Physica Scripta

PAPER

Multi-wing chaotic system based on meminductor and its application in image encryption

RECEIVED
14 July 2024

REVISED
23 September 2024

ACCEPTED FOR PUBLICATION
2 October 2024

PUBLISHED
15 October 2024

Pengfei Ding*, Weiwei Hu, Penghui Geng and Le Yang

School of Electronics and Engineering, Xi'an University of Posts and Telecommunications, Xi'an, shaanxi, 710121, People's Republic of China

* Author to whom any correspondence should be addressed

E-mail: dpf@xupt.edu.cn

Keywords: magnetically controlled meminductor, image encryption, bit plane decomposition, multi-wing chaotic system

Abstract

Meminductor is a novel type of nonlinear device following the memristor, characterized by its memory properties. Currently, research on meminductors is still in its infancy, with their physical devices yet to be formally realized. Therefore, conducting fundamental research on their nonlinear circuit properties and applications is of great significance. In this paper, a new multi-wing chaotic system is proposed based on the mathematical model of a magnetically controlled meminductor. By varying the values of its parameters, the system can generate two-wing, three-wing, and four-wing chaotic attractors. Various analytical methods are employed to study the dynamical behaviours of the proposed chaotic system. The results demonstrate that the system is highly sensitive to its initial conditions and control parameters, which makes it suitable for image encryption. Based on the new system, we propose a new algorithm for image encryption that combines the newly established four-dimensional multi-wing chaotic system with bit plane decomposition technique, firstly, the high four-bit planes containing 94% image information are disordered by S-type permutation, then the disordered bit planes perform operation of XOR with the random matrix generated by chaotic sequences, and finally, the encrypted image is obtained by merging the bit planes.

1. Introduction

In 1971, Chua discovered the existence of a new basic circuit element, which was named the memristor [1]. The physical realization of a memristor using nanotechnology in HP labs in 2008 led to a wide scope of research on memristors and their various application circuits, more and more scholars carried out research on memory devices [2]. In 2009, Ventra *et al* further expanded the scope of memory elements by introducing new memory elements of meminductor and memcapacitor. The memcapacitor describes the mathematical relationship between the integral of charge and magnetic flux, while the meminductor describes the mathematical relationship between the integral of magnetic flux and charge [3]. Compared to the memristor, there is fewer research on the meminductor and memcapacitor, and the characteristics and applications of meminductor and memcapacitor are still not well understood, so it is of great significance to research on meminductor and memcapacitor.

As an emerging discipline in the 20th century, the chaos phenomenon has penetrated into almost every corner of human society, and has become a research hotspot in recent years [4–8]. In 1963, American meteorologist Lorenz proposed the first mathematical model of a chaotic system [9]. In 1975, Li and Yorke pointed out that 'Period three means chaos' for the first time [10], and 'chaos' as a professional term is formally established. The first combination of chaos and nonlinear circuits was Chua's circuit [11], followed by the chaotic oscillator circuit constructed by paralleling two memristors [12]. Since then, there has been a steady stream of research on the modeling of memory devices [13–16]. Compared with the general chaotic systems, the chaotic systems based on memory devices have some new dynamical phenomena, such as richer dynamics and

more complex chaotic attractors, so the study of memory devices and their chaotic oscillatory circuits has great theoretical and engineering significance.

Multi-wing chaotic system means that the generated attractor has two or more wings, which reflecting the diversity of its structure [17–19]. Therefore, multi-wing chaotic systems exhibit more complex dynamical behaviours and have important engineering applications and theoretical research significance [20–23]. In 2020 Xie *et al* introduced a new function to generate various multi-scroll and multi-wing hidden attractors [24]. In 2021 Lin *et al* constructed a simple multi-winged chaotic system without polynomial functions by introducing a sinusoidal function into Sprott system [25].

With the rapid development of network communication, there are a large number of images are generated or transmitted every day, which means that network security threat is an important issue. Chaotic system has the properties of good randomness and sensitivity to initial value, which is very suitable for image encryption. At present, image encryption using chaotic system became a research hotspot [26–28]. In recent years, researchers proposed the method of decomposing the image into bit plane level and then encrypting it, the advantage of this method is that each pixel is decomposed into 8-bit planes, and the disorder and diffusion can achieve better results [29, 30].

With a long period of research, the field of image encryption is fruitful, but there are still some problems to be solved: Firstly, some encryption algorithms utilize simple one-dimensional chaotic mapping, which has the disadvantage of low security. Secondly, some algorithms use a single disorder algorithm or diffusion algorithm that has a security risk. Finally, some encryption algorithms use a fixed single key, which makes the encryption algorithm insensitive to the change of the plaintext. In this paper, the chaotic system is combined with the bit plane decomposition to design the image encryption algorithm, which can better protect the image information.

This paper is structured as follows: in section 2, we introduce and analyze the proposed meminductor based multi-wing chaotic system. In section 3, we examine the dynamical properties of the proposed system through the 0–1 test and complexity analysis. In section 4, we propose a new image encryption scheme by combining the proposed meminductor multi-wing chaotic system with bit-plane decomposition. The performance analyses of the proposed encryption scheme are presented in section 5, where we also compare the results with other encryption schemes. Finally, section 6 provides the concluding analysis of our research.

2. The construction of new system

As a new type of memory component, meminductor is often used for the construction of chaotic circuits, and it can also be used in the design of chaotic systems. This paper gives a new chaotic system based on a magnetically-controlled meminductor model [31], and it is described by equation (1).

$$\begin{cases} \dot{x} = ax - yz + W(w) \\ \dot{y} = xz - by \\ \dot{z} = xy - cz - hxw \\ \dot{w} = -kxW(w) \end{cases} \tag{1}$$

In equation (1), a, b, c, k and h are system parameters, x, y and z are system variables, $W(w)$ is the mathematical model of the magnetic-controlled meminductor, The φ-i characteristics of the magnetically-controlled meminductor are shown in equation (2) and equation (3).

$$\begin{cases} q(w) = mw + n'w^3 \\ W(w) = \dfrac{dq(w)}{dw} = m + 3n'w^2 = m + nw^2 \end{cases} \tag{2}$$

$$\begin{cases} i(t) = L^{-1}(w)\varphi(t) \\ L^{-1}(w) = \dfrac{dq(w)}{dw} \end{cases} \tag{3}$$

In equation (2), $n = 3n'$, m and n are constant coefficients of the meminductor model, where $w = \int_{-\infty}^{t} \varphi(\tau)d(\tau)$ is an intermediate integral variable, and $q(w)$ is a smooth cubic monotonic convex nonlinear function. In equation (3), $i(t)$ is the current flowing through the meminductor at time t, φ(t) represents the flux of the meminductor at time t, and $L^{-1}(w)$ is the reciprocal value of the meminductor.

The φ-i hysteresis curve of the magnetically controlled meminductor model with $m = 0.42$, $n = -0.5$ and the excitation voltage of $A\sin(2\pi fx)$ is shown in figure 1.

Where A is the amplitude, and f is the frequency. When the current amplitude is taken as $A = 1.0$ A and the frequencies are set as $f_1 = 60$ Hz, $f_2 = 90$ Hz, and $f_3 = 120$ Hz, respectively. The hysteresis loop of the

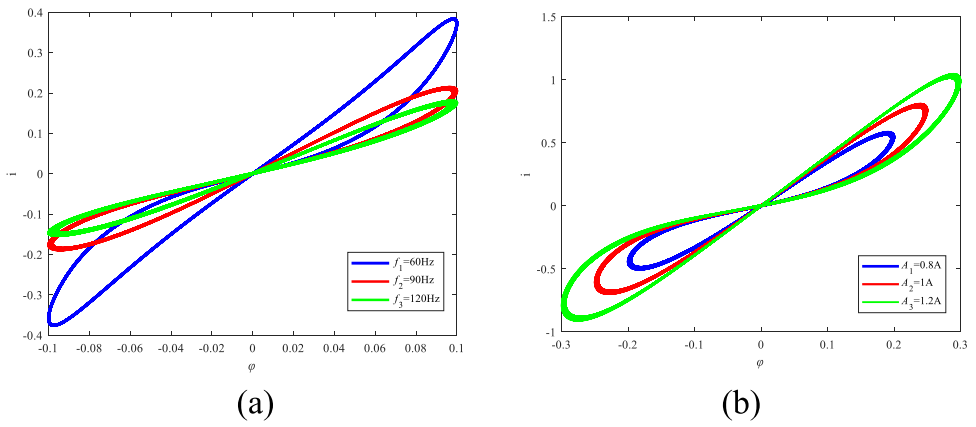

Figure 1. The hysteresis loops of the new meminductor. (a) The hysteresis loops of frequencies with $f_1 = 60$ H_Z, $f_2 = 90$ H_Z and $f_3 = 120$ H_Z ($A = 1.0$ A). (b) The hysteresis loops of amplitudes with $A_1 = 0.8$ A, $A_2 = 1.0$ A and $A_3 = 1.2$ A ($f = 90$ H_Z).

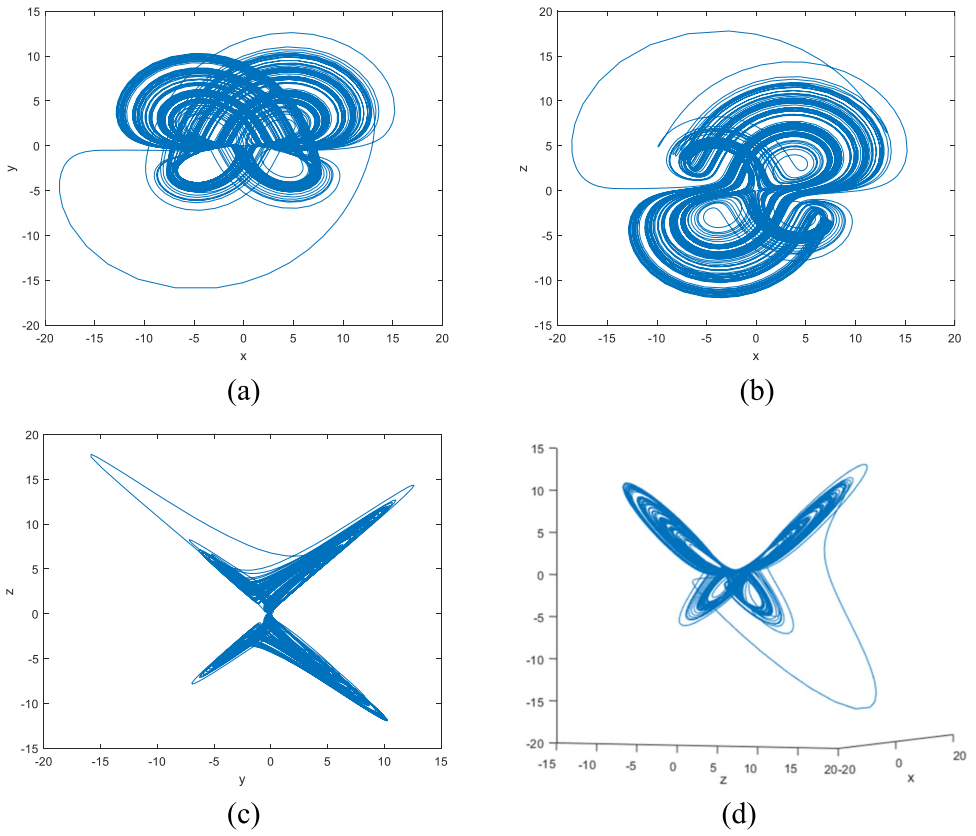

Figure 2. Chaotic attractor phase diagram of the system. (a) on x-y plane (b) on x-z plane (c) on y-z plane (d) on x-y-z space.

meminductor gradually transforms into a straight line as the frequency increases, as shown in figure 1(a). When the frequency is taken as $f = 90$ H_Z, and the current amplitudes are set to $A_1 = 0.8$ A, $A_2 = 1.0$ A and $A_3 = 1.2$ A, the hysteresis loop gradually tightens with decreasing amplitude as shown in figure 1(b), so the introduced meminductor satisfies the basic property of meminductor.

When the parameters of the system are given as $a = 1.25$, $b = 5.35$, $c = 3.3$, $k = 1.25$, $m = 0.42$, $n = -0.5$, and $h = 0.5$, and the initial conditions are (0, 0, 0.001, 0.001), the chaotic attractor generated by the system is presented in figure 2. As can be observed from figure 2 that the system generates a four-wing chaotic attractor, which is proved by the phase diagram in the x-y, x-z, y-z, and x-y-z planes.

In particular, the system is a multi-wing chaotic system with high sensitivity to system parameters. When the system parameters are changed, the number of wings corresponding to its attractor is also changed. When parameter c is 0.86 and 1.12, the new system is capable of obtaining two-wing and three-wing chaotic attractors respectively, and they are shown in figures 3 and 4.

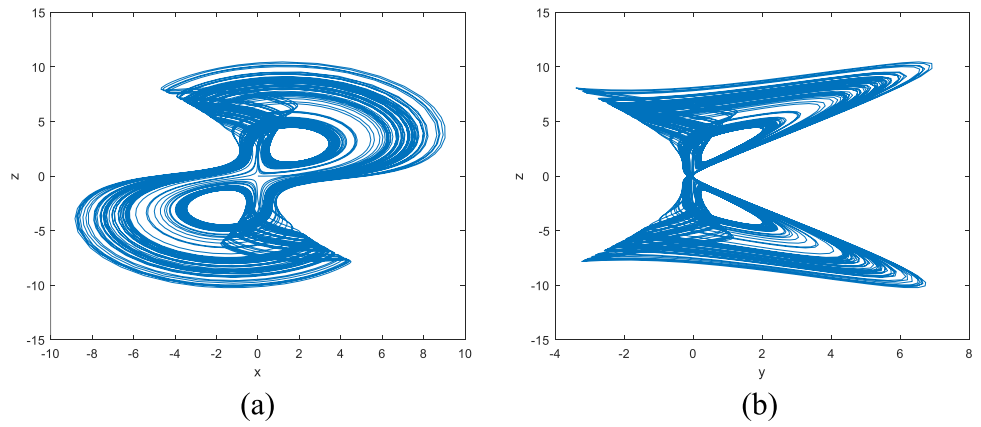

Figure 3. Two-wing chaotic attractor. (a) on x-z plane (b) on y-z plane.

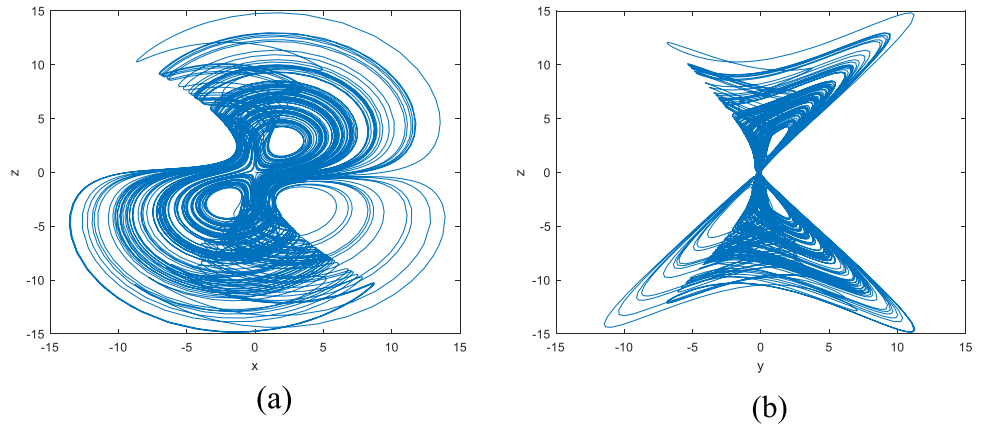

Figure 4. Three-wing chaotic attractor. (a) on x-z plane (b) on y-z plane.

