Journal of Systems Engineering and Electronics
Vol. 33, No. 3, June 2022, pp.534 – 550

Image encryption based on a novel memristive chaotic system, Grain-128a algorithm and dynamic pixel masking

HUANG Lilian [1,2], SUN Yi [1,2], XIANG Jianhong [1,2,*], and WANG Linyu [1,2]

1. College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China;
2. Key Laboratory of Advanced Marine Communication and Information Technology-Ministry of Industry and Information Technology, Harbin Engineering University, Harbin 150001, China

Abstract: In this paper, we first propose a memristive chaotic system and implement it by circuit simulation. The chaotic dynamics and various attractors are analysed by using phase portrait, bifurcation diagram, and Lyapunov exponents. In particular, the system has robust chaos in a wide parameter range and the initial value space, which is favourable to the security communication application. Consequently, we further explore its application in image encryption and present a new scheme. Before image processing, the external key is protected by the Grain-128a algorithm and the initial values of the memristive system are updated with the plain image. We not only perform random pixel extraction and masking with the chaotic cipher, but also use them as control parameters for Brownian motion to obtain the permutation matrix. In addition, multiplication on the finite field GF(2^n) is added to further enhance the cryptography. Finally, the simulation results verify that the proposed image encryption scheme has better performance and higher security, which can effectively resist various attacks.

Keywords: memristive chaotic system, super-wide parameter range, image encryption, Grain-128a algorithm, dynamic pixel masking.

DOI: 10.23919/JSEE.2022.000053

1. Introduction

Since the 21st century, with the explosive development of mobile Internet and the continuous progress in technologies such as artificial intelligence and digital images have become one of the most popular information carrier in human social activities. The use of digital images has been extended beyond social communication to a variety of fields such as medical, financial, and even military. Due to the serious security threats to the sending, transmission and reception of information, high requirements are placed on the confidentiality of images. Images bear enormous amounts of data, strong correlation between adjacent pixels, and lots of redundancy, therefore, classical encryption algorithms perform unsuitably and ineffecti vely when encrypting images.

In the late 20th century, Matthews [1] and Fridrich [2] applied the chaos theory to text encryption and image encryption. The studies of chaos have revealed its extreme sensitivity to initial conditions and the unpredictability of its dynamical behaviour, which allows fast and flexible generation of satisfactory pseudo-random sequences for encryption. Researchers since then became interested in chaotic cryptography and have produced many multidisciplinary achievements in combination with modern science, including modern mathematics [3−6], bioscience [7−9], and quantum science [10].

Chaotic systems have the advantages of ideal randomness, superior confidentiality, and rich key changeability in cryptography, so it is a hot topic to explore chaotic systems that are more suitable for image encryption algorithms. Natiq et al. [11] proposed a new hyperchaotic map based on the sine map and the Henon map and designed a row-column permutation operation on the basis of this novel two-dimensional Sine-Henon alteration model. Wang et al. [12] introduced a new logistic modulation map derived from logistic map and piecewise linear chaotic map (PWLCM), named LP map. Wang et al.[12] further designed a pseudo-random coupled LP map lattices (PCLML) spatiotemporal chaos in pseudo-random coupling method based on the LP map for image encryption. It was experimentally proved that the LP chaotic map and the PCLML model with strong bifurcation and the security of this encryption algorithm had a good performance. Liu et al. [13] mitigated the dynamical degradation of the complex hyper chaotic Lü chaotic system by injecting impulse into control parameter. This randomness

Manuscript received March 09, 2021.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61203004), the Natural Science Foundation of Heilongjiang Province (F201220), and the Heilongjiang Provincial Natural Science Foundation of Joint Guidance Project (LH2020F022).

enhanced chaotic system combined with modulo and circshift operations to obtain a color image encryption scheme.

Since the first successful fabrication of memristor in 2008 [14], researchers have found that memristor-based chaotic systems are not only equipped with the good properties of ordinary chaotic systems, but also have a richer dynamical behaviour. Thus, memristive chaotic systems are also used in image encryption algorithms. Peng et al. [15] discussed that if the Chua diode in the Chua's circuit was replaced by a memristor, the derived chaotic system had coexisting attractors with multistability, and Peng et al. further encrypted the image with the simple XOR operation. Li et al. [16] introduced an image encryption algorithm based on a simple memristive chaotic system and dynamic deoxyribonucleic acid (DNA) operation. They implemented complex dynamical behaviors via simple circuits, including antimonotonicity, multistability and transient chaos.

