represents the average exponential rate of convergence or divergence between adjacent orbits in the phase space.

Every dimension of a chaotic system has a corresponding Lyapunov exponent. If the system has only one positive Lyapunov exponent, it indicates that the system is a chaotic system. If the system has two or more positive Lyapunov exponents, it shows that the system is hyperchaotic, with two or more trajectory separation directions, and its dynamic behavior on the trajectory is more complex and hyperchaotic.

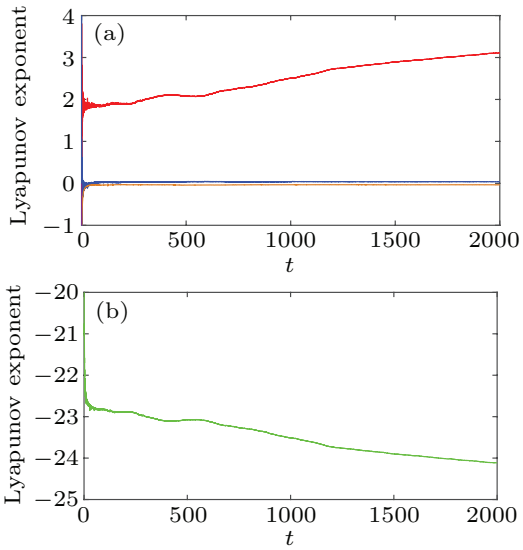


Fig. 3. Lyapunov exponents diagram of the 4D MHS: (a) exponents diagram of L_1, L_2, and L_3, (b) exponents diagram of L_4.

Under the conditions of the above system parameters and initial values, we solve the Lyapunov exponent of the system by the Jacobian matrix method, as shown in Fig. 3. The Lyapunov exponents of the system are $L_1 = 3.1104$, $L_2 = 0.0392$, $L_3 = -0.0350$, and $L_4 = -24.1146$. The system has two positive Lyapunov exponents L_1 and L_2, and the sum of all Lyapunov exponents is less than zero, so the system is in hyperchaotic state.

$$D_L = j + \frac{1}{|L_{j+1}|}\sum_{i=1}^{j} L_i = 3 + \frac{L_1 + L_2 + L_3}{|L_4|}. \tag{14}$$

The Lyapunov dimension is calculated by Eq. (14), and the Lyapunov dimension $D_L = 3.1292$ is a fractional order, which indicates that the system has a more complex fractal structure.

3.1.6. Effect of parameter changes on system dynamics

System parameters have a very important influence on the dynamic behavior of chaotic systems. Keeping other parameters unchanged, taking the parameter b as a variable, with the change of parameter b in the interval [0, 8], the bifurcation diagram of system state variable x and the Lyapunov exponents diagram of the system are shown in Fig. 4. It can be seen that when the parameter b changes, the system presents a variety of motion states, and its motion states correspond to the Lyapunov exponents diagram. When $b \in [2, 8]$, the system has two positive Lyapunov exponents, and the system is in a hyperchaotic state.

3.1.7. Analysis of the NIST SP800-22 standard test

Currently, the most well-established standard for evaluating the randomness of sequences is the NIST SP800-22 standard test, which has 15 tests. If the test result is P-value > 0.01, the sequence passes the test.

In order to further analyze the performance of the 4D MHS, four hyperchaotic sequences X, Y, Z, W are tested by the NIST SP800-22 standard, and the results are shown in Table 1. It can be seen that all hyperchaotic sequences have passed the NIST SP800-22 standard test, which shows that the 4D MHS has excellent hyperchaotic characteristics, and the hyperchaotic sequences generated by it are safe enough for image encryption.

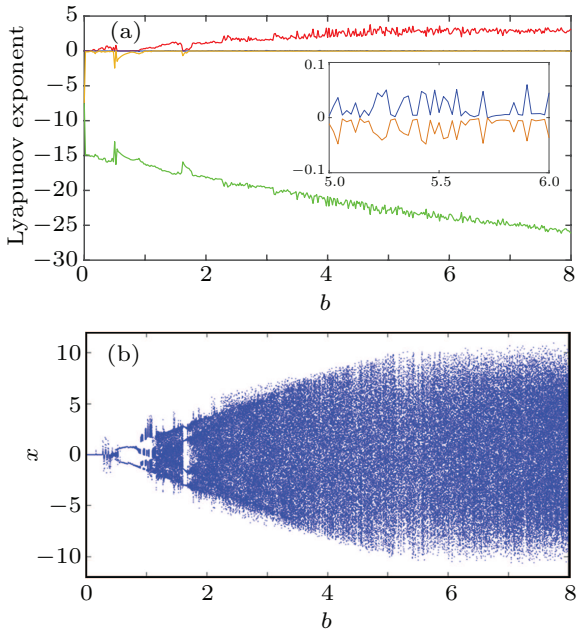


Fig. 4. (a) Lyapunov exponents diagram of the system varying with the parameter b, (b) bifurcation diagram of system state variable x varying with the parameter b.

Table 1. NIST SP800-22 standard test results of hyperchaotic sequences X, Y, Z, and W.

Test Items	P-value				Results
	X	Y	Z	W	
Frequency	0.595549	0.554420	0.514124	0.574903	pass
Block frequency	0.637119	0.334538	0.739918	0.514124	pass
Cumulative sums	0.616305	0.275709	0.678686	0.401199	pass
Runs	0.334538	0.334538	0.474986	0.366918	pass
Longest run	0.122325	0.494392	0.867692	0.534146	pass
Rank	0.779188	0.058984	0.779188	0.637119	pass
FFT	0.897763	0.816537	0.096578	0.058984	pass
Non-overlapping template	0.851383	0.834308	0.964295	0.616305	pass
Overlapping template	0.145326	0.897763	0.867692	0.055361	pass
Universal	0.798139	0.419021	0.978072	0.455937	pass
Approximate entropy	0.334538	0.224821	0.779188	0.574903	pass
Random excursions	0.964295	0.834308	0.637119	0.772760	pass
Random excursions variant	0.689019	0.964295	0.980883	0.568055	pass
Serial	0.474986	0.534146	0.739918	0.779188	pass
Linear complexity	0.657933	0.798139	0.455937	0.401199	pass

3.2. Construction method of deterministic chaotic measurement matrix

In this section, a method of constructing deterministic chaotic measurement matrix (DCMM) is proposed based on 4D MHS and 1D CCS. Compared with some commonly used random measurement matrices that require the entire measurement matrix to be stored and transmitted as the key, DCMM only requires the 256-bit hash key K associated with the plaintext image and the external keys t_i ($i = 1, 2, 3, 4$). It can effectively reduce the space consumption during key storage and transmission, and also ensure good image reconstruction quality.

Set the compression rate CR $= 0.5$, the size of plaintext image P is $M \times N$, and the size of DCMM is $m \times n$ ($m = M/2$, $n = N/2$). The construction process of DCMM is shown in Fig. 5, and the specific steps are as follows:

Step 1 The 256-bit hash key K of the plaintext image $P_{M\times N}$ is calculated, and K is converted into thirty-two decimal numbers k_i ($i = 1, 2, \ldots, 32$) in groups of eight. The initial values x_0, y_0, z_0, w_0 of the 4D MHS are generated by k_i and the external keys t_i, as shown below:

$$\begin{cases} a = (k_1 \otimes k_5 \otimes k_9 \otimes k_{13} + k_{17} \otimes k_{21} \otimes k_{25} \otimes k_{29})/256, \\ b = (k_2 \otimes k_6 \otimes k_{10} \otimes k_{14} + k_{18} \otimes k_{22} \otimes k_{26} \otimes k_{30})/256, \\ c = (k_3 \otimes k_7 \otimes k_{11} \otimes k_{15} + k_{19} \otimes k_{23} \otimes k_{27} \otimes k_{31})/256, \\ d = (k_4 \otimes k_8 \otimes k_{12} \otimes k_{16} + k_{20} \otimes k_{24} \otimes k_{28} \otimes k_{32})/256, \end{cases} \tag{15}$$

$$\begin{cases} x_0 = ((\sum_{i=1}^{8} k_i)/256) \bmod a + t_1, \\ y_0 = ((\sum_{i=9}^{16} k_i)/256) \bmod b + t_2, \\ z_0 = ((\sum_{i=17}^{24} k_i)/256) \bmod c + t_3, \\ w_0 = ((\sum_{i=25}^{32} k_i)/256) \bmod d + t_4, \end{cases} \tag{16}$$

where $\oplus$ indicates the XOR operation, $\sum$ indicates the summation operation, mod( ) indicates the remainder operation.

Step 2 The initial values x_0, y_0, z_0, w_0 are brought into the 4D MHS and iterated $m \times n + N_0$ times. To avoid transient effects, the first $N_0 (N_0 = 1000)$ values are discarded and four hyperchaotic sequences X, Y, Z, W of size $1 \times mn$ are obtained, which are arranged in ascending order to obtain four index sequences S_X, S_Y, S_Z, S_W. The index sequences S_X, S_Y, S_Z, S_W are used to permute the sequences X, Y, Z, W to obtain sequences P_1, Q_1, P_2, Q_2 in turn.

Step 3 The sequences of system control parameters u_i, v_i ($i = 1, 2, \ldots, m \times n$) and the initial system value s_0 of the 1D CCS are generated from the sequences P_1, Q_1, P_2, Q_2 as shown below:

$$\begin{cases} u_i = (P_1/Q_1) \bmod 10, \\ v_i = (P_2/Q_2) \bmod 20, \end{cases} \tag{17}$$

$$s_0 = \sum_{i=1}^{m\times n} \{[(P_1 + P_2)/(Q_1 + Q_2) \bmod 1]\}/(m \times n). \tag{18}$$

Given that the control parameters $u \in (0, 10]$, $v \in [8, 20]$, the sequences u_i and v_i are quantized into the above intervals.

Step 4 The sequence u_i, v_i and the initial value s_0 are brought into the 1D CCS and iterated $m \times n$ times to obtain the chaotic sequence s_i ($i = 1, 2, \ldots, m \times n$). The sequence s_i is rearranged into a matrix Φ of size $m \times n$, as shown in Eq. (19). The matrix $\Phi_{m\times n}$ is the DCMM. The system control parameters u_i, v_i change in each iteration, which enhances the randomness of DCMM.

$$\Phi = \text{reshape}(s, m, n), \tag{19}$$

where reshape (s, m, n) indicates the rearrangement of the sequence s into a matrix with a size of $m \times n$.