3. Dynamic characteristics analyses of chaotic system

3.1. Dissipative analysis

For the new system, its dissipativity is described as equation (4).

$$\nabla V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{w}}{\partial w} = a - b - c - 2\,knxw \tag{4}$$

By substituting the parameters and initial values of the chaotic attractor in the above section into the calculation, the result of $\nabla V = -7.4 < 0$ can be obtained. The divergence value of the system is less than zero, so when $t \to \infty$, the volume element of the system track gradually decreases at an exponential rate of $e^{(a-b-c-2knxw)}$, its progressive dynamic behavior will eventually be fixed on the attractor.

3.2. Equilibrium point analysis

Let the right-hand side of equation (1) be zero, the following equation can be obtained.

$$\begin{cases} ax - yz + W(w) = 0 \\ xz - by = 0 \\ xy - cz - hxw = 0 \\ -kxW(w) = 0 \end{cases} \tag{5}$$

To facilitate discussion, substitute the following values into the new system: $a = 1.25$, $b = 5.35$, $c = 3.3$, $k = 1.25$, $m = 0.42$, $n = -0.5$, $h = 0.5$. The system can be obtained with 10 equilibrium points: $P_{1,2} = (0, 0, 0, \pm 0.917)$, $P_{3,4} = (4.704, 2.274, 2.568, \pm 0.917)$, $P_{5,6} = (4.704, 2.274, -2.568, \pm 0.917)$, $P_{7,8} = (-3.756, -2.273, 2.586, \pm 0.917)$, $P_{9,10} = (-3.756, -2.273, -2.586, \pm 0.917)$. The Jacobian matrix J at equilibrium point

Table 1. Eigenvalues corresponding to different equilibrium points.

	λ_1	λ_2	λ_3	λ_4
P_1	−3.3	−0.0344	1.2844	−5.35
P_2	−3.3	−0.5572	1.8072	−5.35
P_3	−9.0941	0.8482+3.1883i	0.8482–3.1883i	5.3897
P_4	0.8493+3.3943i	0.8493–3.3493i	−5.3446	−9.146
P_5	−7.963	−0.0082	0.5662	5.397
P_6	0.3055	−0.3921	−0.8301	−4.6753
P_7	0.3908+3.4597i	0.3908–3.4597i	−8.195	−4.492
P_8	−8.152	0.5970+2.9221i	0.5970–2.9221i	3.8632
P_9	0.405+0.417i	0.405–0.417i	−6.5902	−4.3052
P_{10}	−6.9559	3.8944	0.5481	−0.5813

$p_i(i = 1, 2, 3...10)$ is presented as follows:

$$J = \begin{vmatrix} a & -z & -y & 2\,nw \\ z & -b & x & 0 \\ y - hw & x & -c & -hx \\ -kW & 0 & 0 & -2\,knxw \end{vmatrix} \tag{6}$$

The eigenvalue at equilibrium point $P_i(i = 1, 2, 3\ldots10)$ is obtained through calculating the formula of det $(\lambda E-J) = 0$, in which λ denotes the eigenvalue, and E is a unit matrix. Substituting the parameters of the system and the value of the equilibrium point into formula of det $(\lambda E-\mathrm{J}) = 0$. The eigenvalues associated with each equilibrium point can be acquired and shown in table 1.

The stability judgment conditions for a fourth order polynomial according to Routh-Hurwit, the equilibrium point P_1, P_2, P_5, P_6 and P_{10} is characterized as an unstable saddle point, while the equilibrium points P_3, P_4, P_7, P_8 and P_9 exhibits characteristics of an unstable saddle focus.

3.3. Impact of system parameters on dynamic behaviours

The dynamical behaviours of the system rely on its parameters and initial values. In order to conduct further study and analysis on the chaotic dynamical behaviours of the new system, the dynamical behaviors are specifically analyzed as follows. The parameters are fixed as $b = 5.35$, $k = 1.25$, $m = 0.42$, $n = -0.5$ and $h = 0.5$, initial conditions are (0, 0, 0.001, 0.001), while parameters a and c are independently adjustable, the dynamical behaviours of the new system with the variation of the system parameters of a and c is investigated with the help of conventional dynamical analytical methods such as Lyapunov exponents diagrams and bifurcation diagrams.

Take the system parameter a as a variable with range of $a \in (1, 2.5)$, the Lyapunov exponential spectrum and the bifurcation diagram are shown in figures 5(a)–(c), it can be seen that when a is increased from 1 to 1.11, the system is in the periodical state, in which the phase diagram with $a = 1.08$ as an example is shown in figure 6(a) . When $a \in (1.11, 1.358)$, the system enters into chaotic state, select $a = 1.25$ as an example, the chaotic attractor phase diagram of the system is shown in figure 2, and the system exhibits a four-wing chaotic state. When $a \in (1.358, 1.41)$, the system enters periodic state from chaotic state, where the phase diagram with $a = 1.363$ as an example is shown in figure 6(b), it can be seen that the state of the system at this point is quasi-periodic. When $a \in (1.41, 2.11)$, the system enters into chaotic state from periodic state, at $a = 1.58$, the chaotic phase diagram of the system is shown in figure 6(c). At $a > 2.11$, analysis in conjunction with the bifurcation diagram in figure 5(c) shows that the dynamics of the system shifts to a periodic behavior and eventually enters a steady state, the chaotic phase diagram when $a = 2.5$ is shown in figure 6(d), where it can be clearly seen that the system is in a periodic state.

Taking the system parameter c as a variable, when $c \in (1, 5)$, the Lyapunov exponential spectrum of the system and the bifurcation diagram with the variation of c are plotted in figures 7(a)–(c) , when $c \in (1, 4.46)$, the system is in a chaotic state, and there is a hyper chaotic state at some points. For example, hyperchaotic state occurs when $c = 2.25$, and its corresponding phase diagram is shown in figure 8(a). when $c > 4.46$, the dynamical state of the system with c is evolved from chaotic state to periodic state, where the chaotic phase diagram with $c = 5$ is shown in figure 8(b) and the system presents a periodic state.

3.4. Initial value sensitivity test

For system equation (1), when choosing the system parameters $a = 1.25$, $b = 5.35$, $c = 3.3$, $k = 1.25$, $m = 0.42$, $n = -0.5$ and $h = 0.5$, the initial values of the system $x_0 = 0$, $y_0 = 0$, $z_0 = 0.001$ and $w_0 = 0.001$, with a small change of 10^{-14} are used for analyzing the initial value sensitivity. The time series of state variable x with initial values of $x_0 = 0$ and $x_0 = 0+10^{-14}$ are simulated and shown in figure 9(a). From figure 9(a), it can be seen that after a finite time, the time series with initial values of x_0 and $x_0 = 0+10^{-14}$ for the state variable x are completely

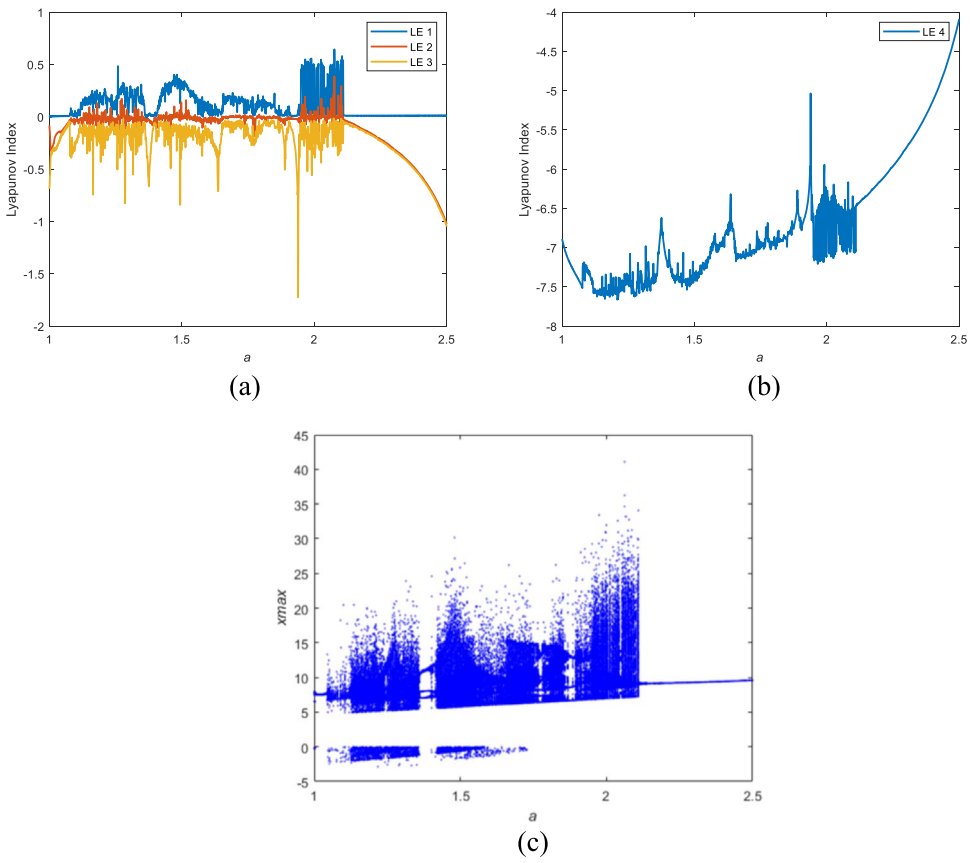

Figure 5. (a) Lyapunov exponents spectrum of LE_1, LE_2 and LE_3 with $a \in (1, 2.5)$; (b) Lyapunov exponents spectrum of LE_4 with $a \in (1, 2.5)$; (c) bifurcation diagram with $a \in (1, 2.5)$.

separated. Additionally, the state variables of x, y, z and w with different initial values are confirmed, and the simulation results are shown in figures 9(b)–(d), the time series of state variables still showing significant difference, which illustrate the sensitivity of the system to its initial values.

The initial value sensitivity test results in figure 9 indicates that if the minimum precision of a computer is less than 10^{-14}, it recognizes the value of x_0+10^{-14} as x_0. As a results, the results calculated by the computer with the initial values of x_0 and x_0+10^{-14} are exactly the same, which calculated results is quite different from the actual calculation result with x_0+10^{-14} as the initial value. From this, it can be concluded that the stronger the sensitivity of the system, the higher the minimum precision of the computer is required.

3.5. 0–1 test of the system

The 0–1 test of the system is a method for distinguishing the type of system behavior that can be used to determine whether a dynamical system in a periodic or chaotic state. This method has particular advantages as it can be directly applied to time series without the requirement phase space reconstruction, making it versatile for analyzing diverse dynamical systems. The characteristics of the dynamical system can be inferred from the results of 0–1 test.

As to a regular dynamical system, the motion trajectories of 0–1 tests usually behave as bounded, while the motion trajectories of 0–1 tests of a chaotic dynamical systems similar to Brownian motion. When the initial value of (x_0,y_0,z_0,w_0) equals $(0, 0, 0.001, 0.001)$ and $a = 1.25$, $b = 5.35$, $c = 3.3$, $k = 1.25$, $m = 0.42$, $n = -0.5$, $h = 0.5$, the 0–1 test experimental result of the novel system is presented in figure 10, and the results indicate that the motion trajectories of the 0–1 tests exhibit a striking resemblance to Brownian motion, thus revealing that the system is in chaotic state.

3.6. Complexity analysis

In addition to phase diagrams, bifurcation diagrams, Lyapunov exponents, and complexity can effectively describe the dynamic characteristics of a chaotic system. In particular, the similarity between chaotic and random sequences is quantified by using complexity. A higher degree of randomness sequence is indicated by a larger complexity value; hence the sequence's unpredictability can be evaluated by its complexity. The

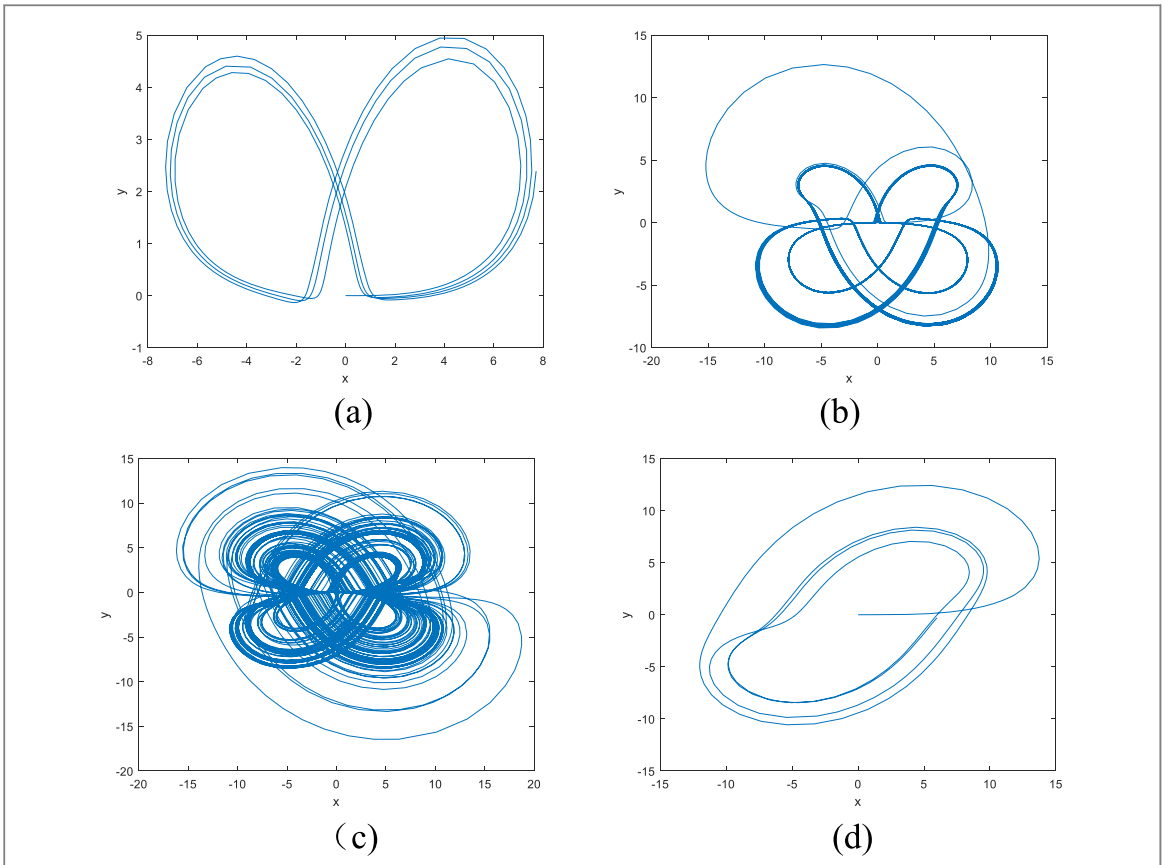

Figure 6. (a) Phase diagram with $a = 1.08$; (b) phase diagram with $a = 1.363$; (c) phase diagram with $a = 1.58$; (d) phase diagram with $a = 2.5$.

complexity can be analyzed in terms of both behavioral complexity and structural complexity. In the following, the complexity of the new system will be analyzed in terms of behavioral complexity of Spectral Entropy (*SE*) as well as structural complexity of C_0-algorithm, respectively.