By observation, chaotic systems typically used in image encryption algorithms have several common problems. One is that the range of initial values and parameters for chaotic maps is quite small, making it difficult to extend the flexibility of the key. What is more, most chaotic systems also have a lot of periodic windows within the available parameter range. Due to the limited accuracy of computers, unfavourable occurrences of losing the chaotic effect may happen.

To solve the above problems, we take advantage of the specific non-linearity of the memristor to construct a memristive chaotic system with an ultra-wide range of parameters and analyse its dynamics. Further, the outstanding performance of the proposed chaotic system is applied to an image encryption algorithm. In this scheme, not only the Grain-128a algorithm is used to update the key, but also pixel masking methods such as dynamic pixel confusion and chaos-based generation of Brownian motion matrices are implemented. In the final simulation experiments, the scheme is proven to have excellent security.

The rest of this paper is divided into three parts. Section 2 gives the equation and dynamic behavior analysis of the new memristive chaotic system. The specific process and experimental simulations of the image encryption algorithm are displayed in Section 3, and conclusions are drawn in Section 4.

2. A novel memristive chaotic system with ultra-wide parameter range (UWPR-MCS)

2.1 Mathematical model

The memristor is one of the basic electronic components proposed by Chua et al. [17] to characterise the relationship between charge q and magnetic flux φ. We build a new chaotic system with the smooth quadratic piecewise flux-controlled memristor [18], and the φ−q characteristic W is given by

$$\begin{cases} q(\varphi) = -A\varphi + 0.5B\varphi^2 \mathrm{sgn}(\varphi) \\ W(\varphi) = -A + B|\varphi| \end{cases} \tag{1}$$

where A and B are positive internal parameters of the memristor, and sgn(·) identifies the sign function. Then a new memristive chaotic system is constructed:

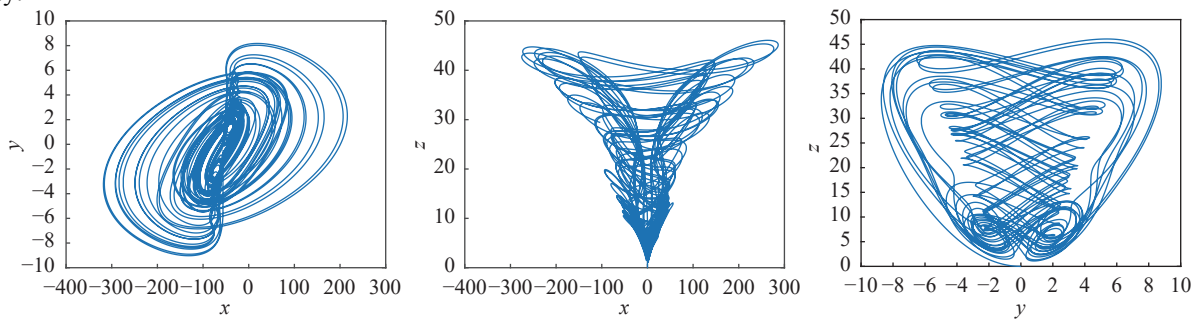
$$\begin{cases} \dot{x} = -ax + yzW(w) \\ \dot{y} = -x + by \\ \dot{z} = -cz + dy^2 \\ \dot{w} = yz \end{cases} \tag{2}$$

where x, y, z, and w are state variables; a, b, c, and d are parameters to be determined and A=0.666 7, B=1.5.

By changing the parameters and initial values respectively, the chaotic system we design generates different chaotic attractors. The values in Table 1 correspond to the three different shapes of attractors in Fig. 1, as shown in Fig. 1(a)−Fig. 1(c), Fig. 1(d)−Fig. 1(f), and Fig. 1(g)−Fig. 1(i), respectively.