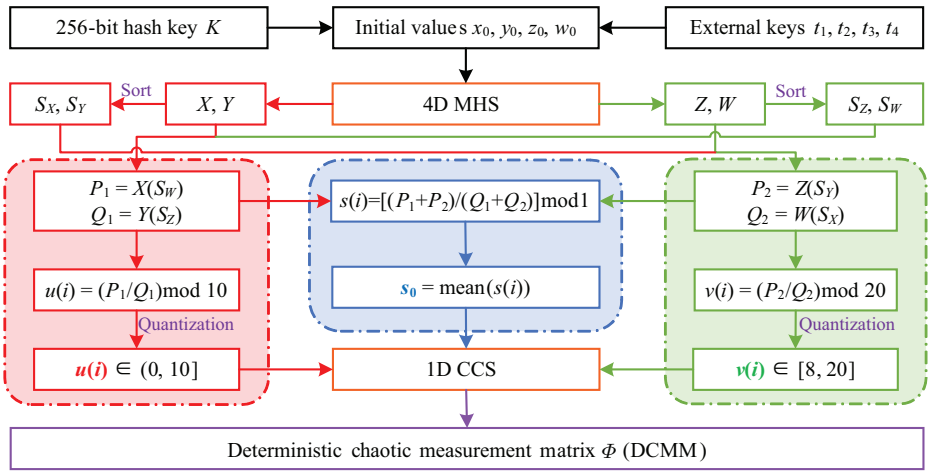


Fig. 5. The construction process of DCMM.

3.3. Confusion method determined by plaintext information, hyperchaotic sequences, and Latin square matrix (CMDPHL)

In this section, a confusion method based on the plaintext information, hyperchaotic sequences, and Latin square matrix is proposed to determine the confusion matrix. First, the row key k_r and the column key k_c are generated from the plaintext image $P_{M\times N}$. Then, the Latin square matrix $L_{m\times n}$ is generated from the hyperchaotic sequences X, Y according to the method in Subsection 2.4. Finally, the row scrambling matrix $R_{m\times n}$ and the column scrambling matrix $C_{m\times n}$ are generated from k_r, k_c, $L_{m\times n}$, and hyperchaotic sequences Z, W. The matrix $P1_{m\times n}$ is permuted by $R_{m\times n}$ and $C_{m\times n}$. The specific steps are shown as follows.

Step 1 Calculate the information entropy H_p of the plaintext image $P_{M\times N}$. Seek the maximum value $r_{\max}$ and the minimum value $r_{\min}$ for the sum of the row pixel value of $P_{M\times N}$. Similarly, find the maximum value $c_{\max}$ and the minimum value $c_{\min}$ for the sum of column pixel values. Calculate the row key k_r and the column key k_c from

$$k_r = \text{floor}(H_P \times (r_{\max}/r_{\min}) \times 10^{14} \bmod m), \tag{20}$$

$$k_c = \text{floor}(H_P \times (c_{\max}/c_{\min}) \times 10^{14} \bmod n), \tag{21}$$

where floor(x) indicates taking the largest integer not greater than x.

Step 2 The four hyperchaotic sequences X, Y, Z, W of size $1 \times mn$ generated in Subsection 3.2 are quantized in the interval [0, 255].

Step 3 The sequences X and Y are used to generate the Latin square matrix $L_{m\times n}$ according to the method in Subsection 2.4.

Step 4 The row confusion matrix $R_{m\times n}$ is generated by the row key k_r, the Latin square matrix $L_{m\times n}$ and the sequence Z together, as shown below:

$$R(i,j) = (k_r + L(i,j) + Z(i,j)) \bmod m + 1. \tag{22}$$

Step 5 The column confusion matrix $C_{m\times n}$ is generated by the column key k_c, the Latin square matrix $L_{m\times n}$ and the sequence W together, as shown below:

$$C(i,j) = (k_c + L(i,j) + W(i,j)) \bmod n + 1. \tag{23}$$

Step 6 Each row of matrix $P1_{m\times n}$ is scrambled by matrix $R_{m\times n}$ and each column of matrix $P1_{m\times n}$ is scrambled by matrix $C_{m\times n}$ to obtain the scrambled matrix $P2_{m\times n}$.

The pseudo-code of the scrambling algorithm is shown in Algorithm 2, and the procedure of the scrambling method is illustrated in Fig. 6.

Algorithm 2: The confusion method based on CMDPHL

Input: matrix $P1$; size of $P1$: m, n; hyperchaotic sequences X, Y, Z, W; the row key k_r; the column key k_c

Output: the scrambled matrix $P2_{m\times n}$

$X = \text{floor}\,(\text{mod}(X \times 10^{14}, m))$; $Y = \text{floor}\,(\text{mod}(Y \times 10^{14}, m))$; $Z = \text{floor}\,(\text{mod}(Z \times 10^{14}, m))$; $W = \text{floor}\,(\text{mod}(W \times 10^{14}, m))$;
$P = P1$;
$[\sim, S_X] = \text{sort}(X)$; $[\sim, S_Y] = \text{sort}(Y)$;
for $i = 1 : m$ do
    $L(i,:) = \text{circshift}\,(S_X, S_Y(i))$;
end for
for $i = 1 : m$ do
    for $j = 1 : n$ do
        $R(i,j) = \text{mod}(k_r + L(i,j) + Z(i), m) + 1$;
        $P(i,j) = P(i, R(i,j))$;
    end for
end for
for $i = 1 : m$ do
    for $j = 1 : n$ do
        $C(i,j) = \text{mod}(k_c + L(i,j) + W(i), n) + 1$;
        $P(i,j) = P(C(i,j), j)$;
    end for
end for
$P2 = P$;

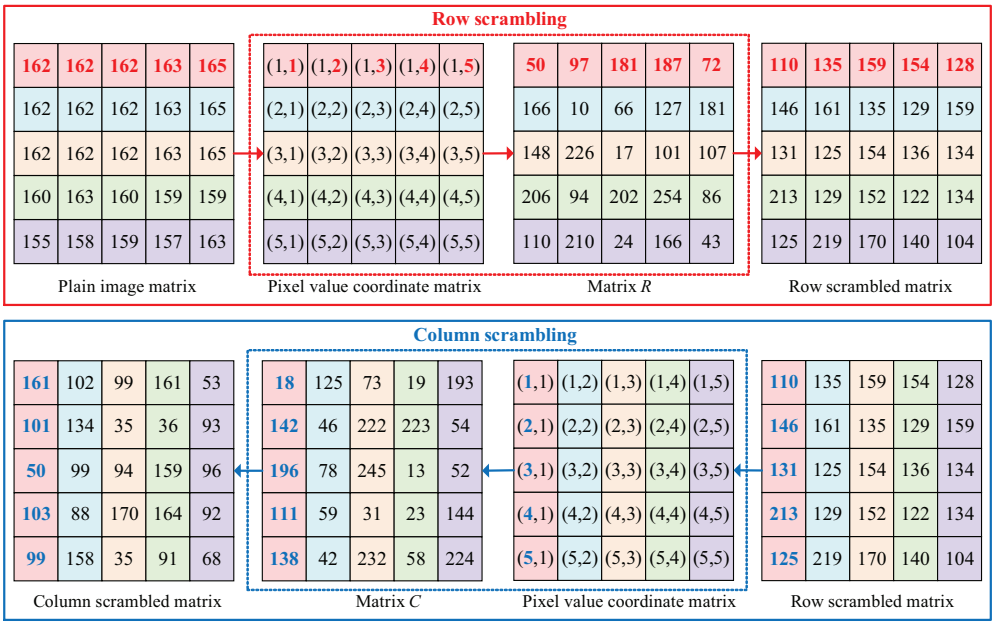


Fig. 6. The process example of the confusion method based on CMDPHL.

3.4. Diffusion method determined by plaintext information and hyperchaotic sequences (DMDPH)

In this section, a diffusion method based on plaintext information and hyperchaotic sequence is proposed to determine the diffusion matrix. The pseudo-code of the diffusion algorithm is shown in Algorithm 3. The specific steps of the diffusion algorithm are as follows:

Step 1 The quantized sequences X and Y of size $1\times mn$ are rearranged into a matrix A of size $m\times N$, and similarly, the matrix $B_{m\times N}$ is obtained from the sequences Z and W.

Step 2 The row key k_r and the matrix $A_{m\times N}$ jointly determine the forward diffusion of the pixel values of the matrix $P1_{m\times N}$, and the matrix $H_{m\times N}$ is obtained, as shown below:

$$H(i,j)=\begin{cases} P1(i,j)\oplus k_r\oplus A(i,j), & i=1,j=1,\\ P1(i,j)\oplus P1(i-1,N)\oplus A(i,j), & i\geq 2,j=1,\\ P1(i,j)\oplus P1(i,j-1)\oplus A(i,j), & j\neq 1. \end{cases} \tag{24}$$

Step 3 The column key k_c and matrix $B_{m\times N}$ jointly determine the inverse diffusion of the pixel values of the matrix $H_{m\times N}$, and the matrix $V_{m\times N}$ is obtained, which is the matrix $P2_{m\times N}$ after diffusion, as shown below:

$$V(i,j)=\begin{cases} H(i,j)\oplus k_c\oplus B(i,j), & i=m,j=N,\\ H(i,j)\oplus H(1,j+1)\oplus B(i,j), & i=m,j<N,\\ H(i,j)\oplus H(i+1,j)\oplus B(i,j), & i\neq m. \end{cases} \tag{25}$$

Algorithm 3: The diffusion method based on DMDPH

Input: matrix $P1$; size of $P1$: m, N; hyperchaotic sequences X, Y, Z, W; the row key k_r; the column key k_c
Output: the diffused matrix $P2_{m\times N}$

```
A = [X,Y]; A = reshape(A,m,N);
B = [Z,W]; B = reshape(B,m,N);
P = P1;
for i = 1 : m do
    for j = 1 : N do
        if (i = 1) & (j = 1)
            H(i,j) = bitxor (bitxor (P(i,j),k_r),A(i,j));
        else if (i ≥ 2) & (j = 1)
            H(i,j) = bitxor (bitxor (P(i,j),P(i−1,N)),A(i,j));
        else
            H(i,j) = bitxor (bitxor (P(i,j),P(i,j−1)),A(i,j));
        end if
    end for
end for
for j = N : −1 : 1 do
    for i = m : −1 : 1 do
        if (i = m) & (j = N)
            V(i,j) = bitxor (bitxor (H(i,j),k_c),B(i,j));
        else if (i = m) & (j < N)
            V(i,j) = bitxor (bitxor (H(i,j),H(1,j+1)),B(i,j));
        else
            V(i,j) = bitxor (bitxor (H(i,j),H(i+1,j)),B(i,j));
        end if
    end for
end for
P2 = V;
```

3.5. Encryption process

The flow chart of the proposed image encryption algorithm in this paper is shown in Fig. 7. It consists of three main stages: CS-based encryption stage, optical encryption stage based on FrFT, and confusion and diffusion stage.