The initial values of the system are kept as $x_0 = 0$, $y_0 = 0$, $z_0 = 0.001$, $w_0 = 0.001$, along with the parameters $a = 1.25$, $k = 1.25$, $m = 0.42$, $n = -0.5$, and $h = 0.5$, the complexity with respect to parameters b and c is analyzed and shown in figure 11. The image's color depth reveals how complicated the chaotic system is in this range of parameters. Higher levels of system complexity are presented by darker colors, while lighter colors are associated with lower levels of complexity.

3.7. NIST statistical test

The National Institute of Standards and Technology (NIST) test is an authoritative tool for pseudo-random testing. The test results are measured using *P*-value, when the *P*-value is greater than or equal to 0.01, it indicates that the chaotic sequence passes the test and has a strong randomness. The initial values of the system are kept as $x_0 = 0$, $y_0 = 0$, $z_0 = 0.001$, $w_0 = 0.001$, along with the parameters $a = 1.25$, $b = 5.35$, $k = 1.25$, $m = 0.42$, $n = -0.5$, and $h = 0.5$, the randomness of the system is tested at $c = 3.04$ and $c = 3.792$, respectively, and the test results are obtained and shown in table 2. From the NIST test results, it can be seen that the chaotic sequence passes the NIST test at $c = 3.34$. Therefore, the new system proposed in this paper generates chaotic sequences in chaotic state with strong randomness, which is very suitable for image encryption.

4. Encryption scheme

4.1. Encryption algorithm

4.1.1. Bit plane decomposition

For a grayscale image, the grayscale value can be represented by multiple bits, in which the planes composed of the bits at the same position of grayscale values are called bit planes. For a grayscale image with a gray value between 0 and 255, the gray value can be represented by 8-bits, and after the bit plane decomposition, 8-bit planes are obtained, and different bit planes carry different amounts of information about the original image. Taking a 256×256 grayscale image as an example, the grayscale image can be decomposed into 8-bit levels

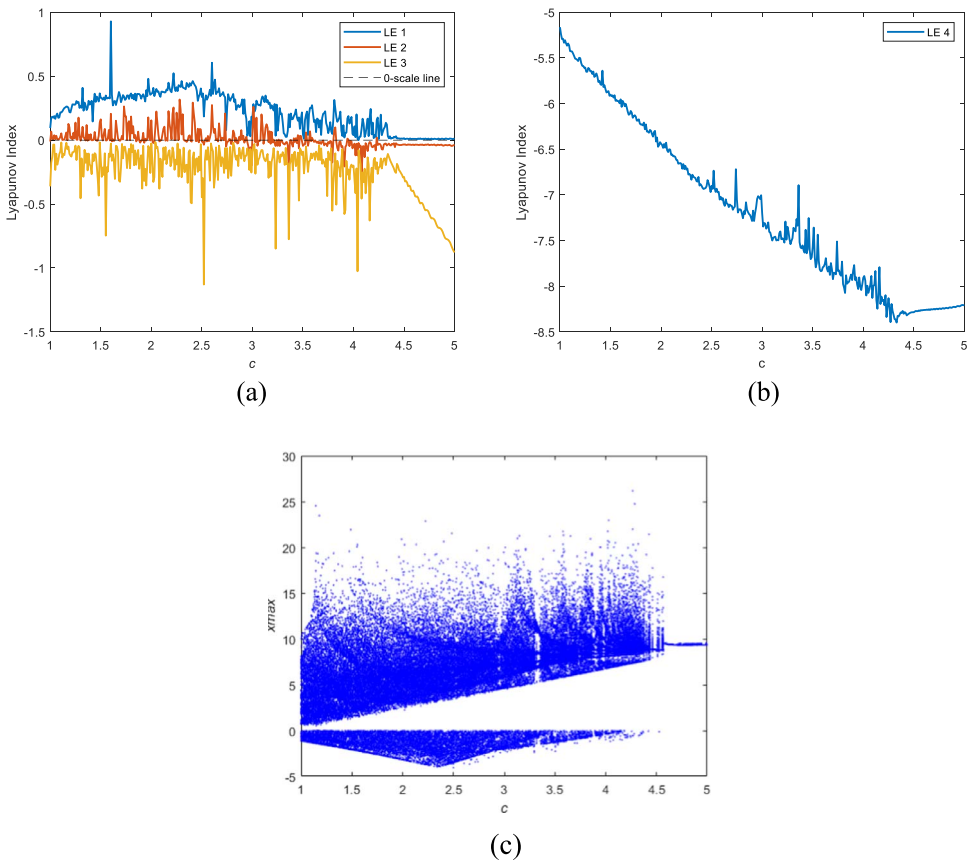

Figure 7. (a) Lyapunov exponents spectrum of LE_1, LE_2 and LE_3 with $c \in (1, 5)$; (b) Lyapunov exponents spectrum of LE_4 with $c \in (1, 5)$; (c) bifurcation diagram with $c \in (1, 5)$.

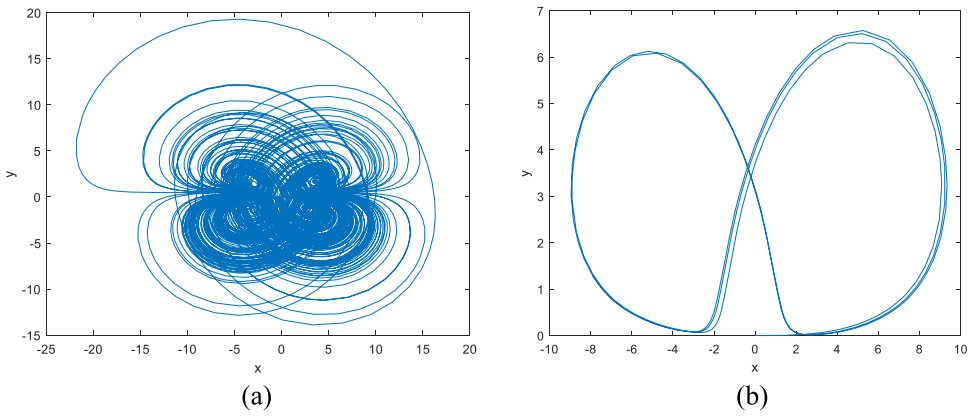

Figure 8. (a) Phase diagram with $c = 2.25$; (b) phase diagram with $c = 5$.

according to equation (7). As shown in figure 12, In order to gain a deeper understanding of the bit plane decomposition of the grayscale image, the pixels are decomposed into eight binary matrices, as illustrated in figure 13, this example demonstrates the process for an image of size 3×3.

$$I_n(i, j) = \frac{I(i, j)}{2^{n-1}} \bmod 2,\ n = 1, 2 \cdots, 8,\ I_n(i, j) \in [0, 1] \tag{7}$$

Where each bit plane contains the information of one bit in the plaintext image. $I_4 I_3 I_2 I_1$ denotes the least significant bits planes. $I_8 I_7 I_6 I_5$ denotes the most significant bits planes. The quantity of visual information in each bit plane is different and arranged in ascending sequence from I_1 to I_8.

As can be seen in figure 12, from the 8th bit plane to the 5th bit plane, as the number of bits decreases, the outline of the image becomes less and less clear, the sharpness is decreasing, but it still shows the general outline

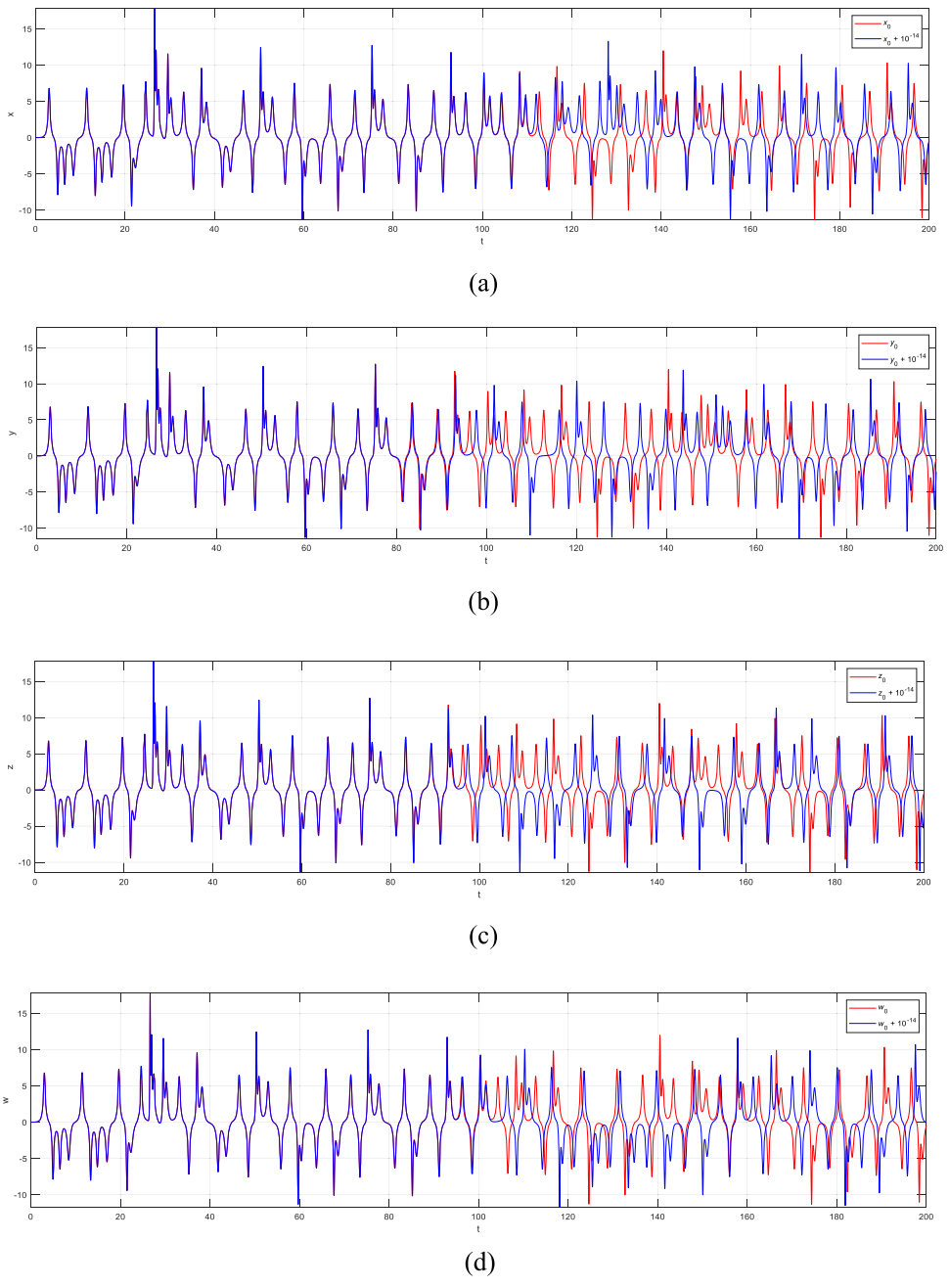

Figure 9. curve of state variables over time: (a) curve of state variable x_0 over time, (b) curve of state variable y_0 over time, (c) curve of state variable z_0 over time, (d) curve of state variable w_0 over time.

of the image. However, the information in the image is not clearly visible in the 4th to 1st planes. This results due to the high 4-bit plane holds more than 94% of the information of the original image.

4.1.2. S-shaped disorder

Since the 8th bit plane to the 5th bit plane after the bit plane decomposition has more than 94% information of the original image, so only the higher four planes $I_8I_7I_6I_5$ need to be disrupted, and this also increases the efficiency of our encryption. Firstly, the high 4-bit plane obtained from the bit plane decomposition is represented by a matrix respectively, which is scanned from the upper left corner to the right according to the S-type, and arranged into a one-dimensional matrix according to the scanning order, and then according to the value of the chaotic sequence, the scanned one-dimensional matrix is traversed into the matrix with the same size as that of the original matrix, and an element is selected from the leftmost side of the one-dimensional matrix when the first value of the chaotic sequence is 1, and an element is selected from the rightmost side of the one-dimensional matrix when the first value of the chaotic sequence is 0, and the bit plane $I_8'I_7'I_6'I_5'$ is obtained

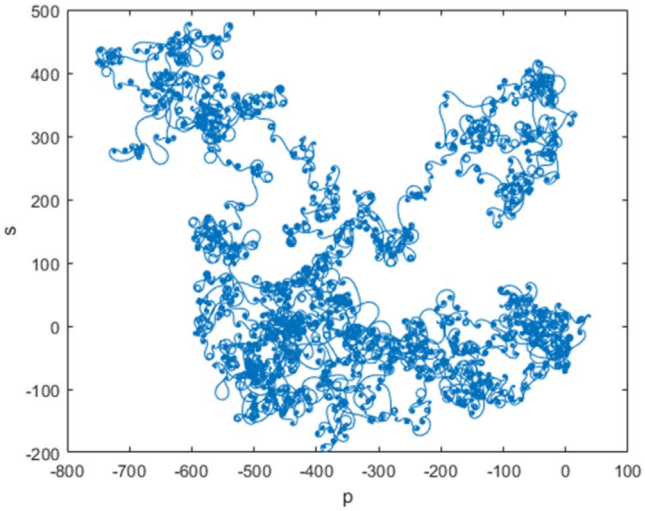

Figure 10. The 0–1 portraits of the proposed system (1).

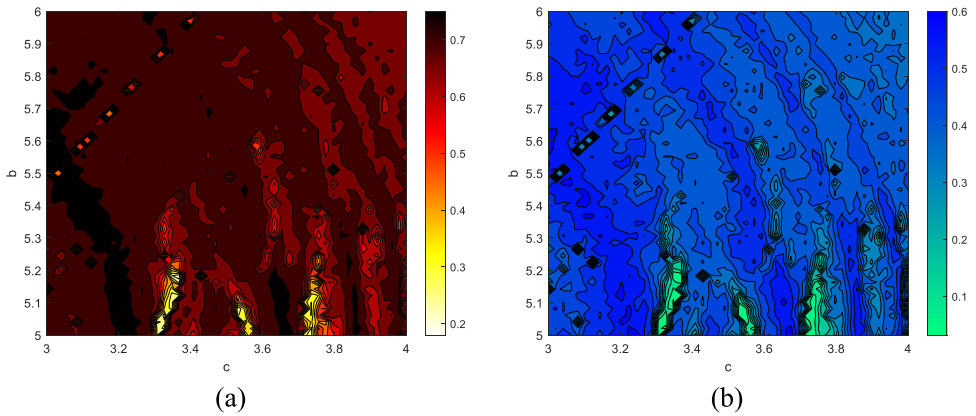

Figure 11. Complexity analysis with respect to parameters b and c. (a) SE complexity; (b) C_0–algorithm complexity.

by traversing according to the values of the chaotic sequence in turn. Figure 14 shows the transition process of the S-type chaotic state.

4.2. Secret keys generation

The key structure of the cryptosystem is divided into two parts, the first part is given by the user of the cryptosystem and the other part is calculated from the Secure Hash Algorithm (SHA-256) function of the original image. A slight alteration to the original image yields an entirely new key value since SHA-256 is extremely sensitive to the starting value. Therefore, we can use the hash value of the plaintext image to generate the key for the encryption algorithm, making the algorithm resistant to both known-plaintext attacks and chosen-plaintext attacks. The image using the SHA-256 function to obtain a 256-bit key, which is divided into one piece with every eight bits and can be represented as $k_1, k_2, k_3 \ldots, k_{32}$. Then the initial value can be obtained as follows:

$$\begin{cases} x_0 = \mathrm{mod}(sum(k_1, k_2, k_3, k_4)/256, 1) + x_0' \\ y_0 = \mathrm{mod}(sum(k_5, k_6, k_7, k_8)/256, 1) + y_0' \\ z_0 = \mathrm{mod}(sum(k_9, k_{10}, k_{11}, k_{12})/256, 1) + z_0' \\ w_0 = \mathrm{mod}(sum(k_{13}, k_{14}, k_{15}, k_{16})/256, 1) + u_0' \end{cases} \tag{8}$$

Table 2. NIST statistical test result.