Table 1 Parameters and initial values for memristive chaotic system

Attractor	Parameter				Initial value
	a	b	c	d	(x_0,y_0,z_0,w_0)
Fig. 1(a)−Fig. 1(c)	16	9	5	8	(1,0,0,1)
Fig. 1(d)−Fig. 1(f)	16	9	30	8	(1,0,0,1)
Fig. 1(g)−Fig. 1(i)	16	9	5	8	(1,0,0,40)



(a) x-y plane (Attractor 1) (b) x-z plane (Attractor 1) (c) y-z plane (Attractor 1)

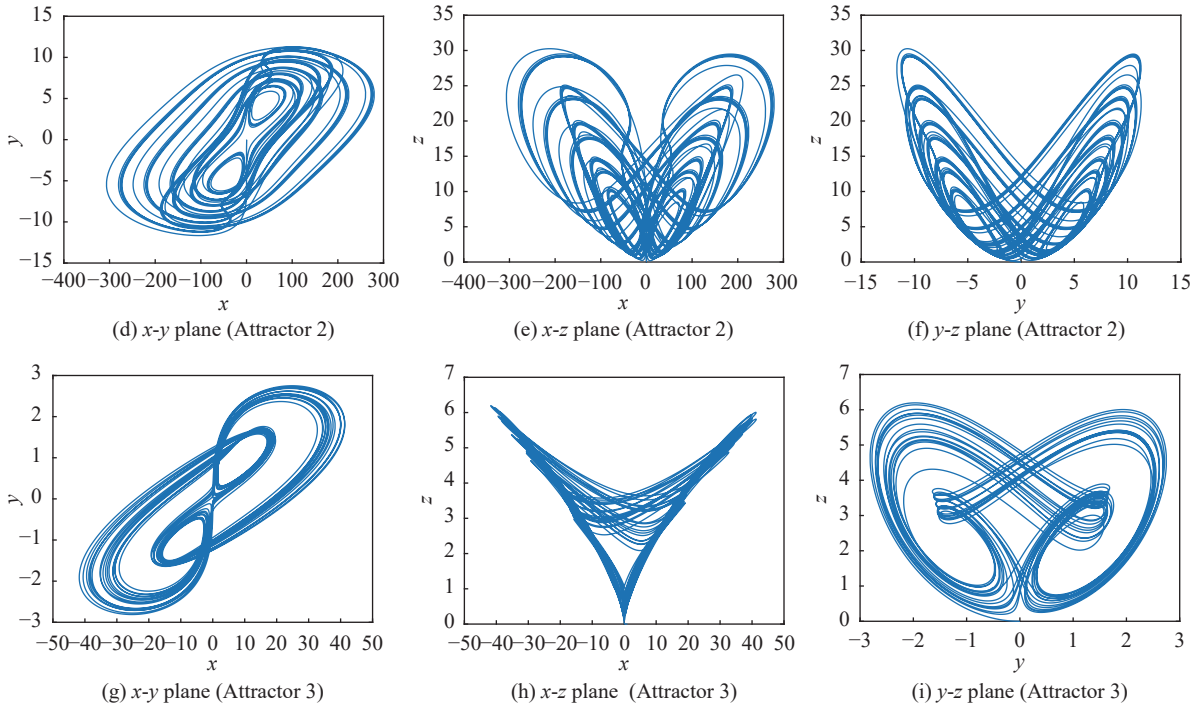


Fig. 1 Phase portraits of three different attractors of the UWPR-MCS with different parameters and initial values

2.2 Circuit implementation

To demonstrate the practicality of the UWPR-MCS, a circuit is implemented based on the defined differential equations. As can be seen in Fig. 1, the range of state variables exceeds the actual saturation voltage of the operational amplifier. Therefore, the variables in (2) are transformed by a proportional compression before designing the circuit as

$$\begin{cases} \dot{x} = -ax + 0.25yz(-A + 5B|w|) \\ \dot{y} = -20x + by \\ \dot{z} = -cz + 5dy^2 \\ \dot{w} = 5yz \end{cases} \tag{3}$$

Make the ranges of x, y, z and w as 1/100, 1/5, 1/5 and 1/5 of the original, respectively.

The equations about the voltages and time of the circuit according to (3) are

$$\begin{cases} RC_1 \dfrac{\mathrm{d}v_x}{\mathrm{d}t} = -\dfrac{R}{R_a}v_x - \dfrac{R}{R_A}v_y v_z + \dfrac{R}{R_B}v_y v_z |v_w| \\ RC_2 \dfrac{\mathrm{d}v_y}{\mathrm{d}t} = -\dfrac{R}{R_1}v_x + \dfrac{R}{R_b}v_y \\ RC_3 \dfrac{\mathrm{d}v_z}{\mathrm{d}t} = -\dfrac{R}{R_c}v_z + \dfrac{R}{R_d}v_y^2 \\ RC_4 \dfrac{\mathrm{d}v_w}{\mathrm{d}t} = \dfrac{R}{R_2}v_y v_z \end{cases} \tag{4}$$

where R =100 kΩ and C_1=C_2=C_3=C_4 =10 nF. Then, the values of the individual resistors are obtained from the correspondence of the parameters, where R_a=6.25 kΩ, R_b= 1.11 kΩ, R_C =20 kΩ, R_d =20 kΩ, R_A =60 kΩ, R_B= 53.3 kΩ, R_1 =5 kΩ, R_2 =20 kΩ, R_0 =10 kΩ. The circuit shown in Fig. 2 is built on the basis of the above preparations. Fig. 3 displays the results of the circuit simulation and they are identical to the numerical simulation of the UWPR-MCS.