In the CS-based encryption process, the measurement matrix used is the DCMM proposed in Subsection 3.2; the confusion method is CMDPHL elaborated in Subsection 3.3; and the diffusion method is DMDPH proposed in Subsection 3.4. The specific steps are shown below.

Step 1 Input the plaintext image $P_{M\times N}$ and set the compression rate as CR $=0.5$. Calculate the 256-bit hash key K of $P_{M\times N}$ by the SHA-256 function, and generate the initial values x_0, y_0, z_0, w_0 of 4D MHS by use of K and the external keys t_i. Obtain four hyperchaotic sequences X, Y, Z, W of size $1\times mn$ from 4D MHS. Construct the $\Phi_{m\times n}$ from the sequences X, Y, Z, W, and 1D CCS. The specific steps are shown in Subsection 3.2.

Step 2 The discrete wavelet transform (DWT) is performed on the plaintext image $P_{M\times N}$ to obtain the approximation coefficients matrix $cA_{m\times n}$ and detail coefficients matrices $cH_{m\times n}$, $cV_{m\times n}$, $cD_{m\times n}$. The "haar wavelet" is chosen, and the detailed process is shown below:

$$[cA,cH,cV,cD]=\mathrm{dwt2}(P,'\mathrm{haar}'). \tag{26}$$

Step 3 Scramble the matrix $cA_{m\times n}$ by the sequences X, Y, Z, W to obtain the matrix $cA1_{m\times n}$, as shown in Eq. (27). The matrix $\Phi_{m\times n}$ is divided into the matrices $\Phi_{cH(m/2\times n)}$, $\Phi_{cV(m/4\times n)}$, $\Phi_{cD(m/4\times n)}$, and the matrices $cH_{m\times n}$, $cV_{m\times n}$, $cD_{m\times n}$ are compressed and measured in turn to obtain the

matrices $cH1_{(m/2\times n)}$, $cV1_{(m/4\times n)}$, $cD1_{(m/4\times n)}$, as shown in Eq. (28), where the matrix ψ is the sparse basis of DWT and the matrix ψ^{T} is the transpose matrix of matrix ψ.

$$\begin{cases} [\sim, S] = \mathrm{sort}((X+Y+Z+W)/4), \\ cA = \mathrm{reshape}(cA, 1, m\times n), \\ cA1(S) = cA, \\ cA1 = \mathrm{reshape}(cA1, m, n), \end{cases} \tag{27}$$

$$\begin{cases} cH1 = \Phi_{cH} \times (\psi \times cH \times \psi^{\mathrm{T}}), \\ cV1 = \Phi_{cV} \times (\psi \times cV \times \psi^{\mathrm{T}}), \\ cD1 = \Phi_{cD} \times (\psi \times cD \times \psi^{\mathrm{T}}). \end{cases} \tag{28}$$

Step 4 The matrices $cH1_{(m/2\times n)}$, $cV1_{(m/4\times n)}$, $cD1_{(m/4\times n)}$ are combined into a matrix $cHVD1_{m\times n}$. The matrix $cA1_{m\times n}$ is used as the real part and the matrix $cHVD1_{m\times n}$ is used as the imaginary part, and the above two are combined into a complex matrix $cAHVD_{m\times n}$, as shown in Eq. (29). The random phase masks $M1_{m\times n}$, $M2_{m\times n}$ are generated by the sequences X, Y, Z, W, as shown in Eq. (30). The DRPE encryption is manipulated on the complex matrix $cAHVD_{m\times n}$ to obtain the complex matrix $cAHVD1_{m\times n}$, and the optical transform is the fractional Fourier transform, as shown in Eq. (31):

$$\begin{cases} cHVD1 = [cH1; cV1; cD1], \\ cAHVD = cA1 + \mathrm{i} \times cHVD1, \end{cases} \tag{29}$$

$$\begin{cases} S1 = (X+Y)/2; & S2 = (Z+W)/2, \\ S1 = \mathrm{reshape}(S1, m, n); & S2 = \mathrm{reshape}(S2, m, n), \\ M1 = \exp(\mathrm{i} \times 2\pi \times S1); & M2 = \exp(\mathrm{i} \times 2\pi \times S2), \end{cases} \tag{30}$$

$$cAHVD1 = F^{-t_3}F^{-t_4}[F^{t_3}F^{t_4}(cAHVD \times M1) \times M2], \tag{31}$$

where $F^{t_3}F^{t_4}(\ )$ is the fractional Fourier transform with order t_3 in the x direction and order t_4 in the y direction, t_i is the external key.

Step 5 The real part of the complex matrix $cAHVD1_{m\times n}$ is taken to obtain the matrix $cA2_{m\times n}$; the imaginary part is taken to obtain the matrix $cHVD2_{m\times n}$. The element values of the matrices $cA2_{m\times n}$, $cHVD2_{m\times n}$ are quantized into the interval [0, 255] to obtain the matrices $cA3_{m\times n}$, $cHVD3_{m\times n}$, as shown below:

$$\begin{cases} cA2 = \mathrm{real}(cAHVD1), \\ cHVD2 = \mathrm{imag}(cAHVD1), \\ cA3 = \mathrm{round}[255 \times (cA2 - cA2_{\min})/(cA2_{\max} - cA2_{\min})], \\ cHVD3 = \mathrm{round}[255 \times (cHVD2 - cHVD2_{\min})/(cHVD2_{\max} - cHVD2_{\min})], \end{cases} \tag{32}$$

where $A_{\max}$ indicates the maximum element in matrix A, $A_{\min}$ indicates the minimum element in matrix A, real (A) indicates taking the real part of the matrix A, imag (A) indicates taking the imaginary part of the matrix A, round (x) indicates the integer rounded to the nearest x.

Step 6 The matrices $cA3_{m\times n}$ and $cHVD3_{m\times n}$ are scrambled by CMDPHL to obtain matrices $cA4_{m\times n}$ and $cHVD4_{m\times n}$. The specific steps are shown in Subsection 3.3. Combine the matrices $cA4_{m\times n}$ and $cHVD4_{m\times n}$ into a matrix $cAHVD2_{m\times N}$, as shown below:

$$cAHVD2 = [cA4; cHVD4]. \tag{33}$$

Step 7 The matrix $cAHVD2_{m\times N}$ is diffused by DMDPH to obtain the matrix $cAHVD3_{m\times N}$, which is the ciphertext image $C_{m\times N}$. The specific steps are shown in Subsection 3.4.

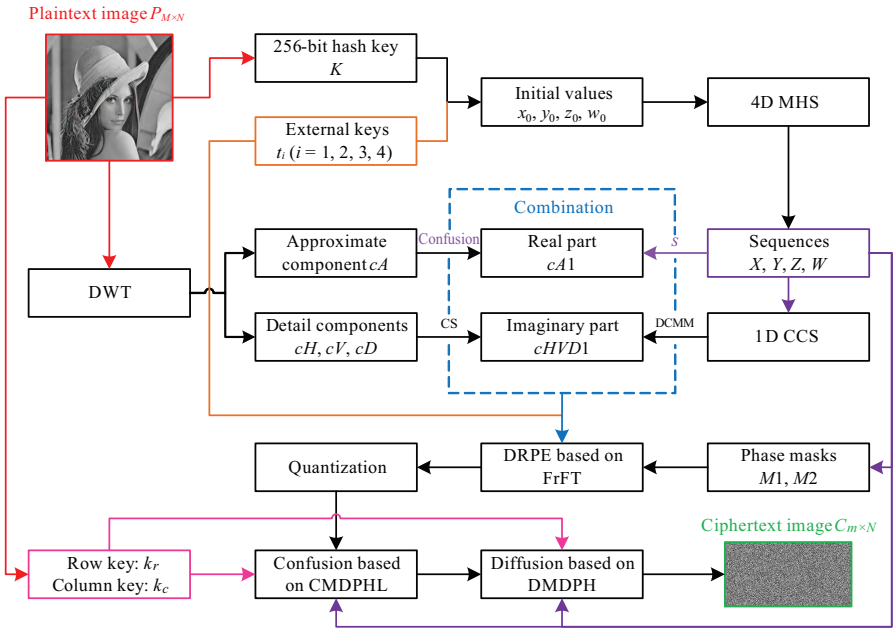


Fig. 7. The flow chart of the proposed image encryption algorithm.

3.6. Decryption process

The decryption algorithm is the inverse process of the encryption algorithm, and the flow chart of the decryption algorithm is shown in Fig. 8. The 256-bit hash key K, the external keys t_i ($i = 1, 2, 3, 4$) and the intermediate keys $cA2_{\min}$, $cA2_{\max}$, $cVHD2_{\min}$, $cVHD2_{\max}$, k_r and k_c need to be transmitted to the receiver for decryption. The OMP algorithm is chosen for the image reconstruction method. The specific steps of the decryption algorithm are not described here.

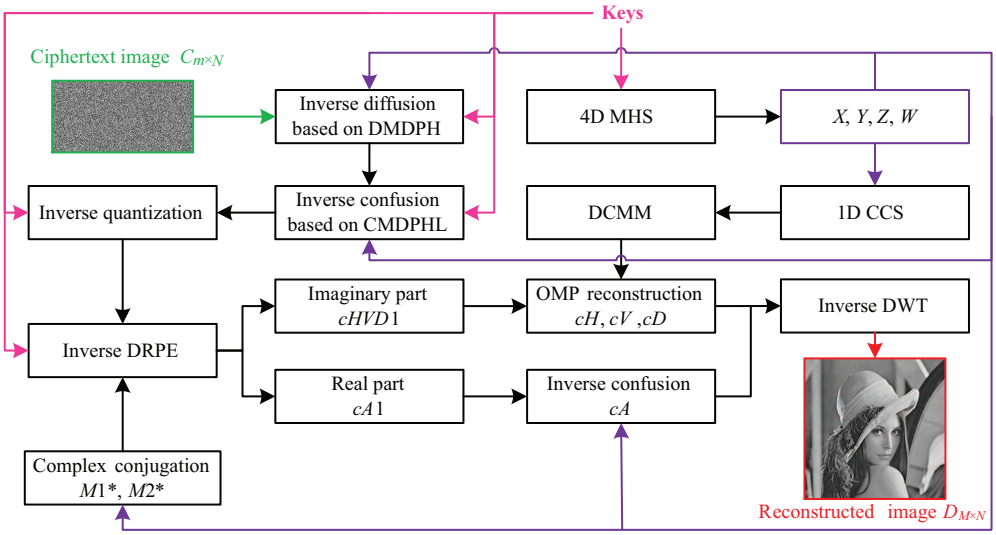


Fig. 8. The flow chart of the proposed image decryption algorithm.