NO.	Statistical test	Reference *P*-value	$c = 3.34$	
			P-value	Results
01	Frequency	⩾0.01	0.374825	Pass
02	Block frequency	⩾0.01	0.871329	Pass
03	Runs	⩾0.01	0.452431	Pass
04	Longest run	⩾0.01	0.724837	Pass
05	Rank	⩾0.01	0.137835	Pass
06	Discrete Fourier Transform	⩾0.01	0.276243	Pass
07	Overlapping template	⩾0.01	0.483282	Pass
08	Non-overlapping template	⩾0.01	0.364128	Pass
09	Universal	⩾0.01	0.783475	Pass
10	Linear complexity	⩾0.01	0.589761	Pass
11	Approximate entropy	⩾0.01	0.563489	Pass
12	Cumulative sums	⩾0.01	0.328495	Pass
13	Serial	⩾0.01	0.779384	Pass
14	Random excursions	⩾0.01	0.649327	Pass
15	Random excursions variant	⩾0.01	0.134872	Pass

$$\begin{cases} a = \mathrm{mod}(sum(k_{17}, k_{18}, k_{19}, k_{20})/256, 1) + a' \\ b = \mathrm{mod}(sum(k_{21}, k_{22}, k_{23}, k_{24})/256, 5) + b' \\ c = \mathrm{mod}(sum(k_{25}, k_{26}, k_{27}, k_{28})/256, 3) + c' \\ h = \mathrm{mod}(sum(k_{29}, k_{30}, k_{31}, k_{32})/256, 1) + h' \end{cases} \tag{9}$$

Where, x_0, y_0, z_0, w_0 are the initial values of the meminductor multi-wing chaotic system, a, b, c, h are the initial parameters, x_0', y_0', z_0', w_0' and a′, b′, c′, h′ are the custom key values. The parameters k, m, n of the meminductor multi-wing chaotic system are used as fixed keys.

4.3. Encryption scheme

Step 1: Input a plaintext image P of size M × N.

Step 2: Firstly, the initial key is entered to perform an iterative operation on the chaotic system and remove the first 1500 terms to make the system have better randomness. The four chaotic sequences of $\{x_i, y_i, z_i, w_i\}$ are obtained from the calculation are.

Step 3: Extract the 8-bit planes of the plaintext image according to equation (7) $I_8 I_7 I_6 I_5 I_4 I_3 I_2 I_1$. With bit plane decomposition, more than 94% information from the original image is retained from the 8th bit plane to the 5th bit plane, so only the high four planes $I_8 I_7 I_6 I_5$ need to be disrupted. Firstly, the higher 4-bit planes from the bit plane decomposition are represented by the matrix, the chaotic sequence x_i, y_i is then transformed into the 01 sequence according to equation (10) and equation (11). Next, the chaotic sequence x_i is used to control the planes 5 and 7, the chaotic sequence y_i used to control the planes 6 and 8, if the value of the chaotic sequence is 1, the first unselected element is selected from the beginning. If the value of the chaotic sequence is 0, the last unselected element is selected first from the end. Then traversing in turn in a matrix of the same size as the original matrix, hence bit plane $I_8' I_7' I_6' I_5'$ is obtained.

$$x_i = \mathrm{mod}(floor(x_i \times 10^{14}), 2) \tag{10}$$

$$y_i = \mathrm{mod}(floor(y_i \times 10^{14}), 2) \tag{11}$$

Step 4: In order to increase the randomness of the chaotic sequence, so the chaotic sequences z_i and w_i are multiplied to get a new chaotic sequence z_i'. The chaotic sequence z_i' is then used to generate a random sequence R' of length M × N according to equation (12), with values in the sequence ranging from 0 to 1, M × N is the size of the original plaintext image. Then the modulo operation according to equation (13) ensures that every value in R' is between 0 and 255. Then R' is obtained by multiplying the small number in R' by the large integer θ according to equation (14), converting it to a positive integer, and removing the fractional part. Finally, the random sequence R' is converted to a two-dimensional matrix R as the random matrix.

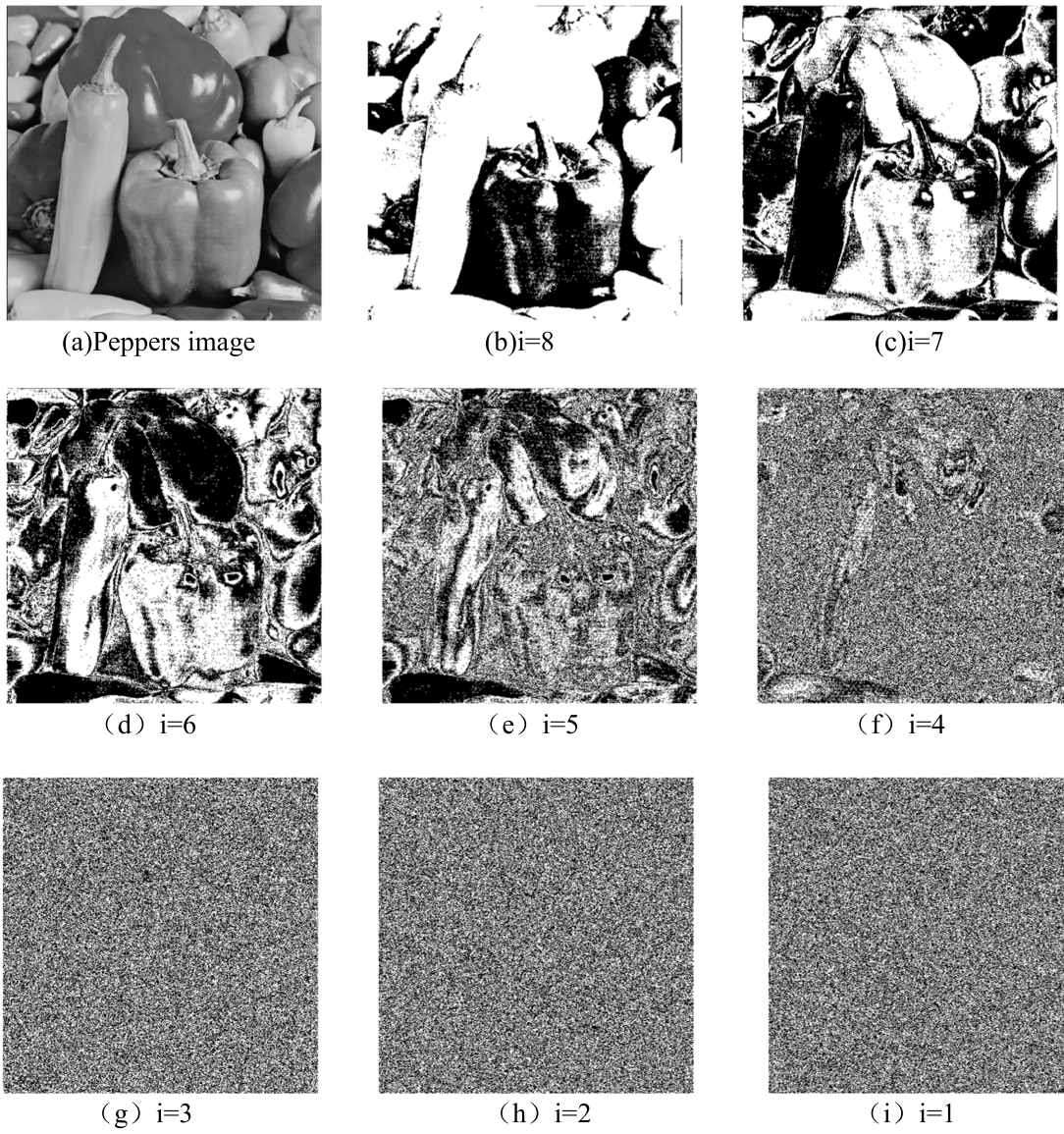

Figure 12. (a) Original gray scale image; (b)–(i) bit plane decomposition from high to low.

$$\begin{cases} R_i' = \{z_1', z_2', z_3', \ldots, z_{M\times N}'\} \\ R_i' = \max(0, \min(1, R_i')) \end{cases} \tag{12}$$

$$R(i) = R'(i) \bmod 256,\ i \in [1, mn] \tag{13}$$

$$R''(i) = \theta \times R(i) (i = 1, 2, \ldots, M \times N) \tag{14}$$

Step 5: The 8-bit planes R_8, $R_7, \cdots, R_1$ of the random image R are extracted according to equation (7). Up to this point, there are 16 different binary bit planes: I_8', I_7', I_6', I_5', I_4, I_3, I_2, I_1 as well as R_8, $R_7, \cdots, R_1$.

Step 6: Take XOR operation on the corresponding bit planes according to equation (15).

$$C_i = \begin{cases} I_i \oplus R_i,\ i = 1, \cdots, 4; \\ I' \oplus R_i,\ i = 5, \cdots, 8; \end{cases} \tag{15}$$

Step 7: The 8-bit planes after diffusion are merged according to equation (16) to obtain the final ciphertext image C, Block diagram of the encryption scheme as shown in figure 15.

$$C = \sum_{i=1}^{8} C_i \times 2^{i-1} = C_1 + 2 \times C_2 + 2^2 \times C_3 + \ldots + 2^7 \times C_8 \tag{16}$$

4.4. Decryption program

Step 1: The input key is iterated according to the chaotic system to obtain the chaotic sequence and extract the 8-bit planes of the cipher image.

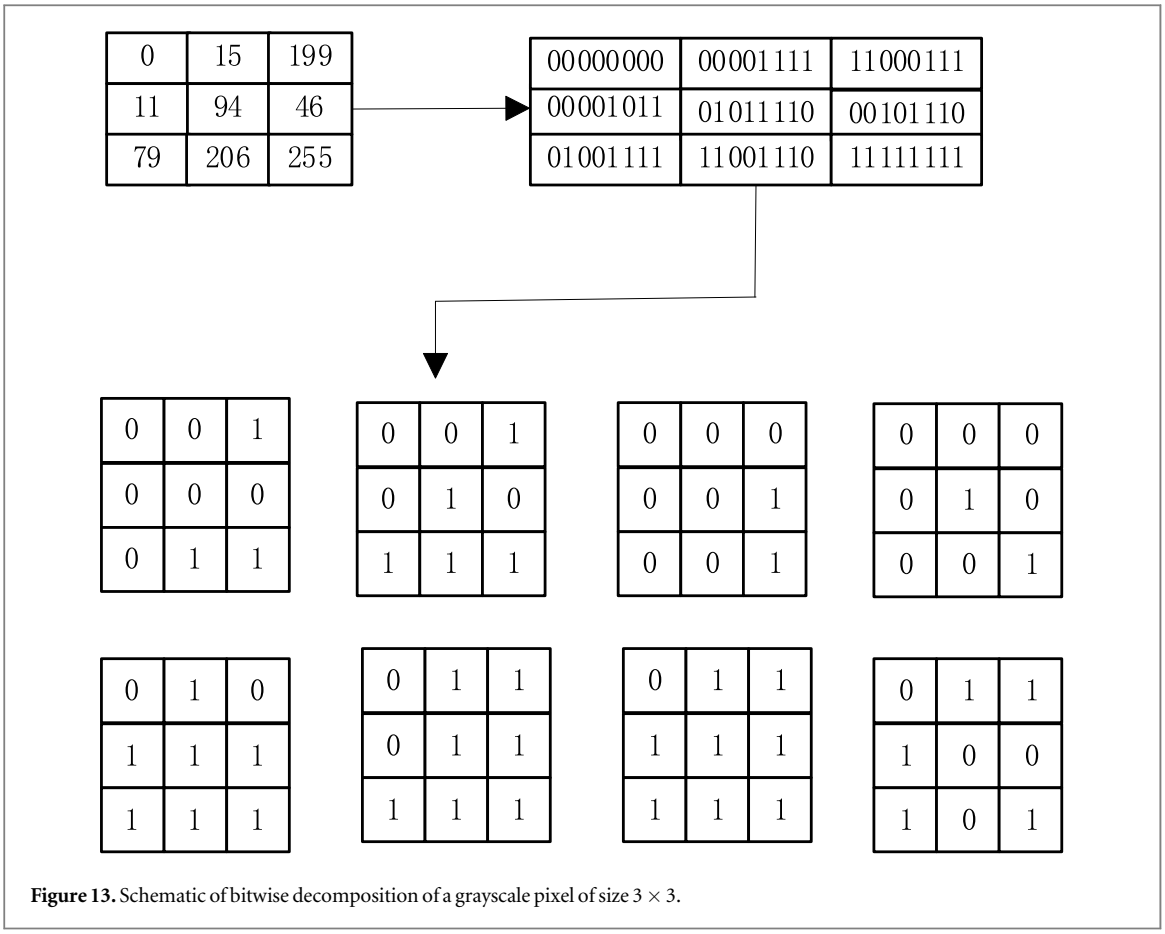

Figure 13. Schematic of bitwise decomposition of a grayscale pixel of size 3 × 3.

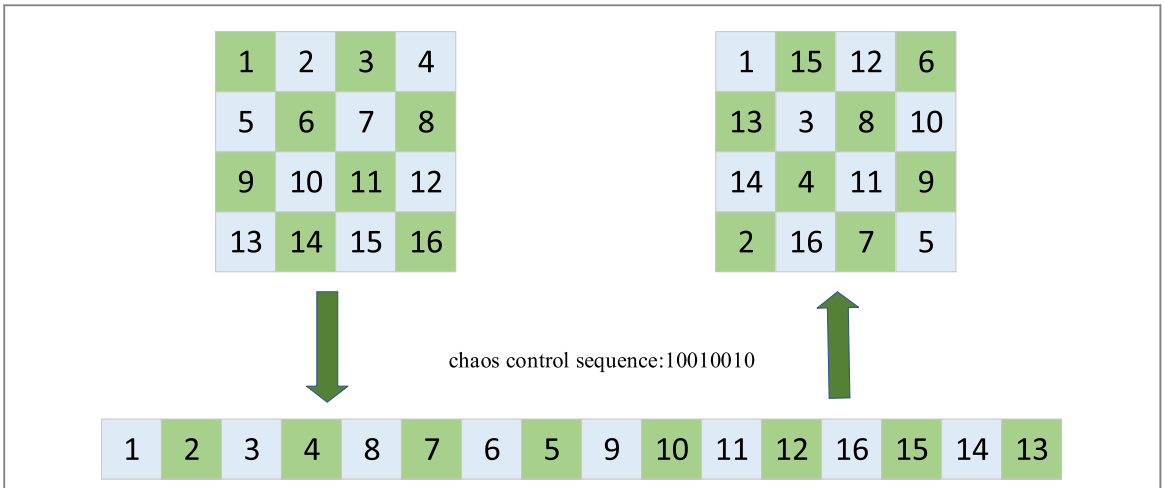

Figure 14. S-type scrambling.

Step 2: The chaotic sequences z_i, w_i are multiplied to obtain a new chaotic sequence z_i', The chaotic sequence z_i' is then used to generate a random sequence R' of length M × N according to equation (12), with values in the sequence ranging from 0 to 1, Then the modulo operation according to equation (13) ensures that every value in R' is between 0 and 255. Then R' is obtained by multiplying the small number in R' with the large integer θ according to equation (14), converting it to a positive integer, and removing the fractional part. Finally, the random sequence R is converted into a two-dimensional matrix as the image B.