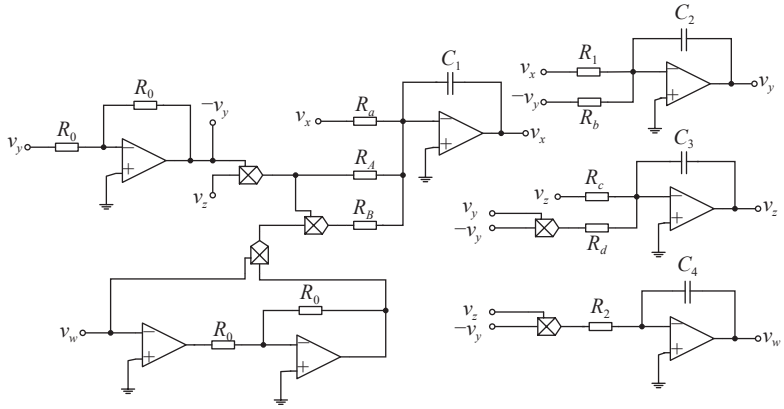


Fig. 2 Circuit implementation of the chaotic system

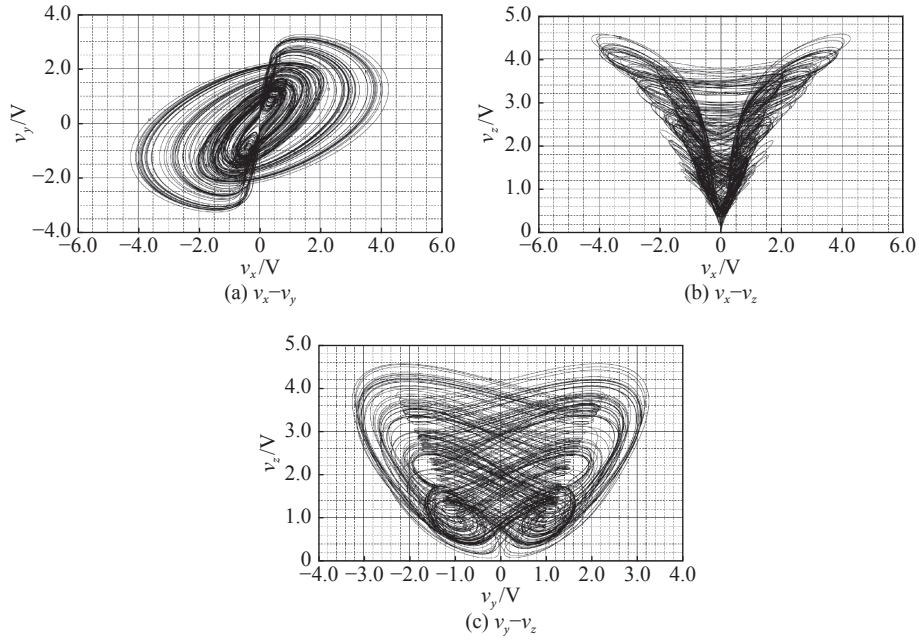


Fig. 3 Phase diagrams from the circuit simulation

2.3 Dynamical behaviors analysis

2.3.1 Dissipative analysis

The vector field divergence of the UWPR-MCS is given by

$$\nabla V = \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{w}}{\partial w} = -a + b - c. \tag{5}$$

Apparently, ∇V is negative while setting the control parameters $c>b-a$. In this circumstance, all trajectories of the system are limited to a certain area, which means the UWPR-MCS is dissipative.