4. Experimental simulation and security analysis

In this section, the proposed image encryption scheme is simulated, and the simulation results and the security performance of the algorithm are analyzed. The simulation environment used is in a personal computer with an Intel(R) Core (TM) i5-6300HQ 2.30 GHz CPU and 8 GB memory, the operating system is Microsoft Windows 11, 64-bit, and the software is Matlab2016b.

The fixed parameters used in the experimental simulations are the external keys $t_1 = 0.3$, $t_2 = 0.4$, $t_3 = 0.5$, $t_4 = 0.6$, the system control parameters $a = 35$, $b = 6$, $c = 20$, $k = 6$, $\alpha = 0.1$, $\beta = 0.02$ of 4D MHS, the discard value $N_0 = 1000$ of the hyperchaotic sequences, and the compression rate CR $= 0.5$.

The test objects of the experimental simulation are five grayscale images of size 512×512 "Baboon", "Cameraman", "Lake", "Lena", and "Peppers".

4.1. Encryption and decryption results

The proposed encryption scheme is used to test the encryption and decryption of the above five images, and the experimental results are shown in Fig. 9.

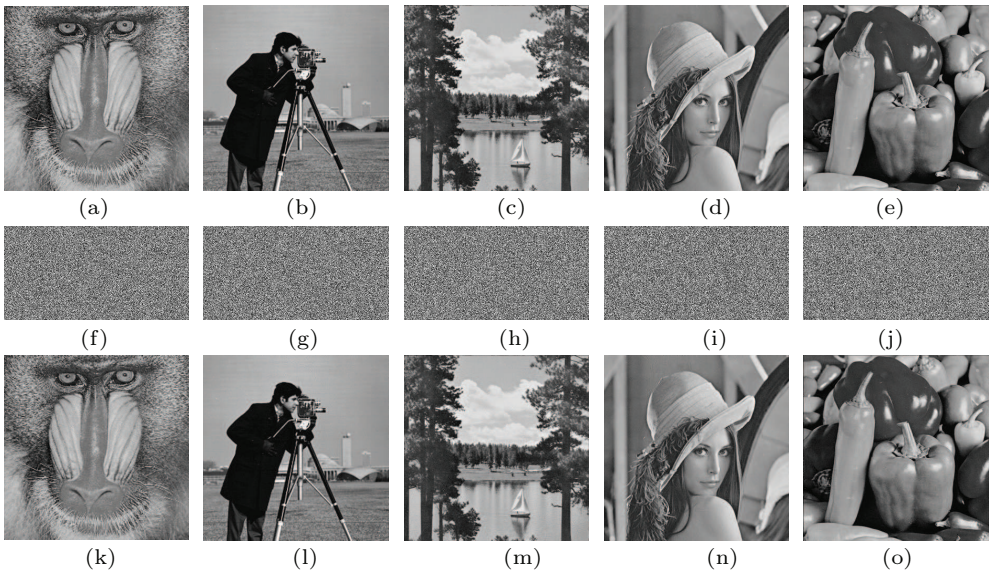


Fig. 9. Different grayscale images to test encryption and decryption effect: (a)–(e) plaintext images, "Baboon", "Cameraman", "Lake", "Lena", and "Peppers"; (f)–(j) corresponding ciphertext images; (k)–(o) corresponding reconstructed images.

The size of the plaintext image P and the reconstructed image D are 512×512, and the size of the ciphertext image C is 256×512. The ciphertext image is a noise-like image, from which no valid information related to the plaintext image can be obtained, and the quality of the reconstructed image after decryption is good, which indicates that the proposed scheme has good encryption and decryption effects, and the size of the plaintext image is compressed, which is more convenient for image transmission and storage.

4.2. The analysis of reconstructed image quality

In order to quantitatively evaluate the quality of the reconstructed images, peak signal-to-noise ratio (PSNR) and mean structural similarity (MSSIM) are used to evaluate the visual quality of the images.

4.2.1. Peak signal-to-noise ratio (PSNR)

The PSNR is often used to evaluate the quality of the decrypted image, which is calculated according to

$$\mathrm{PSNR} = 10\log \frac{255^2}{(1/MN)\sum_{i=1}^{M}\sum_{j=1}^{N}[X(i,j)-Y(i,j)]^2}, \quad (34)$$

where $X(i,j)$ and $Y(i,j)$ indicate the two images to be tested, M and N are the number of rows and columns of the images.

The PSNR values between the reconstructed images and the plaintext images are shown in Table 2. The PSNR values of the five images are between 30.1355 dB and 36.4641 dB with an average value of 34.1084 dB, and the PSNR value of each image is higher than 30 dB, which indicates that the proposed algorithm is effective and the decryption algorithm has a good image reconstruction effect.

Table 2. The PSNR values between the reconstructed images and the plaintext images.

Image	Baboon	Cameraman	Lake	Lena	Peppers	Average
PSNR (dB)	30.1355	36.4641	32.3941	35.6329	35.9155	34.1084

4.2.2. Mean structural similarity (MSSIM)

The MSSIM is a measure of the similarity of two images, the closer the MSSIM value is to 1, the smaller the difference between the two images, the more similar the two images are, and the smaller the image distortion. The MSSIM is calculated according to

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(2\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (35)$$

$$\mathrm{MSSIM}(X,Y) = \frac{1}{N}\sum_{i=1}^{N}\mathrm{SSIM}(X_i, Y_i), \quad (36)$$

where μ_x and μ_y are the mean values of x and y, respectively. σ_x^2 and σ_y^2 are the variances of x and y, respectively. σ_{xy} is the covariance of x and y. X_i and Y_i indicate the i-th block of the plaintext image X and the reconstructed image Y, respectively, and N is the number of blocks. As usual, we set the $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$, $k_1 = 0.01$, $k_2 = 0.03$, $L = 255$.

The MSSIM values between the reconstructed images and the plaintext images are shown in Table 3. The MSSIM values of the five images are between 0.9913 and 0.9974, and the average value is 0.9937. The MSSIM value of each image is above 0.99 and close to 1, which indicates that there is a high similarity between the reconstructed image and the plaintext image, the quality of the reconstructed image is good, and the proposed algorithm has a good image fidelity.

Table 3. The MSSIM values between the reconstructed images and the plaintext images.

Image	Baboon	Cameraman	Lake	Lena	Peppers	Average
MSSIM	0.9974	0.9913	0.9938	0.9941	0.9921	0.9937

4.3. Statistical analysis

4.3.1. Histogram analysis

The histogram reflects the distribution of pixel values in the image. The plaintext image has its own characteristics, and its histogram is not uniformly distributed, and the histogram obtained by the secure image encryption algorithm should be uniform and flat, from which no valid information about the plaintext image can be obtained.

We plotted the histograms of the plaintext image, ciphertext image, and reconstructed image of the above five images in Fig. 10, respectively. It can be seen that the histograms of each ciphertext image are uniform and flat, and the histograms of the ciphertext images are similar and flat regardless of which plaintext image is encrypted. The histograms of the reconstructed images and the histograms of the plaintext images have obviously similar characteristics.

4.3.2. Correlation analysis between adjacent pixel

The adjacent pixels of the plaintext image are generally strongly correlated in all directions, and their correlation coefficients are close to 1. However, the adjacent pixels of the ciphertext image obtained by the secure image encryption algorithm should be weakly correlated in all directions, and their correlation coefficients are close to 0. The correlation coefficient ρ_{xy} is calculated according to

$$\rho_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2} \times \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}, \quad (37)$$

where N is the number of pixels of the image, x_i and y_i are the pixel values of the adjacent pixels of the image, and $\bar{x}$ and $\bar{y}$ are the average values of two adjacent pixels.

The correlation coefficients are calculated by randomly selecting 5000 pairs of adjacent pixels from plaintext images and ciphertext images in horizontal, vertical, and diagonal directions. The results are shown in Tables 4 and 5, and the distributions of adjacent pixels in each direction are shown in Figs. 11 and 12.

As can be seen in Fig. 11, the adjacent pixels of the plaintext image in three directions are concentrated on the diagonal. From Fig. 12, it can be seen that the adjacent pixels of the corresponding ciphertext image in the three directions are spread over the whole coordinate space. From Tables 4 and 5, it can be seen that the correlation coefficient of the plaintext image in each direction is close to 1, and the correlation coefficient of its corresponding ciphertext image in each direction is close to 0. This indicates that the proposed algorithm can effectively reduce the strong correlation between the neighboring pixels of the plaintext image and can effectively resist the statistical analysis attack.

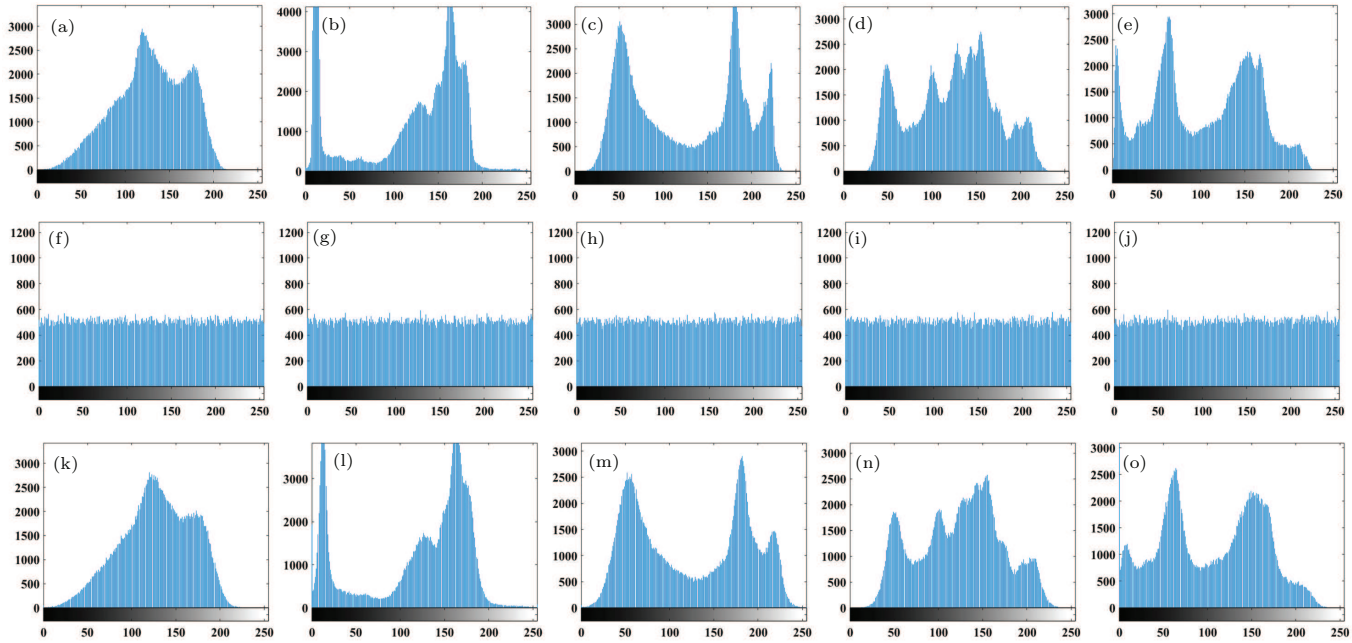


Fig. 10. Histograms plotting results: (a)–(e) histograms of plaintext images, "Baboon", "Cameraman", "Lake", "Lena", and "Peppers"; (f)–(j) histograms of ciphertext images; (k)–(o) histograms of reconstructed images.