Step 3: Decompose the image B into 8-bit planes.

Step 4: Decompose the ciphertext image into 8-bit planes, and then carry out XOR with the 8-bit planes decomposed in step 3 to obtain the 8-bit planes after XOR operation.

Step 5: The higher four planes after the XOR are transformed into matrices, and the chaotic sequences x_i and y_i are transformed into 01 sequences, in which the chaotic sequence x_i is used to control the planes 5 and 7, and

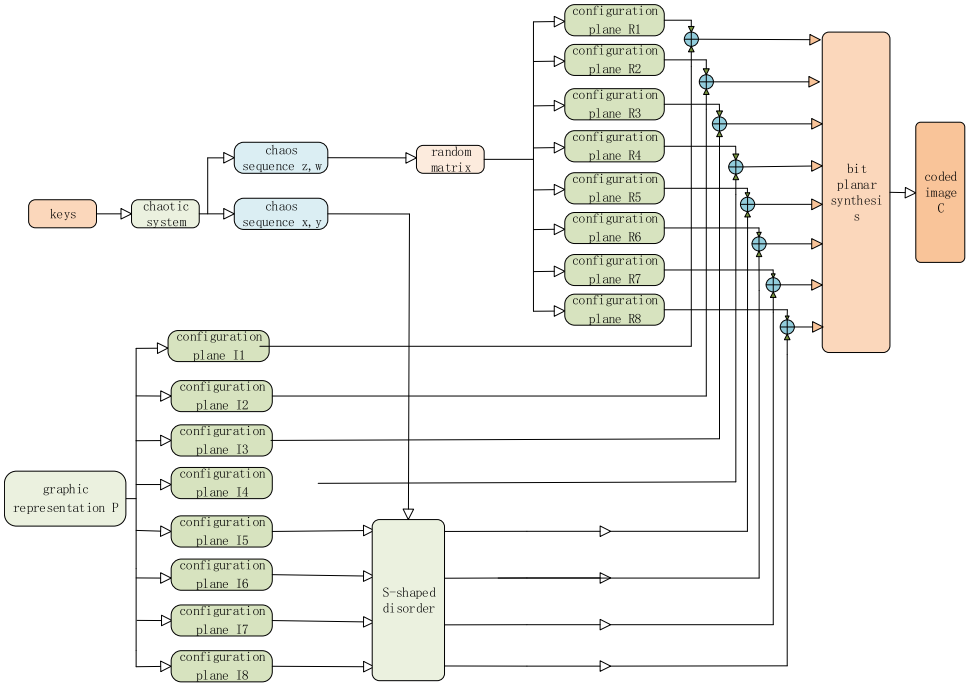

Figure 15. Block diagram of the encryption scheme.

the chaotic sequence y_i is used to control the planes 6 and 8. If the first value of the chaotic sequence is 1, the first untraversed element is placed on the leftmost side of the one-dimensional matrix, if the first value of the chaotic sequence is 0, the first untraversed element is placed on the rightmost side of the one-dimensional matrix, if the second value of the chaotic sequence is 1, the second untraversed element is placed on the second left-hand side of the one-dimensional matrix, if the second value of the chaotic sequence is 0, the second untraversed element is placed on the second right-hand side of the one-dimensional matrix, and so on until all elements of the matrix have been traversed.

Step 6: A new 4-bit plane is obtained by traversing the one-dimensional matrix in a matrix of the same size as the ciphertext matrix, respectively, according to the inverse S-type.

Step 7: The new 4-bit planes are obtained and merged with the lower 4-bit planes obtained from step 4 to get the plaintext image.

5. The evaluation of the proposed encryption algorithm's performance

5.1. Experimental results

The results of the 256 × 256 Baboon image, House image and the Tree image, the 512 × 512 Peppers image, Boat image and the Plane image are displayed in figure 16. The ciphertext image produced by the encryption algorithm have characteristic of snowflake noise style, which making it impossible to directly extract the information of the plaintext, the image obtained after decryption with the legal key is identical to the plaintext image. This results shows that the proposed image encryption scheme has good encryption and decryption effects.

5.2. Histogram analysis

The pixel distribution of an image can be displayed by its histogram, which counts the number of pixels at each intensity level. It should also not be statistically similar to the original image, meaning that the histogram of the encrypted image should not yield any information about the original image. The histograms of the encrypted image of 'Baboon', 'House', 'Tree' with sieze of 256 × 256 and 'Peppers', 'Boat', 'Plane' with size of 512 × 512 are uniformly distributed, as shown in figure 17. These results indicate that the encryption algorithm is able to perfectly hide the details of the plaintext image, and the details of the original image can not be obtained by performing statistical attacks on the ciphertext image.

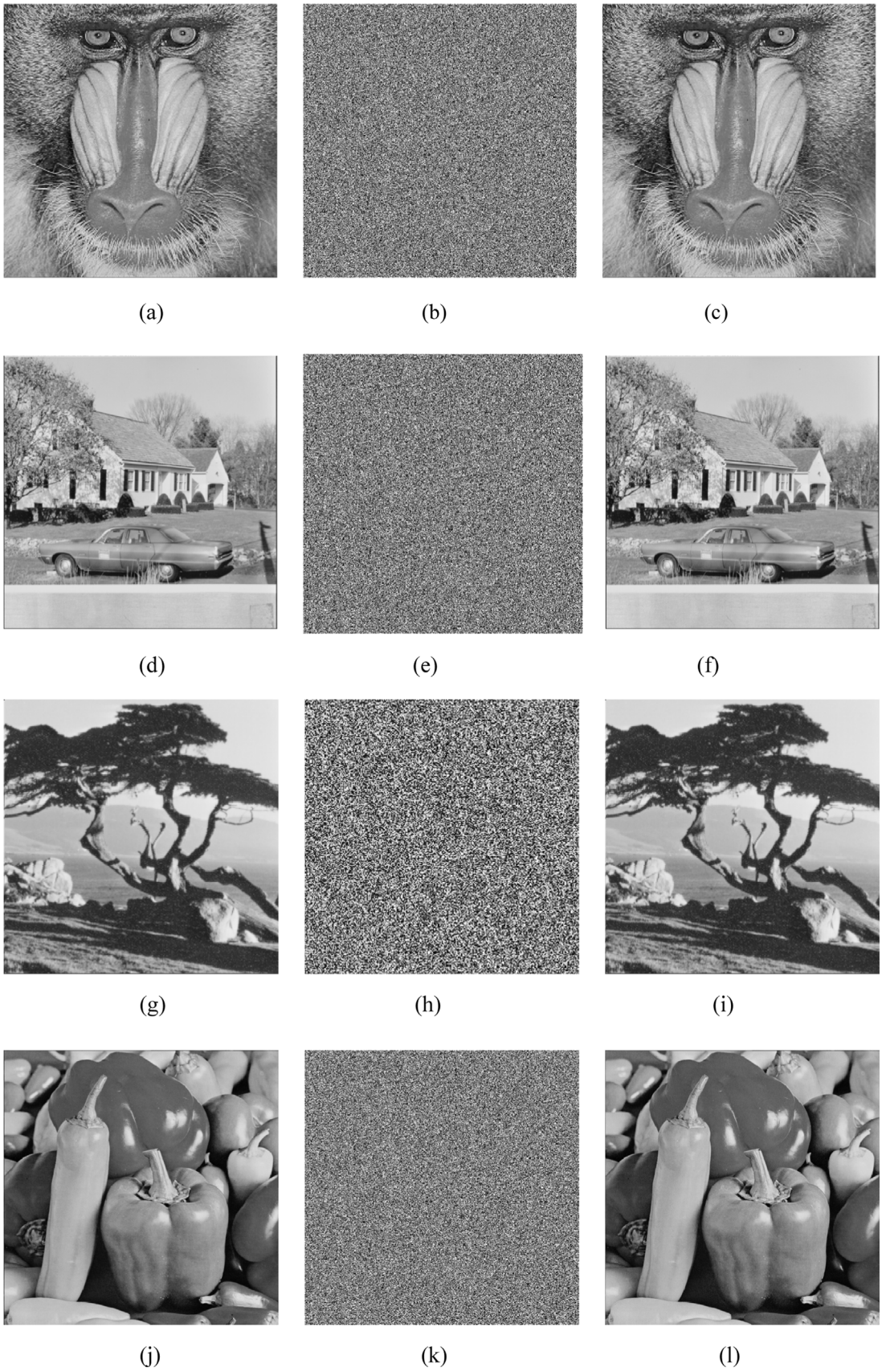

Figure 16. Encrypted and decrypted images: (a)–(c) Baboon; (d)–(f) House; (g)–(i) Tree; (j)–(l) Peppers; (m)–(o) Boat; (p)–(r) Plane.

5.3. Information entropy analysis

One way to evaluate the efficacy of encryption is to use information entropy, which can measure the degree of information disorder. Information entropy of an image is a measurement of how cluttered the pixel

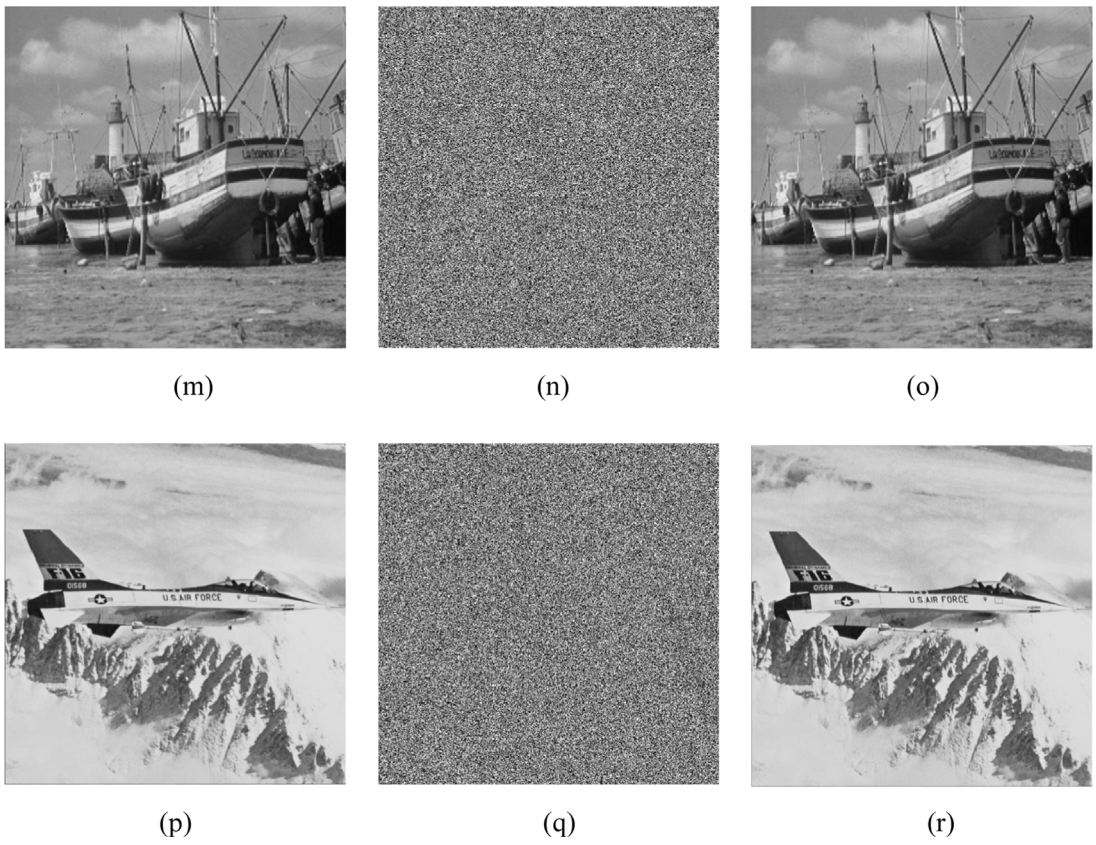

Figure 16. (Continued.)

information. The larger the value, the more cluttered the image is and the less information it contains. The value of information entropy can be calculated by equation (17).

$$G(s) = -\sum_{i=0}^{2^L} p(x_i)\log_2 p(x_i) \tag{17}$$

Where $G(s)$ shows the randomness of message source (s), $p(x_i)$ denotes the probability of the pixel value at that point in the whole image, and 2^L denotes all the states of the pixel value in the whole image. If the information entropy value of an image with grayscale value from 0 to 255 is closer to 8, there is more disorder. Table 3 displays the information entropy of various test images. It can be seen that the information entropy of the encrypted image is increased and close to the ideal value of 8, which demonstrates that this image encryption system can effectively resist the attack based on the Information entropy. The information entropy values for Peppers under various encryption techniques are displayed in table 4. This algorithm's information entropy value is closer to the ideal value than previous image encryption techniques, and the cipher image's irregularity is higher. Thus, the encryption system offers higher security.

5.4. Correlation analysis

Plaintext images are easily attacked due to the high correlation between neighboring pixels, a good encryption algorithm has to break this correlation completely. The neighboring pixels of plaintext image are highly correlated vertically, horizontally, and diagonally, which are susceptible to statistical attacks by attackers. A good ciphertext image has very weak correlation between neighboring pixels, and the correlation among neighboring pixels can be used to judge the performance of the encryption algorithm. The correlation coefficient can be computed using the following formula.

$$R_{xy} = \frac{\mathrm{cov}(x,\, y)}{\sqrt{D(x)D(y)}} \tag{18}$$

$$D(x) = \frac{1}{S_n}\sum_{i=1}^{S_n}(x_i - E(x_i))^2 \tag{19}$$

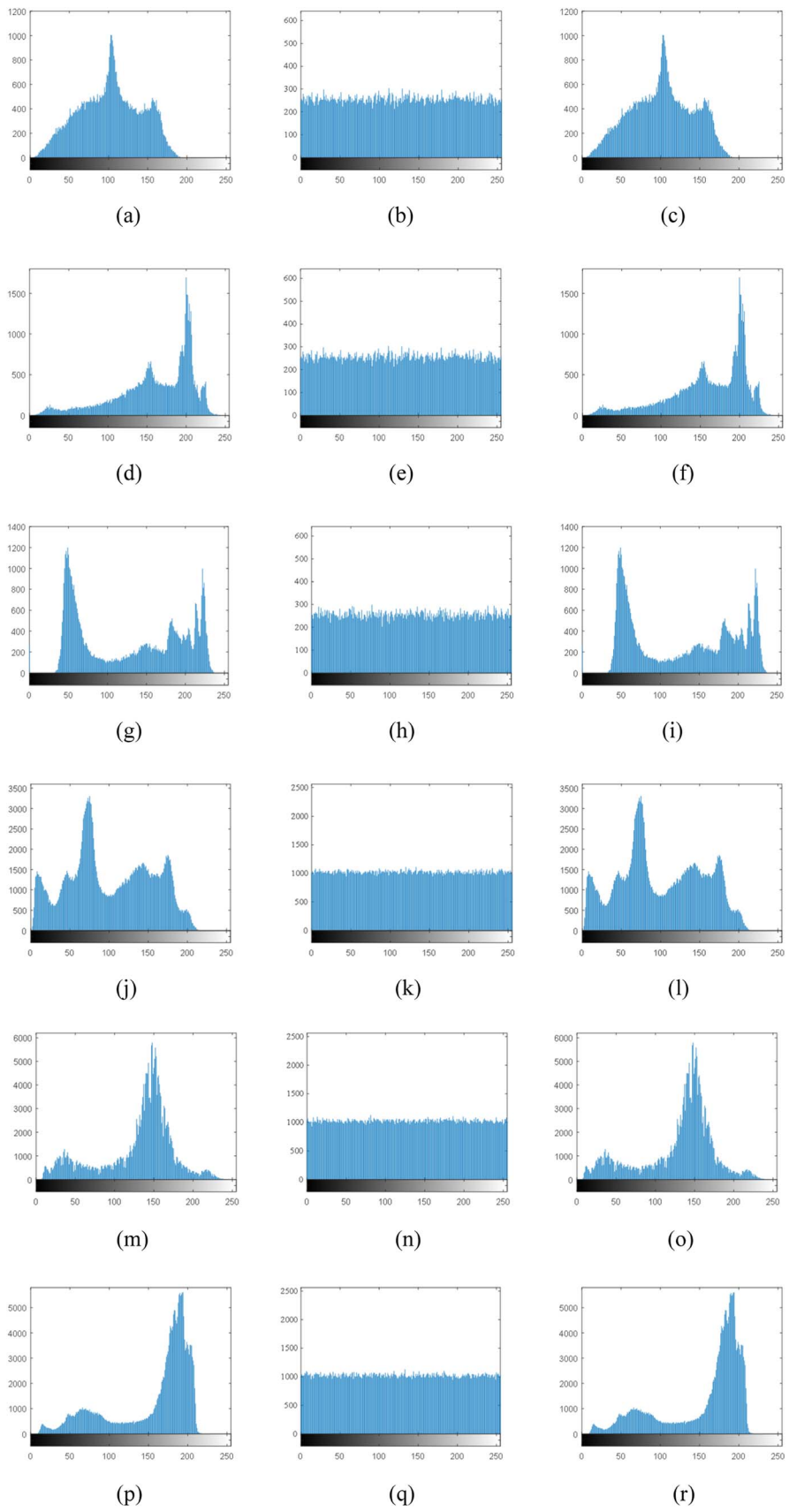

Figure 17. The histogram results of plaintext, encrypted and ciphertext image: (a)–(c) Baboon; (d)–(f) House; (g)–(i) Tree; (j)–(l) Peppers; (m)–(o) Boat; (p)–(r) Plane.

$$E(x) = \frac{1}{S_n}\sum_{i=1}^{S_n} x_i \tag{20}$$

Table 3. Information entropy analysis of six images.

Images	Plaintexts	Ciphertexts
Baboon	7.7623	7.9976
House	7.4858	7.9973
Tree	7.4678	7.9974
Peppers	7.6698	7.9994
Boat	7.5392	7.9993
Plane	7.5412	7.9993

Table 4. Information entropy for Peppers image and comparison with other existing algorithms.

Encryption Algorithms	Proposed	[32]	[33]	[34]	[35]	[36]
Information entropy	7.9994	7.9860	7.9994	7.9993	7.9987	7.9982

$$\mathrm{cov}(x, y) = \frac{1}{S_n}\sum_{i=1}^{S_n}(x_i - E(x_i))(y_i - E(y_i)) \tag{21}$$

For calculating the correlation coefficient, 3000 pairs of neighboring pixels are selected randomly from the ciphertext and plaintext images, and the correlation between neighboring pixels in the horizontal, vertical and diagonal directions of the image is calculated. The horizontal, vertical, and diagonal correlations between neighboring pixels in the plaintext image and the ciphertext image are depicted in figure 18 and its average values are given in table 5. The plaintext image's pixel correlations are obviously clustered along the diagonal, which indicating a high correlation, in contrast the ciphertext image's correlation span the whole space, with very low or even zero correlation coefficients. Furthermore, table 6 gives the correlation coefficients of the image of Peppers encrypted by various encryption techniques. The ciphertext image's correlation coefficient is nearly zero, and there is hardly correlation between adjacent pixels. In comparison to other methods, our suggested encryption method has a lower correlation coefficient, which suggests that it can break the link between pixels and defeat statistical attacks.

5.5. Key space analysis

A robust encryption algorithm should have a very large key space, as this determines how resistant the method is to brute force attacks. Utilizing $x(0)$, $y(0)$, $z(0)$, $w(0)$, a, b, c, and h as encryption keys, and assuming a computer calculation accuracy of 10^{-14}, The key space is: $(10^{14})^8 \approx 2^{375}$, which is significantly larger than the recommended value of 2^{100} in the cryptosystem. As shown in table 7, the key space of this algorithm is not the largest compared to other algorithms, but it is better than other algorithms in terms of information entropy and differential attacks, which are shown in tables 4 and 9, respectively. Furthermore, the key space of this method far exceeds the theoretical requirements, which means that the method can resistant brute force attacks.

5.6. Key sensitivity analysis

An essential indicator for assessing the security of image encryption is key sensitivity. The right key with a few small weak is used for the image decryption mechanism, figure 19 displays the sensitivity test results of different keys with minor changes. The findings of the experiment demonstrate that the encryption algorithm is extremely sensitive to its key, as even minor modifications of key used to the encrypted image, the encrypted image cannot be properly decrypted. Correct decryption can only be achieved when the correct key is utilized.

5.7. Differential attack analysis

In general, differential attack is implemented by comparing the encryption results of the original image and the modified image. It is frequently required to test an encryption system's sensitivity to plaintext in order to assess how strong resistant is it facing differential attacks. A good encryption scheme should guarantee that even little modifications to the plaintext leads to entirely distinct ciphertexts. The pixel difference between two images is often measured using the Number of Pixel Change Rate (NPCR) and Uniform Average Change Intensity (UACI), thus reflecting the ability of an encryption system to resist differential attacks. The calculation formulas are shown in equations (22)–(24).

$$NPCR = \sum_{i=0}^{M}\sum_{J=0}^{N}\frac{D(i, j)}{M \times N} \times 100\% \tag{22}$$

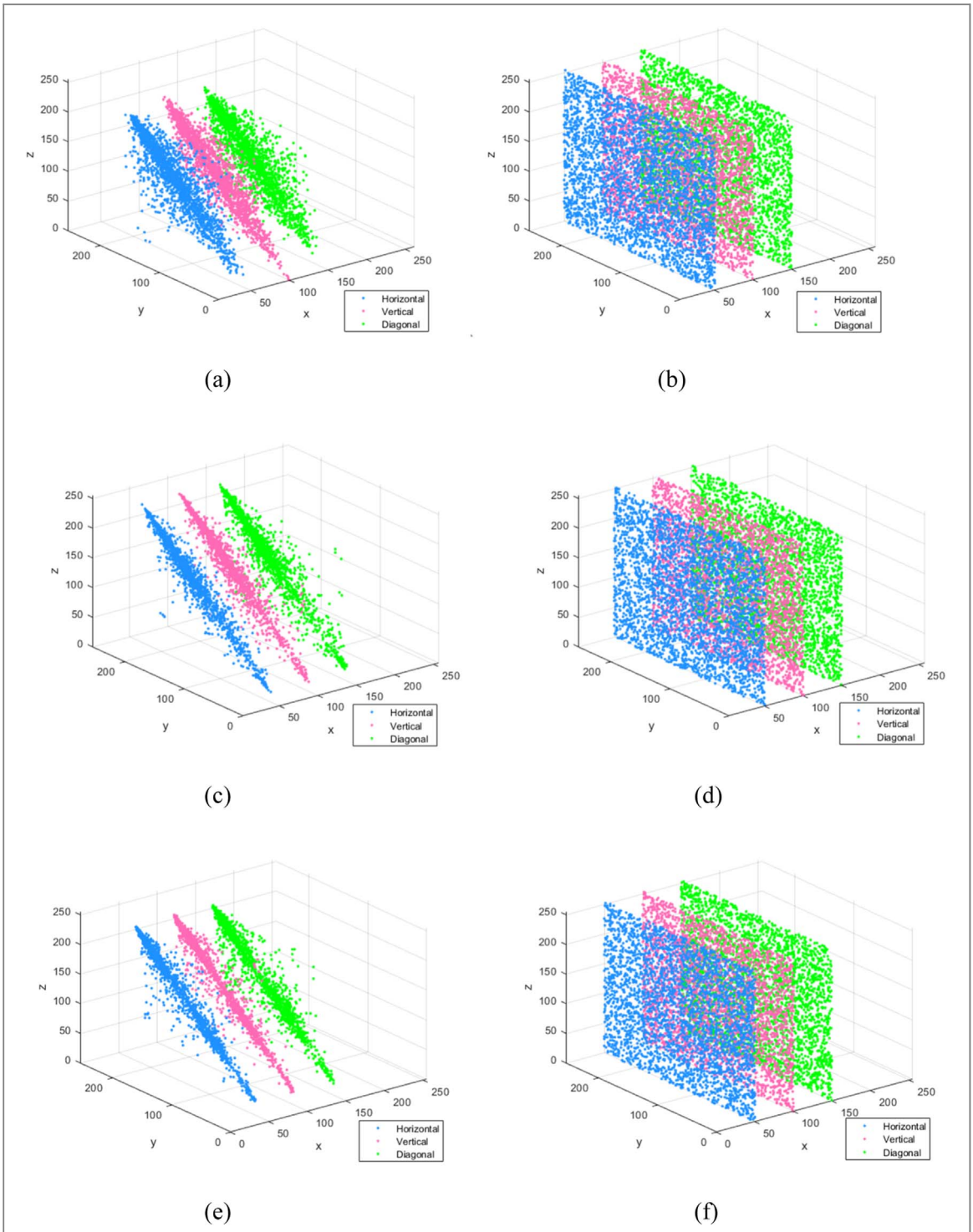

Figure 18. Correlation analysis results of plaintext images and ciphertext images. (a)–(b) Baboon; (c)–(d) House; (e)–(f) Tree; (g)–(h) Peppers; (i)–(j) Boat; (k)–(l) Plane.

$$D(i,j) = \begin{cases} 1, E1(i,j) \neq E2(i,j) \\ 0, E1(i,j) = E2(i,j) \end{cases} \tag{23}$$

$$UACI = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\frac{(|E_1(i,j) - E_2(i,j)|)}{255} \times 100\% \tag{24}$$

Where M and N denote the length and width of the image respectively, E_1 and E_2 denote the encrypted ciphertext image with only one pixel point different, and $D(i,j)$ denotes the pixel point where the two ciphertext images are different. The target UACI value is 33.4635% and the intended NPCR value is 99.6093% when the encryption technique satisfies the security requirements. In this experiment, we modify one bit of a pixel value at various locations in the plaintext images, encrypt these plaintext images using the same key, and calculate the

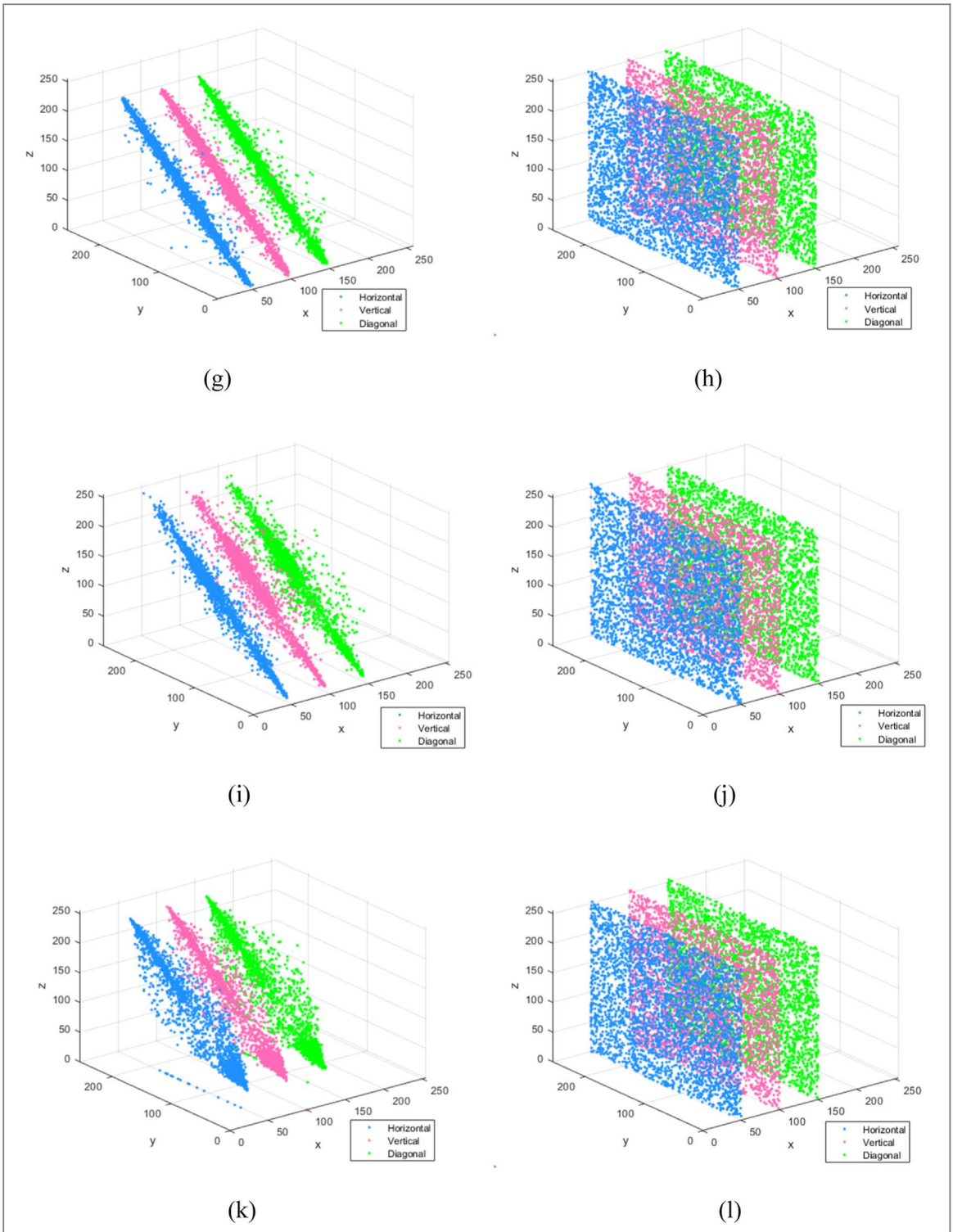

Figure 18. (Continued.)

corresponding values of NPCR and UACI for the ciphertext images in accordance with equations (22)–(24). Table 8 presents the values of NPCR and UACR calculated when a pixel value is altered. It can be observed that the NPCR and UACI for various images are nearly the desired level. In addition, table 9 shows the NPCR and UACI values of the proposed method and other methods on the image 'Peppers'. Therefore, it can be conclude that the scheme has good resistance to differential attacks.

5.8. Robustness analysis

During data transmission, external noise and interference may lead to information loss. Therefore, it is crucial to test the robustness of the new image encryption algorithm. This examination focuses on the ciphertext image's

Table 5. Correlation analysis of six samples.

Image size	Images	Plaintexts			Ciphertexts		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
256*256	Baboon	0.9005	0.8388	0.8119	0.0139	−0.0116	−0.0001
	House	0.9638	0.9260	0.8988	0.0208	−0.0142	−0.0163
	Tree	0.9647	0.9724	0.9681	−0.0032	−0.0067	−0.0174
512*512	Peppers	0.9844	0.9838	0.9776	−0.0039	−0.0078	−0.0021
	Boat	0.9374	0.9736	0.9183	0.0017	0.0039	−0.0025
	Plane	0.9614	0.9713	0.9219	−0.0119	−0.0026	−0.0013

Table 6. Comparison of correlation coefficients with other algorithms.