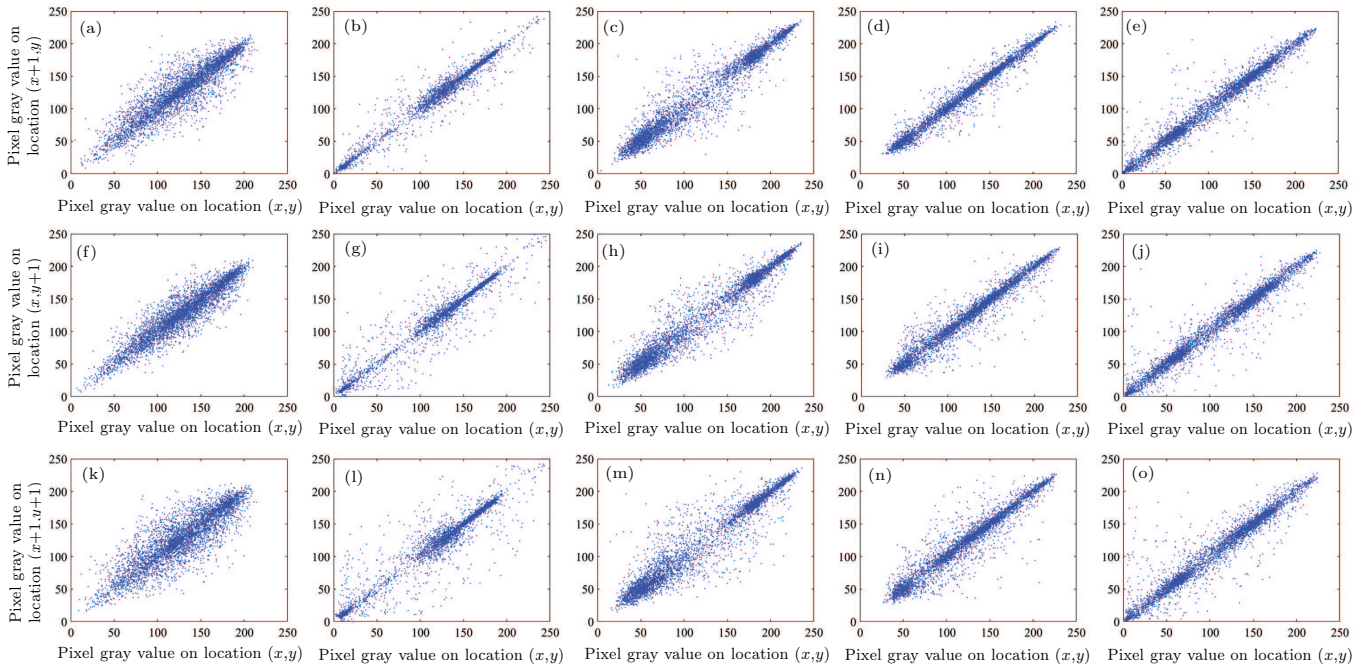


Fig. 11. Adjacent pixels distribution of plaintext images: (a)–(e) horizontal direction, "Baboon", "Cameraman", "Lake", "Lena", and "Peppers"; (f)–(j) vertical direction; (k)–(o) diagonal direction.

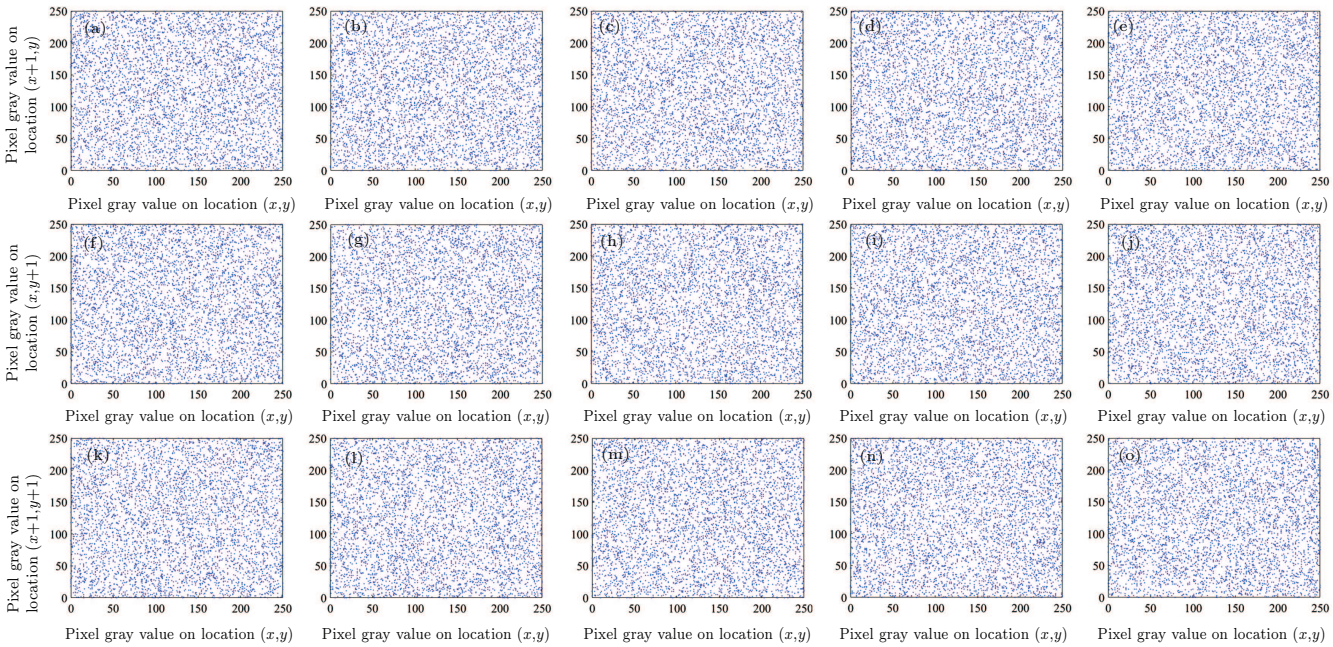


Fig. 12. Adjacent pixels distribution of ciphertext images: (a)–(e) horizontal direction, "Baboon", "Cameraman", "Lake", "Lena", and "Peppers"; (f)–(j) vertical direction; (k)–(o) diagonal direction.

Table 4. Correlation coefficients of adjacent pixels in three directions of the plaintext images.

Direction	Plaintext image					
	Baboon	Cameraman	Lake	Lena	Peppers	Average
Horizontal	0.9154	0.9918	0.9726	0.9859	0.9825	0.9696
Vertical	0.9362	0.9827	0.9729	0.9748	0.9785	0.9690
Diagonal	0.8673	0.9749	0.9577	0.9616	0.9668	0.9457

Table 5. Correlation coefficients of adjacent pixels in three directions of the ciphertext images.

Direction	Ciphertext image					
	Baboon	Cameraman	Lake	Lena	Peppers	Average
Horizontal	–0.0036	0.00073	0.0011	–0.0019	0.00059	–0.00062
Vertical	–0.0033	0.0084	0.0030	0.0051	–0.0098	0.00068
Diagonal	–0.0039	0.0083	0.0059	–0.0078	0.0071	0.00192

4.3.3. Information entropy analysi

The information entropy of an image can represent the randomness of image information, and can be used to evaluate the randomness of an image before and after encryption. For an 8-bit grayscale image, the ideal value of information entropy is 8, which is calculated according to

$$H(s) = -\sum_{i=0}^{2^k-1} v(s_i)\log_2 v(s_i), \tag{38}$$

where $v(s_i)$ indicates the probability of occurrence of s_i and k is the total number of bits of s_i.

The information entropy of the plaintext images, ciphertext images and reconstructed images are tested separately, and the results are shown in Table 6. The information entropy of the five ciphertext images are between 7.9983 and 7.9987, and the average value is 7.99852, and the information entropy of each ciphertext image is above 7.998, which is close to the ideal value 8. This indicates that the ciphertext images have high randomness and the proposed algorithm has better security.

Table 6. The information entropy of different plaintext images, ciphertext images, and reconstructed images.

Image type	Information entropy					
	Baboon	Cameraman	Lake	Lena	Peppers	Average
Plaintext image	7.2925	7.0480	7.4570	7.4451	7.5715	7.36282
Ciphertext image	7.9986	7.9986	7.9987	7.9984	7.9983	7.99852
reconstructed image	7.3169	7.1924	7.5890	7.4909	7.6202	7.44188

4.4. Key space analysis

The keys of the proposed algorithm include the 256-bit hash key K, the external keys t_i ($i = 1, 2, 3, 4$) and the intermediate keys $cA2_{\min}$, $cA2_{\max}$, $cVHD2_{\min}$, $cVHD2_{\max}$, k_r, k_c. If the computing accuracy of the computer is 10^{-14}, the key space of the proposed algorithm is about $(10^{14})^4 \times (10^{14})^6 = 10^{140} > 2^{400} \gg 2^{100}$. Therefore, the key space of the proposed algorithm is large enough to effectively resist exhaustive attacks.

4.5. Key sensitivity analysis

The sensitivity of the algorithm to the keys is an important index to evaluate the security of the algorithm, which can be evaluated from two aspects: first, from the encryption point of view, a small change in the encryption keys can lead to a completely different ciphertext image; second, from the decryption point of view, the plaintext image cannot be accurately reconstructed even if there is only a slight difference between the decryption keys and the encryption keys. The number of pixels change rate (NPCR) and the unified average changing intensity (UACI) are calculated according to

$$\mathrm{NPCR} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j)}{m \times n} \times 100\%, \tag{39}$$

$$D(i,j) = \begin{cases} 0, & D_1(i,j) = D_2(i,j), \\ 1, & D_1(i,j) \neq D_2(i,j), \end{cases} \tag{40}$$

$$\mathrm{UACI} = \frac{1}{m \times n}\left[\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{|D_1(i,j) - D_2(i,j)|}{255}\right] \times 100\%, \tag{41}$$

where m and n are the sizes of the image, $D_1(i,j)$ indicates the pixel value of the correct ciphertext image at coordinate (i,j), and $D_2(i,j)$ indicates the pixel value of the incorrect ciphertext image at coordinate (i,j).