Algorithms	Horizontal	Vertical	Diagonal
Original image	0.9844	0.9838	0.9776
Proposed scheme	−0.0039	−0.0078	−0.0021
[33]	0.0030	0.0143	0.0114
[35]	−0.0075	−0.0071	−0.0042
[37]	−0.0033	0.0011	0.0070
[38]	−0.0052	0.0086	−0.0020
[39]	0.0001	0.0015	0.0078

Table 7. Comparison of key space sizes.

algorithms	Proposed	[32]	[33]	[34]	[35]	[36]
Key space	2^{375}	2^{511}	2^{398}	2^{213}	2^{512}	2^{963}

resistance to clipping and noise attacks. Through testing the resist performance at various noise levels, we discovered that the quality of encrypted images drops with noise levels rise.

In digital image encryption, the most likely interference during image transmission is noise. In order to test the ability to resist noise attacks, salt and pepper noise(SPN) with intensity of 1%, 5% and 10% are added to the ciphertext image. The encrypted images of Baboon and Peppers with different level of noise as example are shown in figures 20 and 21 respectively.

Based on the experimental results, we observe that even with varying degrees of noise interference in the encrypted image, it can still successfully recover the important information of the image. This indicates that our proposed encryption method is able to resist noise attacks to a certain extent. In particular, not only are these images susceptible to interference from various noise sources, resulting in information distortion or quality degradation, but there is also a risk of data loss due to some factors such as instability of the transmission link or physical loss of the storage medium. Therefore, the algorithm's ability to resist cropping attacks is evaluated, and its simulation results are illustrated in figures 22 and 23. By cropping the ciphertext image with varying degrees and evaluating its quality, the result is discovered that even in the face of a cropping assault, the decrypted image may retain the majority information of the plaintext image, which demonstrates the ability of our encryption technique against cropping assaults.

The Structural Similarity Index Measure (SSIM) and Peak Signal to Noise Ratio (PSNR) can be used to assess how similar the original and encrypted images are to one another. When the PSNR number is more than 30 dB, it is generally thought that the decrypted image is very close to the original plaintext image. The higher the PSNR value, the stronger the similarity between the two images. In addition, SSIM values close to 1 indicate that the decrypted image is highly similar to the original plaintext image.

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{25}$$

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[p_1(i,j) - p_2(i,j)]^2}{M \times N} \tag{26}$$

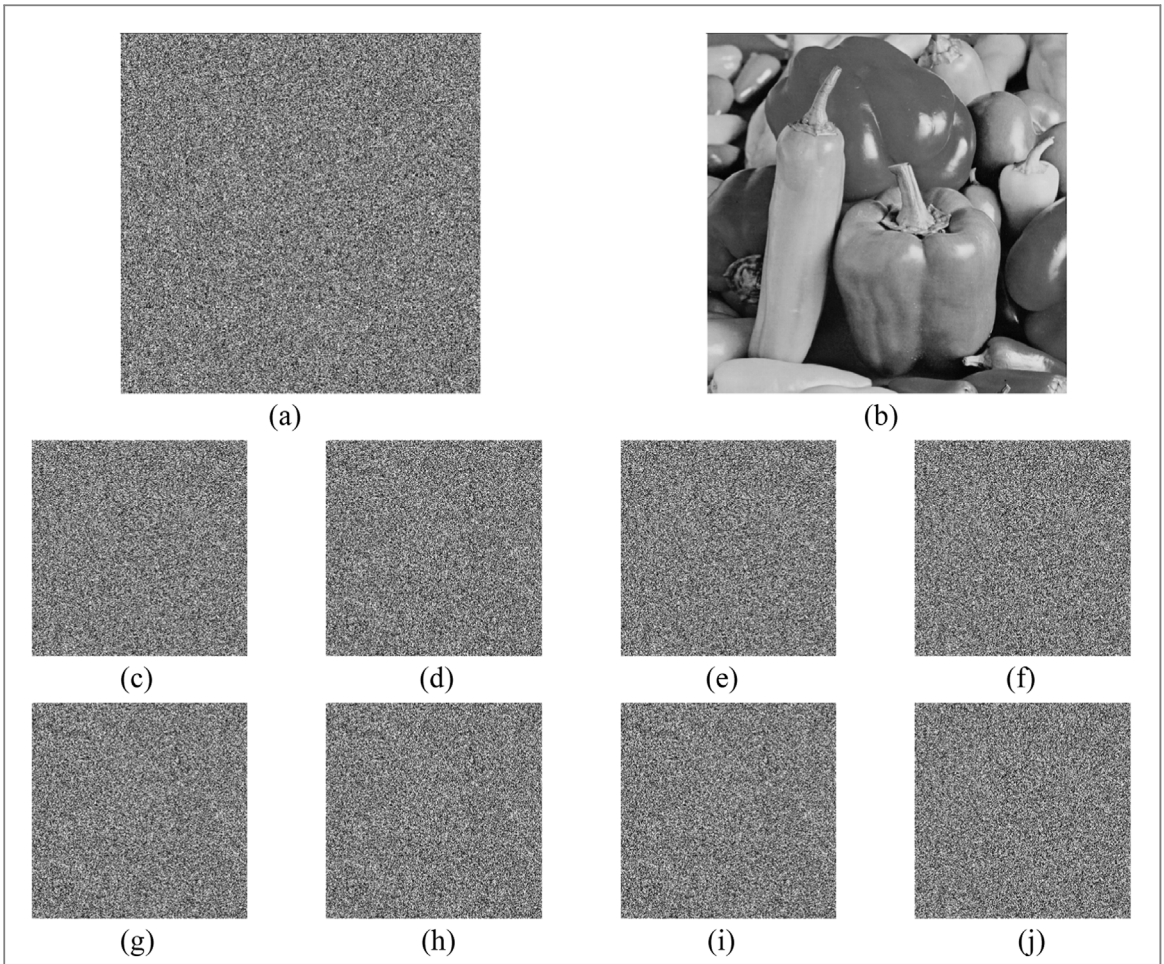

Figure 19. The decryption image obtained by decrypting ciphertext image of 'Peppers' after changing different key values. (a) ciphertext image; (b) decrypted image with correct keys; (c) $x_1(0) = x_1 + 10^{-14}$; (d) $y_1(0) = y_1 + 10^{-14}$; (e) $z_1(0) = z_1 + 10^{-14}$; (f) $w_1(0) = w_1 + 10^{-14}$; (g) $a = 1.25 + 10^{-14}$; (h) $b = 5.35 + 10^{-14}$; (i) $c = 3.3 + 10^{-14}$; (j) $h = 0.5 + 10^{-14}$.

Table 8. The NPCR and UACI for different images.

Image size	Image	NPCR(%)	UACI(%)
256*256	Baboon	99.6143	33.3826
	House	99.6109	33.3843
	Tree	99.6107	33.3837
512*512	Peppers	99.6105	33.4373
	Boat	99.6108	33.4497
	Plane	99.6113	33.4462

Table 9. The NPCR and UACI of image Peppers with other algorithms.

Algorithms	Proposed	[32]	[33]	[34]	[36]	[40]
NPCR(%)	99.61	99.60	99.60	99.60	99.60	99.62
UACI(%)	33.43	33.49	30.41	32.44	33.59	33.59

$$SSIM = \frac{(2\bar{p}_1\bar{p}_2 + \delta_1)(2\sigma_{p_1}\sigma_{p_2} + \delta_2)}{(\bar{p}_1^2 + \bar{p}_2^2 + \delta_1)(\sigma_{p_1}^2 + \sigma_{p_2}^2 + \delta_2)} \tag{27}$$

In equation (26), M and N denote the height and width of the original plaintext image, and $p_1(i, j)$ and $p_2(i, j)$ represent the gray intensity values at point (i, j) in the original and decrypted images. In equation (27), $\delta_2 = (\mu_1 t)$, $\delta_2 = (\mu_2 t)$, $\mu_1 = 0.01$, $\mu_2 = 0.03$ $t = 255$. And $\sigma_{p_1}^2$ and $\sigma_{p_2}^2$ represent the variances of p_1 and p_2.

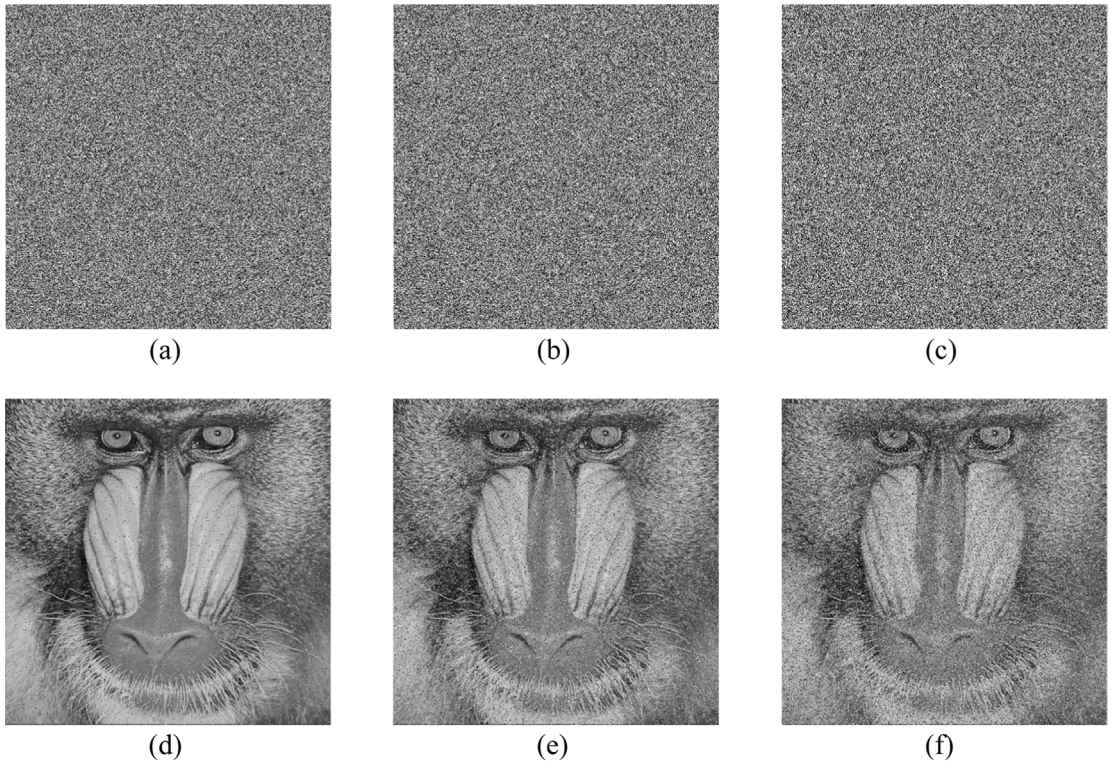

Figure 20. (a)–(c) 1%, 5%, 10% SPN encrypted Baboon image; (d)–(f) 1%, 5%, 10% SPN decrypted Baboon image.

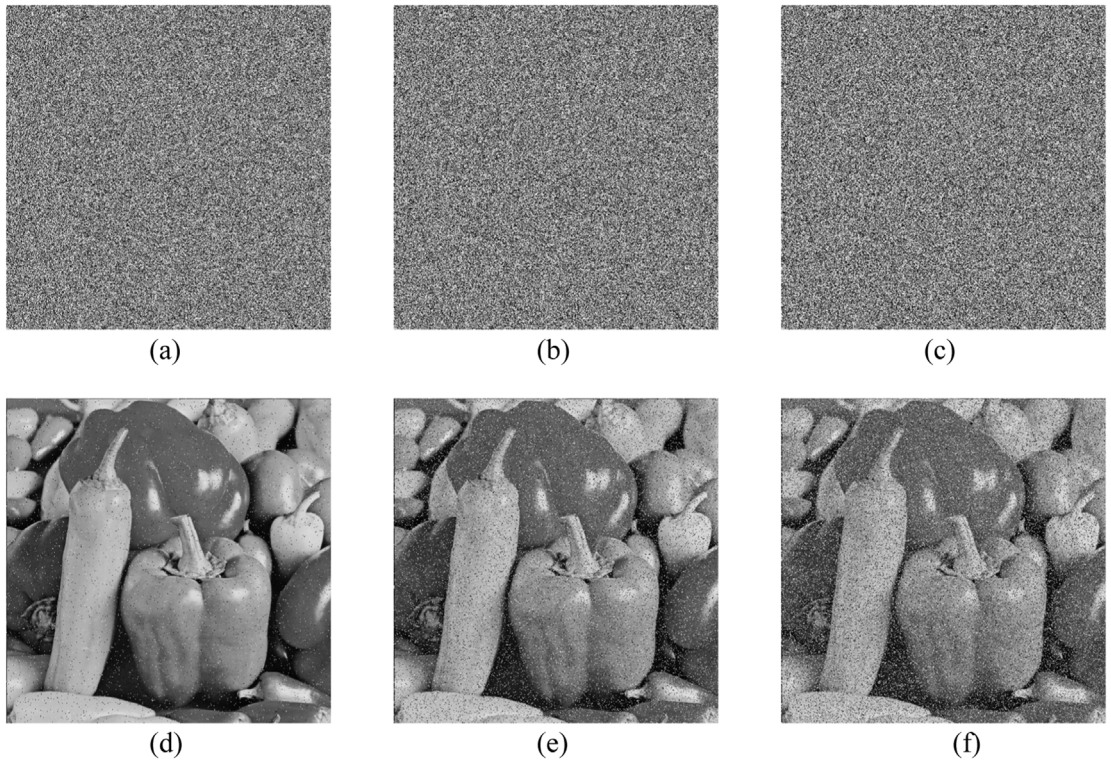

Figure 21. (a)–(c) 1%, 5%, 10% SPN encrypted Peppers image; (d)–(f) 1%, 5%, 10% SPN decrypted Peppers image.

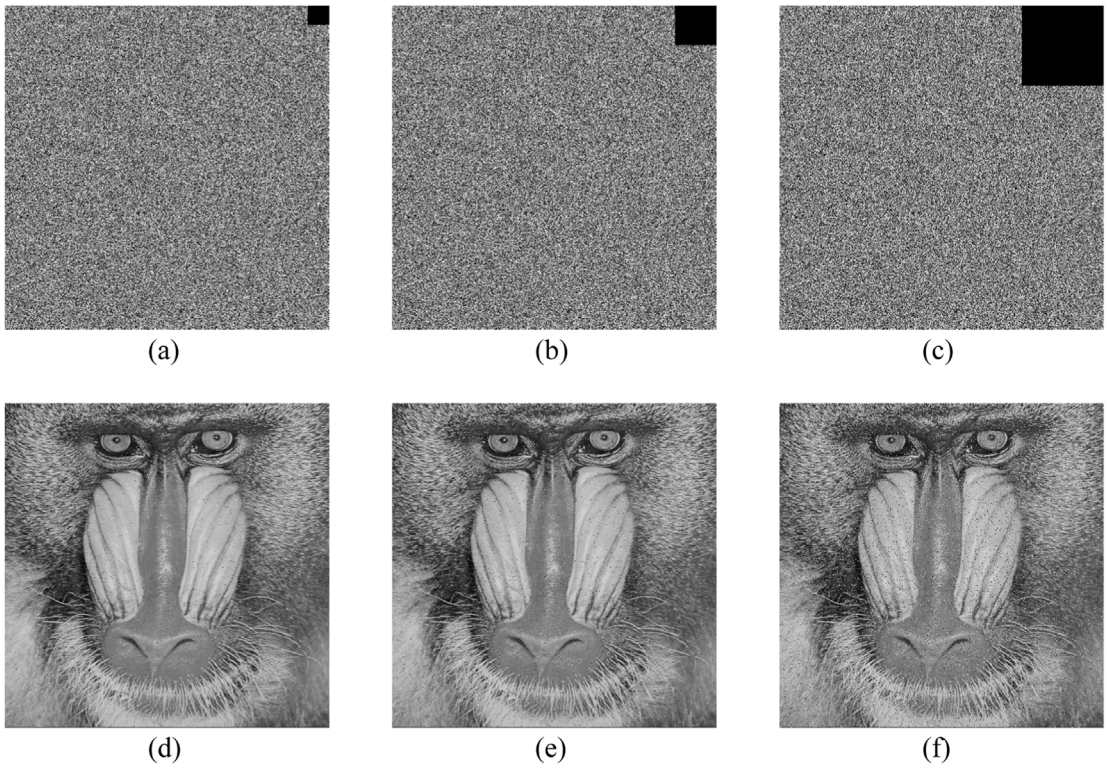

Figure 22. (a)–(c) Encrypted Baboon image with cropping size of 32*32, 64*64 and 128*128 respectively; (d)–(f) the corresponding decrypted image with cropping size of 32*32, 64*64 and 128*128 respectively.

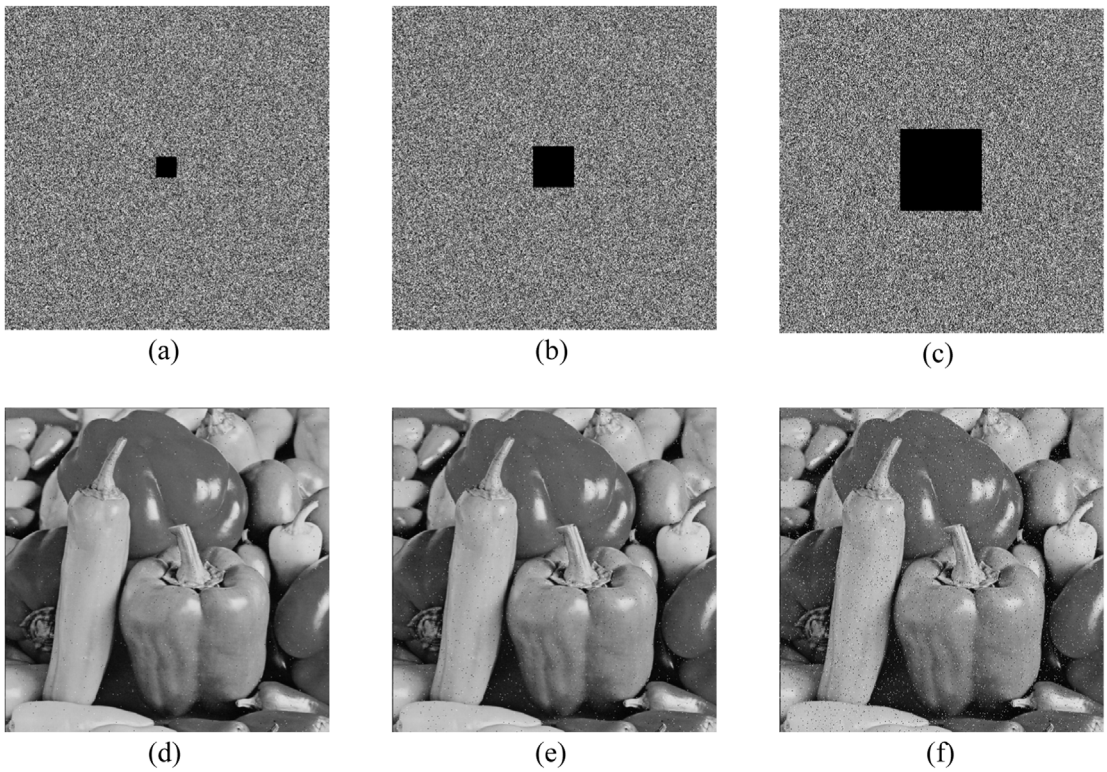

Figure 23. (a)–(c) Encrypted Peppers image with cropping size of 32*32, 64*64 and 128*128 respectively; (d)–(f) the corresponding decrypted image with cropping size of 32*32, 64*64 and 128*128 respectively.

Table 10. SSIM between the plain images and the decrypted images under the SPN and cropping attacks.

Test image	SPN level	Proposed	Cropping size	Proposed
	0.01	0.8979	32*32	0.9538
Baboon	0.05	0.6564	64*64	0.8472
	0.1	0.4928	128*128	0.6203
	0.01	0.6854	32*32	0.8349
House	0.05	0.4908	64*64	0.6672
	0.1	0.3238	128*128	0.3724
	0.01	0.7219	32*32	0.8629
Tree	0.05	0.5872	64*64	0.6417
	0.1	0.2184	128*128	0.3257
	0.01	0.7397	32*32	0.7928
Peppers	0.05	0.4098	64*64	0.6513
	0.1	0.2203	128*128	0.3813
	0.01	0.6928	32*32	0.7063
Boat	0.05	0.3276	64*64	0.5738
	0.1	0.2039	128*128	0.2847
	0.01	0.7924	32*32	0.7436
Plane	0.05	0.6349	64*64	0.5283
	0.1	0.2492	128*128	0.2917

Table 11. PSNR between the plain images and the decrypted images under the SPN and cropping attacks.

Test image	SPN level	Proposed	Cropping size	Proposed
	0.01	38.23	32*32	42.61
Baboon	0.05	31.94	64*64	38.09
	0.1	29.64	128*128	31.73
	0.01	38.17	32*32	43.17
House	0.05	31.97	64*64	36.92
	0.1	28.17	128*128	31.67
	0.01	36.49	32*32	40.68
Tree	0.05	31.32	64*64	37.89
	0.1	28.37	128*128	30, 62
	0.01	39.83	32*32	43.95
Peppers	0.05	33.31	64*64	38.01
	0.1	30.49	128*128	33.35
	0.01	38.59	32*32	42.69
Boat	0.05	32.47	64*64	37.18
	0.1	29.73	128*128	32.64
	0.01	38.28	32*32	41.39
Plane	0.05	32.84	64*64	37.46
	0.1	29.48	128*128	32.47

Table 12. Comparison of the capability to resist Noise and cropping attacks.

Attack Type	Attack Intensity	PSNR			
		[41]	[42]	[43]	Proposed
	0.0001	31.56	28.18	33.44	38.79
SPN	0.0003	30.22	28.18	33.26	33.47
	0.0005	30.02	28.17	33.02	31.58
	32*32	27.21	30.18	24.92	43.95
Cropping	64*64	—	—	—	38.01
	128*128	—	—	—	33.35

Table 13. Encryption and decryption speed.

Image size	Image	Encryption time (s)	Decryption time (s)
256*256	Baboon	0.5782	0.6158
	House	0.4873	0.5017
	Tree	0.4987	0.5126
512*512	Peppers	1.6876	1.9754
	Boat	1.7638	1.8973
	Plane	1.7593	1.9716

Table 14. Runing time of image Peppers compared to different algorithms.

Algorithm	Encryption time (s)	Decryption time (s)
Proposed	1.68	1.97
[39]	2.78	2.42
[44]	8.60	9.63
[45]	16.43	16.43
[46]	5.16	3.10

Table 15 Comparison of time complexity of different algorithms.

Algorithm	Confusion	Diffusion
Proposed	$\Theta(4MN)$	$\Theta(8MN)$
[47]	$\Theta(5MN)$	$\Theta(8MN)$
[48]	$\Theta(8MN \times \log(8MN))$	$\Theta(MN)$
[49]	$\Theta(MN \times \log(MN))+\Theta(4MN \times \log(4MN))$	$\Theta(4MN)$

The ability of this image encryption algorithm to resist noise attack and cropping attack is tested using different images and the simulation results are shown in tables 10 and 11. Table 12 displays the comparison results of the image encryption technique with other algorithms. The technique presented in this study has strong robustness and withstand attacks such noise interference and data cropping.

5.9. Encryption and decryption efficiency evaluation

An efficient algorithm should accomplish image encryption in the shortest possible time. Table 13 displays the duration required for encryption and decryption of images with varying size. The data in table 14 presents a comparative analysis of the encryption and decryption durations of this algorithm against similar algorithms.

5.10. Computational complexity analysis

Algorithm complexity is an important measure of the efficiency of an algorithm's execution, computational complexity and time efficiency are often used to evaluate the complexity of an algorithm. Computational complexity is analyzed on the basis of time, considering only the encryption part, the time complexity of the designed algorithm is measured, for image encryption with input size $M \times N$, the computational complexity is $\Theta(4M \times N)$ at the confusion phase; and the computational complexity at diffusion phase is $\Theta(8M \times N)$, so that the total computational complexity of the encryption algorithm is $\Theta(12M \times N)$. And the time complexity of the proposed algorithm is compared with the time complexity of the existing techniques as shown in table 15, for MN images, the proposed algorithm provides less computational cost in contrast to the existing algorithms.

6. Conclusions

In this paper, a multi-wing chaotic system based on a meminductor model is proposed, the variation of the parameters of the new system can cause the system to produce two-wing, three-wing and four-wing chaotic attractors. Firstly, we analyze the proposed multi-wing chaotic system based on meminductor model, and the results show that the proposed chaotic system has good chaotic characteristics and is highly sensitive to the system parameters. Secondly, an image encryption algorithm is designed by using the random sequence

generated by the chaotic system. In the design of the image encryption algorithm, firstly, the plaintext image is processed to get the grayscale pixel value matrix, then the grayscale pixel value matrix is decomposed into 8-bit planes by bit plane decomposition, after that the high 4-bit planes are disordered by S-type permutation, afterwards a random image is generated by using chaotic sequences and decomposed into 8-bit planes by bit plane decomposition. Then use the chaotic sequence to generate a random image and carry out bit plane decomposition into 8-bit planes, The decomposed 8-bit planes are performed XOR operation with the high four-bit plane after S-type scrambling and the low four-bit plane without scrambling. The bitplanes after XOR operation are merged to obtain the final encrypted image. Furthermore, the encryption algorithm of the new chaotic system is simulated and analyzed in numerical simulation and security analyses. Based on the simulation results and performance evaluation, the new image encryption scheme has the features of large key space, high key sensitivity, high complexity and high security. At the same time, the encryption scheme can resist a certain degree of noise attack and cropping attack.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Future works

In this paper, the dynamic behavior of the proposed multi-wing chaotic system based on the meminductor model as well as its application in image encryption are analyzed theoretically. In addition, the feasibility of circuit realization and image encryption is verified by Multisim and Matlab software. In the future, image encryption could be implemented on FPGA.

Competing interests

The authors declare no conflict of interest.

Authors contributions

Conceptualization, P F Ding; methodology P F Ding; software, P.F Ding and W W Hu; validation, P F Ding, L. Yang and W W Hu; formal analysis, P F Ding, W W Hu. and P H Geng; writing original draft, P F Ding and W W Hu; writing-review and editing, W W Hu, P H Geng and L Yang. All authors have read and agreed to the published version of the manuscript.

ORCID iDs

Pengfei Ding https://orcid.org/0000-0002-3297-4438
Weiwei Hu https://orcid.org/0009-0006-9270-6693

References

[1] Chua L O 1971 *Circuit Theory* **18** 507–19
[2] Strukov D B *et al* 2008 *Nature* **453** 80–3
[3] Ventra M D, Pershin Y V and Chua L O 2009 *Proc. IEEE* **97** 1717–24
[4] Tarasova V V and Tarasov V E 2017 *Solitons &Fractals.* **95** 84–91
[5] Dong C 2022 *Fractal Fract* **6** 190
[6] Vaidyanathan S *et al* 2018 *Eur. Phys. J. C Part Fields* **133** 46–50
[7] Fu Y D *et al* 2018 *Nonlinear Dyn.* **94** 1949–59
[8] Lv Z, Sun F and Cai C 2022 *Nonlinear Dyn.* **109** 3133–44
[9] Lorenz E N 1963 *Eur. Phys. J. C Part Fields* **20** 130–41
[10] Li T Y and Yorke J A 1975 *Am. Math. Mon* **82** 985–92
[11] Itoh M and Chua L O 2008 *Int. J. Bifurcat Chaos* **18** 3183–206
[12] Buscarino A *et al* 2013 *ECCTD* 1–4
[13] Pershin Y V and Ventra M D 2010 *Electron. Lett.* **46** 517–8
[14] Biolek D 2011 *Analog Integ. Circ. S* **66** 129–37
[15] Ventra M D 2011 *Adv. Phys.* **60** 145–227
[16] Biolek D, Biolek Z and Biolkova V 2011 *Electron. Lett.* **47** 1385–7
[17] Praksh S J, Krishna R B and Wei Z C 2018 *Chinese Phys.* B **27** 214–27
[18] Zhou L, Wang C H and Zou L L 2017 *Int. J. Bifurcat Chaos* **27** 1750027

[19] Wang Z et al 2016 Optik 127 2424–2431
[20] Sahoo S and Roy B K 2022 Chaos, Solitons Fractals 164 112598
[21] Cui L et al 2020 Chaos, Solitons Fractals 138 109894
[22] Sahoo S and Roy B K 2022 Chaos, Solitons Fractals 157 111926
[23] Li Y, Li Z J and Ma M L 2020 Multimedia Tools Appl. 79 29161–77
[24] Xie Q and Zeng Y 2020 Eur. Phys. J. Spec. Top. 229 1361–71
[25] Lin H et al 2021 IEEE Trans. Ind. Electron. 68 12708–19
[26] Ding P F, Li K and Wang Z X 2024 Phys. Scr. 99 045221
[27] Wang X et al 2023 Nonlinear Dyn. 111 14513–36
[28] Xin J, Hu H and Zheng J 2023 Nonlinear Dyn. 111 7859–82
[29] Zhu Z L et al 2011 Inform Sciences 181 1171–86
[30] Lin T and Wang X Y 2012 Opt. Commun. 285 4048–54
[31] Yuan F, Wang G Y and Jin P P 2015 Acta Phys. Sin. 64 214–26
[32] Lei R Q and Liu L F 2024 Phys. Scr. 99 075202
[33] Ding P F, Zhu J G and Zhang J 2024 Phys. Scr. 99 105211
[34] Chen W H et al 2023 Phys. Scr. 98 075515
[35] Kumar S and Sharma D 2024 Intell. Rev. 57 1–31
[36] Jiang X et al 2023 Nonlinear Dyn. 111 15531–55
[37] Ullah A, Shah A A and Khan J S 2022 Research Article 16 5680357
[38] Iqbal N, Hanif M, Abbas S and Khan 2021 J. Inf. Sec. Appl. 58 102809
[39] Liang Q and Zhu C X 2023 Opt. Laser Technol. 160 109033
[40] Khalil N, Sarhan A and Alshewimy M A 2021 Opt. Laser Technol. 143 107326
[41] Zhu L et al 2020 Signal Process. 175 107629
[42] Zhu L et al 2022 J. Inf. Sci. 607 1001–22
[43] Wang H et al 2019 Signal Process 155 218–32
[44] Kang X and Guo Z 2020 Signal Process. Image Commun. 80 115670
[45] Jithin K C and Sankar S 2020 J. Inf. Secur. Appl. 50 102428
[46] Dong Y, Huang X and Ye G 2021 Secur. Commun. Netw 2021 1–16
[47] Liu D D et al 2018 Signal Process 151 130–43
[48] Zhou Y C, Cao W J and Chen C L P 2014 Signal Process 100 197–207
[49] Teng L and Wang X Y 2012 Opt. Commun. 285 4048–54