From the encryption point of view, the sensitivity of the algorithm to the encryption key is tested for the above five images. The external keys t_1, t_2, t_3, t_4 are changed in turn by the amount of $\Delta = 10^{-14}$ and the other three keys are kept unchanged to obtain the corresponding ciphertext images, respectively. The values of NPCR between them and the correct ciphertext image are calculated separately and the results are shown in Table 7. It can be seen that the average value of NPCR between each incorrect ciphertext image and the correct ciphertext image is 99.6090%, which is close to the ideal value of 99.6094%. Therefore, when the encryption key is changed slightly, the correct ciphertext image cannot be obtained.

Table 7. The NPCR values of changing the encryption keys t_1, t_2, t_3, and t_4 in turn.

Keys	NPCR (%)					
	Baboon	Cameraman	Lake	Lena	Peppers	Average
$t_1 + 10^{-14}$	99.62	99.60	99.60	99.60	99.65	99.614
$t_2 + 10^{-14}$	99.61	99.60	99.63	99.60	99.59	99.606
$t_3 + 10^{-14}$	99.62	99.64	99.61	99.59	99.62	99.616
$t_4 + 10^{-14}$	99.62	99.62	99.60	99.56	99.60	99.600

Taking the "Lena" image as an example, we change the encryption keys t_1, t_2, t_3, t_4 in turn, and the corresponding ciphertext image and its difference with the correct ciphertext image are shown in Fig. 13. It can be seen that there is a big difference between the correct ciphertext image and the ciphertext image after the small change of encryption key.

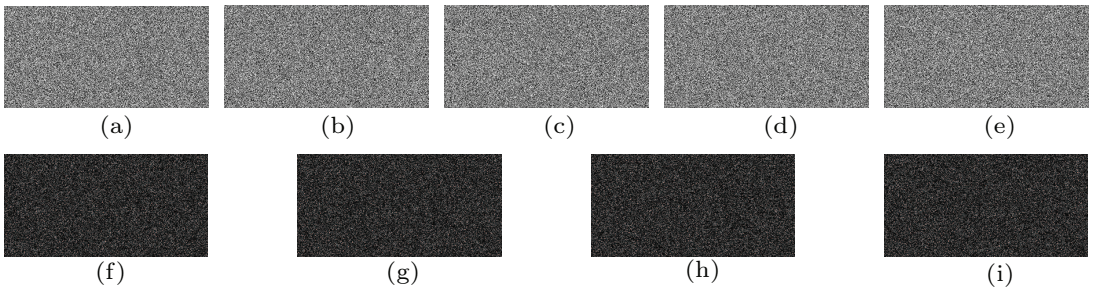
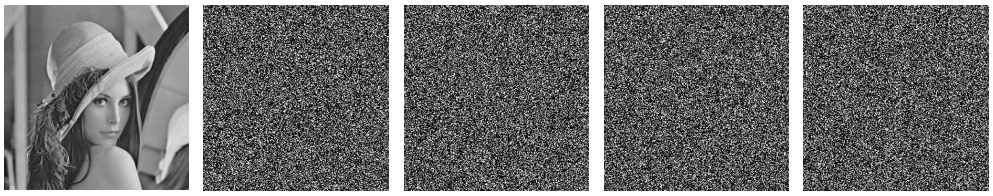


Fig. 13. Ciphertext images of the "Lena" image: (a) ciphertext image under correct encryption keys; (b)–(e) ciphertext images after encryption keys t_1, t_2, t_3, t_4 are changed in turn; (f)–(i) difference images of panels (b)–(e) and (a).

From the decryption point of view, the algorithm is tested for the sensitivity of the decryption key. The ciphertext images are decrypted using the keys with minor changes respectively, where the keys t_1, t_2, t_3, t_4 are changed by 10^{-16}, 10^{-16}, 10^{-15}, 10^{-15}. The decryption results are shown in Fig. 14. It can be seen that when the decryption key used is changed slightly, the ciphertext image cannot be decrypted correctly, and the plaintext image is reconstructed. Therefore, the proposed algorithm has high sensitivity to the key in both the encryption and decryption stages.



(a) (b) (c) (d) (e)

Fig. 14. Reconstructed images of the "Lena" image: (a) reconstructed image under correct decryption keys: (b)–(e) reconstructed images after decryption keys t_1, t_2, t_3, t_4 are changed in turn.

4.6. Differential attack analysis

The ability to resist differential attacks is an important metric to evaluate the encryption effectiveness of the algorithm, and two numerical metrics, NPCR and UACI, are usually used to evaluate the degree of difference between two images.

Table 8. NPCR values of changing the pixels of different positions.

Image	NPCR (%)			
	(1, 1)	(256, 256)	(512, 512)	Average
Baboon	99.60	99.58	99.59	99.5900
Cameraman	99.62	99.61	99.61	99.6133
Lake	99.58	99.61	99.59	99.5933
Lena	99.63	99.61	99.63	99.6233
Peppers	99.62	99.59	99.59	99.6000

Table 9. UACI values of changing the pixels of different positions.

Image	UACI (%)			
	(1, 1)	(256, 256)	(512, 512)	Average
Baboon	33.46	33.42	33.46	33.4467
Cameraman	33.43	33.38	33.46	33.4233
Lake	33.50	33.51	33.46	33.4900
Lena	33.57	33.55	33.60	33.5733
Peppers	33.51	33.50	33.47	33.4933

The pixel values at coordinates (1,1), (256,256), and (512,512) of the above five plaintext images are incremented by 1 to obtain their corresponding ciphertext images. The NPCR values and UACI values between these ciphertext images and the correct ciphertext images are calculated and the results are shown in Tables 8 and 9. It can be seen that the average value of NPCR is 99.6040%, which is close to the ideal value of 99.6094%, and the average value of UACI is 33.4853%, which is close to the ideal value of 33.4635%. This indicates that the proposed algorithm has a high sensitivity to plaintext image changes and can effectively resist differential attacks.

4.7. Robustness analysis

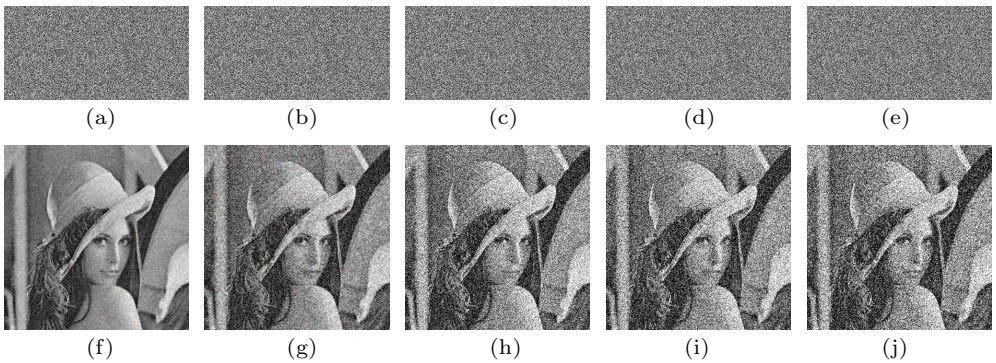
Ciphertext images are vulnerable to noise and data loss attacks during transmission, and the type and intensity of noise and the degree of data loss affect the quality of reconstructed images. The proposed algorithm is tested for noise attacks and clipping attacks to analyze its robustness.

4.7.1. Noise attack analysis

Taking the "Lena" image as an example, we add different levels of Gaussian noise (GN), Speckle noise (SN) and Salt & Pepper noise (SPN) to the ciphertext image to obtain the corresponding ciphertext images. These ciphertext images are decrypted, and the results are shown in Figs. 15–17. The PSNR values between the corresponding decrypted images and the plaintext images are calculated, and the results are shown in Table 10.

It can be seen that GN and SN affect the quality of the reconstructed images, while SPN has almost no effect on the quality of the reconstructed images. And the quality of the reconstructed image gradually decreases with the gradual increase of noise intensity. However, the reconstructed images can still be clearly recognized.

The PSNR values of the images are in the interval [14.4746, 23.6611] dB, [17.1695, 35.6329] dB, and [35.1842, 35.6329] dB when the intensity of the three noises increases from 1×10^{-6} to 9×10^{-6} in order. This indicates that the proposed algorithm is robust to the three noise attacks.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 15. Ciphertext images and the corresponding reconstructed images under GN attack: (a)–(e) the ciphertext images under GN intensity in order of the intensity of 1×10^{-6}, 3×10^{-6}, 5×10^{-6}, 7×10^{-6}, 9×10^{-6}, (f)–(j) reconstructed images corresponding to panels (a)–(e).

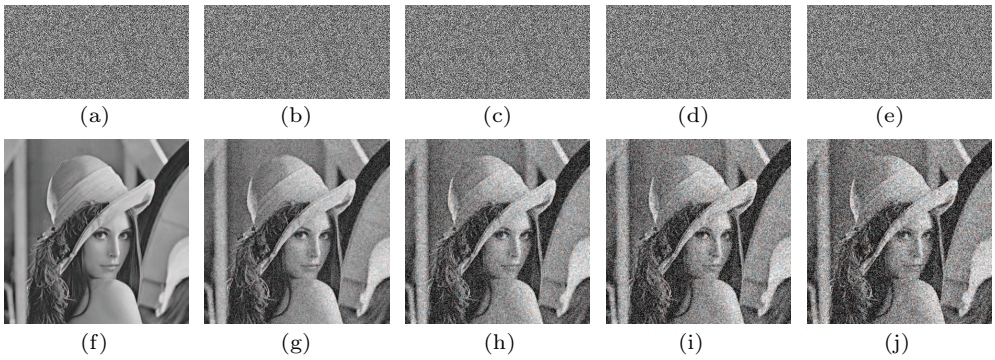


Fig. 16. Ciphertext images and the corresponding reconstructed images under SN attack: (a)–(e) the ciphertext images under SN intensity in order of the intensity of 1×10^{-6}, 3×10^{-6}, 5×10^{-6}, 7×10^{-6}, 9×10^{-6}, (f)–(j) reconstructed images corresponding to panels (a)–(e).

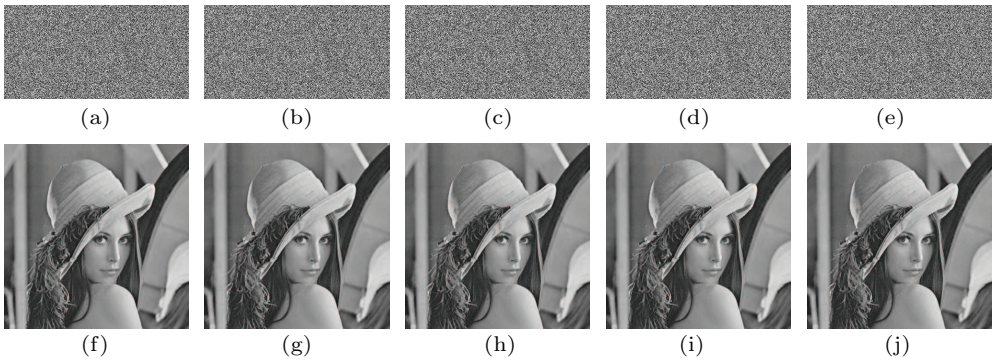


Fig. 17. Ciphertext images and the corresponding reconstructed images under SPN attack: (a)–(e) the ciphertext images under SPN intensity in order of the intensity of 1×10^{-6}, 3×10^{-6}, 5×10^{-6}, 7×10^{-6}, 9×10^{-6}, (f)–(j) reconstructed images corresponding to panels (a)–(e).

Table 10. PSNR values under different noise attack intensities.

Noise type	Noise intensity				
	1×10^{-6}	3×10^{-6}	5×10^{-6}	7×10^{-6}	9×10^{-6}
GN (dB)	23.6611	17.0256	15.6565	15.0281	14.4746
SN (dB)	35.6329	23.3445	19.6822	18.1876	17.1695
SPN (dB)	35.6329	35.6329	35.6329	35.6329	35.1842

4.7.2. Cropping attack analysis

In this section, we use the "Lena" image as an example to test the cropping attack on its ciphertext image. The cut sizes are 32×32, 64×64, 128×128, and the corresponding ciphertext images are decrypted, and the results are shown in Fig. 18. The PSNR values between the corresponding images are calculated, and the results are shown in Table 11. It can be seen that the quality of the reconstructed image decreases as the cut size increases. Even if $1/4$ of the ciphertext image is lost, the reconstructed image can still be recognized by human eyes, which indicates that the proposed algorithm is robust to the cropping attack.

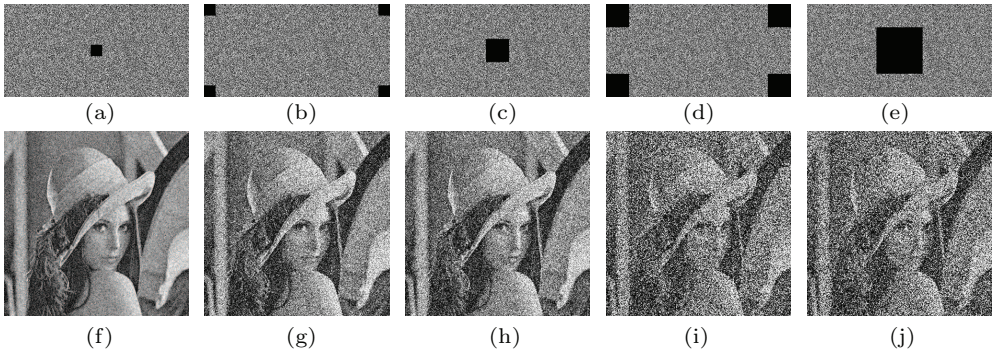


Fig. 18. Ciphertext images and the corresponding reconstructed images under cropping attack.

Table 11. PSNR values under different cropping attack intensities.

Cutting size	32×32	64×64		128×128	
	Middle	Four corners	Middle	Four corners	Middle
PSNR (dB)	20.4576	15.2572	15.1284	11.0488	11.0078

4.8. Known-plaintext and chosen-plaintext attacks analyses

Known-plaintext attacks and chosen-plaintext attacks are two common attacks in cryptosystems, and this section analyzes the ability of the proposed algorithm to resist these attacks.

On the one hand, the 256-bit hash key K based on the plaintext image and the external keys t_i jointly determine the initial values x_0, y_0, z_0, w_0 of the 4D MHS to generate the four hyperchaotic sequences X, Y, Z, W. These sequences will be used to generate the initial value s_0 and the sequences of system control parameters u_i, v_i of the 1D CCS, to scramble the approximation component cA, to generate the two random phase masks $M1$ and $M2$ of the DRPE encryption process, to generate the Latin square matrix L, to participate in the confusion process and the diffusion process.

On the other hand, the row key k_r and the column key k_c based on the plaintext image also participate in the confusion process and the diffusion process. Therefore, the whole process of the proposed algorithm is closely related to the plaintext image information, and the algorithm can effectively resist both known-plaintext attacks and chosen-plaintext attacks.

4.9. Performance analysis of the DCMM

In this section, we compare the performance of the deterministic chaotic measurement matrix (DCMM) with the Gaussian matrix, Bernoulli matrix, Toeplitz matrix, part Hadamard matrix, sparse random matrix, and circular matrix.

The seven measurement matrices mentioned above are tested separately, and their results are shown in Fig. 19 and Table 12. It can be seen that the DCMM has a similar image reconstruction effect with the remaining six random measurement matrices, which shows the effectiveness of the DCMM constructed in this paper. The average PSNR value of DCMM is 34.1084 dB, which is only 0.397 dB less than the maximum average PSNR value of 34.5054 dB. However, the DCMM can substantially reduce the space consumption of keys in transmission and storage.

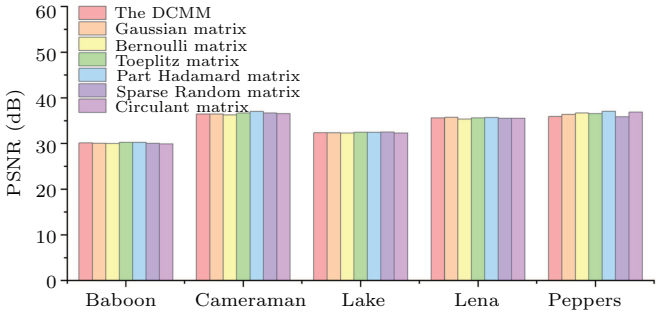


Fig. 19. PSNR values of the reconstruction results using different measurement matrices.

Table 12. Comparison of PSNR (in unit dB) of reconstructed images with different measurement matrices.

Images	DCMM	Different measurement matrices					
		Gaussian matrix	Bernoulli matrix	Toeplitz matrix	Part Hadamard matrix	Sparse random matrix	Circular matrix
Baboon	30.1355	30.0567	29.9893	30.2449	30.2772	30.0732	29.9025
Cameraman	36.4641	36.4816	36.2704	36.6841	37.0158	36.6563	36.5611
Lake	32.3941	32.3537	32.2991	32.4731	32.4722	32.5032	32.3084
Lena	35.6329	35.7494	35.3874	35.6174	35.6915	35.5575	35.5535
Peppers	35.9155	36.3818	36.6666	36.5536	37.0705	35.8826	36.8952
Average	34.1084	34.2046	34.1226	34.3146	34.5054	34.1346	34.2441

4.10. Comparison with other compressed sensing encryption schemes

In this section, the proposed scheme in this paper is compared and analyzed with the schemes proposed in Refs. [30,52–56] in terms of the performance of the algorithms. In order to provide a relatively fair comparison environment, the comparison data are algorithmic data from the relevant literatures, where the test images are all 512×512 in size, and the comparison results are shown in Table 13.

Table 13. Comparison with other compressed sensing encryption schemes.

Scheme	Image	PSNR (dB)	MSSIM	Correlation coefficient			Differential attack (%)		IE	CR	Key space
				H	V	D	NPCR	UACI			
Ref. [30]	Lena	34.5560	0.9995	0.0069	–0.0028	–0.0047	–	–	–	0.5	$2^{360}+2^{144}$
Ref. [52]	Lena	33.2299	0.9935	–0.0029	0.0058	–0.0025	99.60	33.46	7.9986	0.5	$>2^{260}$
Ref. [53]	Lena	26.3369	–	0.0053	0.0193	0.0035	99.62	33.33	7.9984	0.5	10^{135}
Ref. [54]	Lena	35.51	–	–0.0016	0.0010	–0.0015	99.61	33.42	–	0.5	$\approx 2^{149}$
Ref. [55]	Lena	35.3844	0.9964	–0.0018	–0.0070	–0.0024	–	–	7.9986	0.5	$>2^{372}$
Ref. [56]	Peppers	35.3412	0.9944	0.0086	–0.0027	–0.0092	99.61	33.46	7.9987	0.55	$>2^{320}$
Ours	Lena	35.6329	0.9941	–0.0019	0.0051	–0.0078	99.62	33.57	7.9984	0.5	$>2^{400}$
	Peppers	35.9155	0.9921	0.0006	–0.0098	0.0071	99.60	33.49	7.9983	0.5	$>2^{400}$
	Lake	32.3941	0.9938	0.0011	0.0030	0.0059	99.59	33.49	7.9987	0.5	$>2^{400}$
	Cameraman	36.4641	0.9913	0.0007	0.0084	0.0083	99.61	33.42	7.9986	0.5	$>2^{400}$
	Baboon	30.1355	0.9974	–0.0036	–0.0033	–0.0039	99.59	33.45	7.9986	0.5	$>2^{400}$

It can be seen that compared with the schemes proposed in Refs. [30,52–56], The proposed scheme has a good performance in various tests, such as the quality of reconstructed images is better and the security of the scheme is higher. The average PSNR value of the five test images is 34.1084 dB, which is higher than the average PSNR value of the comparison schemes of 33.3931 dB. The average value of MSSIM is 0.9937, which is close to 1. The average correlation coefficient of adjacent pixels in the ciphertext images is 0.0045, which is lower than the average of 0.0064 in the comparison schemes. In the differential attack test, the average NPCR value is 99.6090%, close to the ideal value of 99.6094%, and the average UACI value is 33.4855%, close to the ideal value of 33.4635%; The average information entropy of ciphertext images is 7.99852, which is close to the average of 7.99857 in the comparison schemes. The key space of the proposed scheme is $10^{140} > 2^{400} \gg 2^{100}$, and among all the comparison schemes, the key space is the largest.

4.11. Running time analysis

In this section, the running time of the proposed algorithm is analyzed. Table 14 provides the time required to process different images using this algorithm. It can be observed that the average time required to encrypt a plaintext image of size 512×512 is approximately 2.2028 s, while the average time required to decrypt a ciphertext image of size 256×512 is approximately 8.9843 s.

Taking the "Lena" image as an example, the specific distribution of time required for encryption and decryption is analyzed, as shown in Fig. 20. The encryption process mainly includes the stages of keys and sequences generation, DWT, first round of confusion, CS, DRPE encryption, second round of confusion (CMDPHL), and diffusion (DMDPH), which takes a total of 2.1947 s. Among these stages, diffusion consumes the most time, accounting for nearly 40%. The decryption process mainly includes the stages of inverse diffusion, inverse second round of confusion, DRPE decryption, OMP reconstruction, inverse first round of confusion, and inverse DWT, which takes a total of 8.9773 s. Among these stages, OMP reconstruction takes the longest time, exceeding 80%.

Table 14. Encryption and decryption times for different images.

Image	Time (s)					
	Baboon	Cameraman	Lake	Lena	Peppers	Average
Encryption time	2.1883	2.2150	2.2025	2.1947	2.2136	2.2028
Decryption time	9.0379	8.9849	8.9434	8.9773	8.9782	8.9843

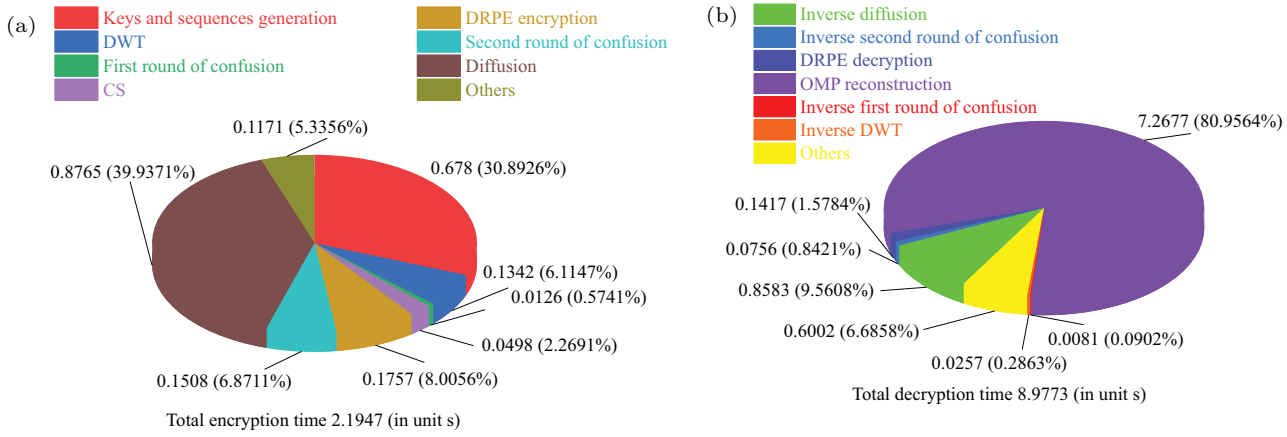


Fig. 20. The encryption and decryption time distribution of the "Lena" image.

5. Conclusion

In this paper, an optical image encryption algorithm based on a new 4D MHS and compressed sensing is proposed. Firstly, we designed the 4D MHS and proved that it has a larger key space, richer dynamical behavior, and more complex hyperchaotic properties. The system improves the security proposed of algorithm. Secondly, we proposed a method for constructing deterministic chaotic measurement matrix (DCMM). The control parameter sequences and initial value of the system are determined by four sequences of 4D MHS, and the DCMM is obtained by bringing them into 1D CCS. The DCMM obtained through this method has strong randomness, effectively reducing the transmission and storage space of keys while ensuring the quality of reconstructed images. Thirdly, DRPE encryption used fractional Fourier transform, which increased the key space and improved the security of the algorithm. Fourthly, the confusion method and diffusion method needed the row key k_r and column key k_c based on plaintext images. The generation of four sequences and the DCMM are closely related to the hash key of plaintext image, and the whole algorithm is sensitive to plaintext image. The experimental results and performance analysis show that the

proposed algorithm has a large key space, high key sensitivity, better image reconstruction quality, good security, and effectiveness, and can resist a variety of attacks.

Although the proposed algorithm has good security, it is not efficient enough in terms of time efficiency. Taking the image with the size of 512×512 as an example, the average encryption time of this algorithm is 2.2028 s, and the average decryption time is 8.9843 s. In the follow-up work, we will try to combine with deep learning to improve the time efficiency of the algorithm on the premise of ensuring sufficient security of the algorithm.

References

[1] Yang Y, Cheng M, Ding Y and Zhang W 2023 *IEEE Transactions on Services Computing* **16** 2387
[2] Jiang D, Njitacke Z T, De Dieu Nkapkop J, Tsafack N, Wang X and Awrejcewicz J 2023 *IEEE Internet of Things Journal* **10** 7143
[3] Yu F, Kong X, Mokbel A A M, Yao W and Cai S 2023 *IEEE Transactions on Circuits and Systems II: Express Briefs* **70** 326
[4] Su Y, Teng L, Liu P, Unar S, Wang X and Fu X 2023 *IEEE Transactions on Circuits and Systems for Video Technology* **33** 4689
[5] Wang B, Song J, Li R, Han R, Zheng G and Yang H N 2019 *Chin. Phys. B* **29** 014207
[6] Kong P, Wang B D, Wang P, Zivkovic V and Zhang J Q 2020 *Chin. Phys. B* **29** 074201
[7] Chai X, Fu J, Gan Z, Lu Y and Zhang Y 2022 *Nonlinear Dyn.* **108** 2671
[8] Cheng J, Yan X, Liu L, Jiang Y and Wang X 2022 *Entropy* **24** 340
[9] Chai X, Wang Y, Chen X, Gan Z and Zhang Y 2022 *IEEE Signal Processing Letters* **29** 972
[10] Fridrich J 1997 *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, 12–15 October 1997, pp. 1105–1110
[11] Wang X, Liu C and Zhang H 2016 *Nonlinear Dyn.* **84** 1595
[12] Liu L and Miao S 2018 *Multimedia Tools and Applications* **77** 21445
[13] Pak C and Huang L 2017 *Signal Processing* **138** 129
[14] Wang S, Wang C and Xu C 2020 *Optics and Lasers in Engineering* **128** 105995
[15] Zhou Y, Bao L and Chen C L P 2014 *Signal Processing* **97** 172
[16] Seyedzadeh S M and Mirzakuchaki S 2012 *Signal Processing* **92** 1202
[17] Hua Z, Fan Jin B X and Huang H 2018 *Signal Processing* **149** 148
[18] Zhou M and Wang C 2020 *Signal Processing* **171** 107484
[19] Li C, Li Z, Feng W, Tong Y, Du J and Wei D 2019 *AEU - International Journal of Electronics and Communications* **110** 152861
[20] Wang Z, Min F and Wang E 2016 *AIP Adv.* **6** 095316
[21] Ye X, Wang X, Gao S, Mou J, Wang Z and Yang F 2020 *Nonlinear Dyn.* **99** 1489
[22] Peng G, Min F and Wang E 2018 *Journal of Electrical and Computer Engineering* **2018** 8649294
[23] Donoho D L 2006 *IEEE Transactions on Information Theory* **52** 1289
[24] Xu Q, Sun K, Cao C and Zhu C 2019 *Optics and Lasers in Engineering* **121** 203
[25] Zhou N, Li H, Wang D, Pan S and Zhou Z 2015 *Opt. Commun.* **343** 10
[26] Yang Y, Guan B, Li J, Li D, Zhou Y and Shi W 2019 *Optics & Laser Technology* **119** 105661
[27] Zhou N, Pan S, Cheng S and Zhou Z 2016 *Optics & Laser Technology* **82** 121
[28] Xu Q, Sun K, He S and Zhu C 2020 *Optics and Lasers in Engineering* **134** 106178
[29] Huo D, Zhu Z, Wei L, Han C and Zhou X 2021 *Opt. Commun.* **492** 126976
[30] Luo Y, Lin J, Liu J, Wei D, Cao L, Zhou R, Cao Y and Ding X 2019 *Signal Processing* **161** 227
[31] Zhang B, Xiao D and Xiang Y 2021 *IEEE Transactions on Multimedia* **23** 2656
[32] Shi H and Wang L D 2019 *Acta Phys. Sin.* **68** 200501 (in Chinese)
[33] Guo Y, Chen W and Jing S W 2020 *Chin. Phys. B* **29** 054203
[34] Wang L and Zhao S 2020 *Chin. Phys. B* **29** 024204
[35] Refregier P and Javidi B 1995 *Opt. Lett.* **20** 767
[36] Javidi B 1997 *Phys. Today* **50** 27
[37] Chen L and Zhao D 2005 *Opt. Commun.* **254** 361
[38] Zhao D, Li X and Chen L 2008 *Opt. Commun.* **281** 5326
[39] Unnikrishnan G, Joseph J and Singh K 2000 *Opt. Lett.* **25** 887
[40] Zhou N, Wang Y and Gong L 2011 *Opt. Commun.* **284** 3234
[41] Liu Z, Zhang Y, Liu W, Meng F, Wu Q and Liu S 2013 *Optics and Lasers in Engineering* **51** 967
[42] Chen J, Zhu Z, Fu C, Yu H and Zhang L 2015 *Commun. Nonlinear Sci. Numer. Simul.* **20** 846
[43] Faragallah O S and Afifi A 2017 *Opt. Quantum Electron.* **49** 89
[44] Zhang L, Zhou Y, Huo D, Li J and Zhou X 2018 *Optics & Laser Technology* **105** 162
[45] Huo D, Zhou X, Zhang L, Zhou Y, Li H and Yi S 2018 *J. Mod. Opt.* **65** 2093
[46] Xie Z L, Ma H T, Qi B, Ren G, Tan Y F, He B, Zeng H L and Jiang C 2015 *Chin. Phys. Lett.* **32** 124203
[47] Xie Z L, Qi B, Ma H T, Ren G, Tan Y F, He B, Zeng H L and Jiang C 2016 *Chin. Phys. Lett.* **33** 044206
[48] McBride A C and Kerr F H 1987 *IMA J. Appl. Math.* **39** 159
[49] Hua Z, Zhu Z, Chen Y and Li Y 2021 *Nonlinear Dyn.* **104** 4505
[50] Wu Y, Noonan J P and Agaian S 2011 *IEEE International Conference on Systems, Man, and Cybernetics*, 9–12 October 2011, pp. 3358–3364
[51] Chua L O and Sung Mo K 1976 *Proc. IEEE* **64** 209
[52] Gan Z, Chai X, Zhang J, Zhang Y and Chen Y 2020 *Neural Computing and Applications* **32** 14113
[53] Liang Y R and Xiao Z Y 2020 *International Journal of Automation and Computing* **17** 292
[54] Zhu L, Song H, Zhang X, Yan M, Zhang L and Yan T 2019 *IEEE Access* **7** 22161
[55] Fu J, Gan Z, Chai X and Lu Y 2022 *Multimedia Tools and Applications* **81** 17401
[56] Lu Y, Gong M, Huang Z, Zhang J, Chai X and Zhou C 2022 *Optik* **263** 169357