2.3.2 Equilibrium stability analysis

The equilibrium point of the system is obtained by replacing the left side of (2) with 0 and solving

$$\begin{cases} -ax + yzW(w) = 0 \\ -x + by = 0 \\ -cz + dy^2 = 0 \\ yz = 0 \end{cases}. \tag{6}$$

It is observed that the result of (6) is independent of w, which means that w can be any real number. As a result, the UWPR-MCS has a line of equilibrium as follows:

$$O = \{(x, y, z, w) | x = y = z = 0, w = \xi\} \tag{7}$$

where ξ represents an uncertain constant.

The Jacobian matrix $\boldsymbol{J}$ is described as

$$\boldsymbol{J} = \begin{pmatrix} -a & zW(w) & yW(w) & yzB\mathrm{sgn}(w) \\ -1 & b & 0 & 0 \\ 0 & 2dy & -c & 0 \\ 0 & z & y & 0 \end{pmatrix}. \tag{8}$$

Then, with the identity matrix $\boldsymbol{E}$, we solve $|\lambda\boldsymbol{E}-\boldsymbol{J}|=0$ to get the characteristic equation of the proposed system at the equilibrium set O as

$$\lambda\left(\lambda^3 + \mu_1\lambda^2 + \mu_2\lambda + \mu_3\right) = 0 \tag{9}$$

where

$$\begin{cases} \mu_1 = a - b + c \\ \mu_2 = -ab + ac - bc \\ \mu_3 = -abc \end{cases}. \tag{10}$$

From (9), there is a zero eigenvalue and three non-zero eigenvalues of Jacobian matrix (8). For these non-zero eigenvalues, the Routh-Hurwitz criterion corresponding to the cubic polynomial in (9) is given by

$$\begin{cases} \mu_1 > 0 \\ \mu_3 > 0 \\ \mu_1\mu_2 - \mu_3 > 0 \end{cases}. \tag{11}$$

Obviously, none of the three cases mentioned above can totally satisfy (11). Thus, the line equilibrium O is unstable, resulting in the periodic or chaotic behavior in the UWPR-MCS.

2.3.3 Lyapunov exponents spectrum and bifurcation diagram analysis

In this case, to explore the influence of parameters on the

dynamic behavior of the proposed memristive chaotic system, the Lyapunov exponents spectrum and bifurcation diagram of state variable z changing with $d\in(0,10^7]$ are shown in Fig. 4. The bifurcation diagram reveals that the state of the memristive system changes when d is in the interval between $(0.8\times10^6, 2.75\times10^6)$ and $(8.9\times10^6, 9.5\times10^6)$.

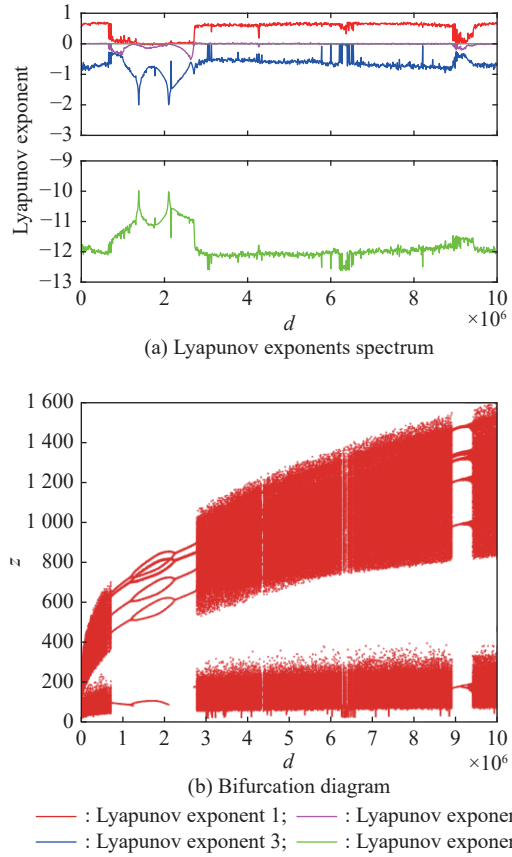


Fig. 4 Lyapunov exponents spectrum and bifurcation diagram for increasing parameter d

However, from the details shown in Fig. 4(a), if d is in range $(1.28\times10^6, 2.29\times10^6)$, one Lyapunov exponent is almost 0 and three are negative and the UWPR-MCS is in a periodic state. If $d\in(9\times10^6, 9.5\times10^6)$, one Lyapunov exponent is positive and three are negative, the UWPR-MCS is in a chaotic state. Except for these regions, all Lyapunov exponents are one positive, one zero, and the others negative. In this case, the UWPR-MCS is not only in the chaotic state, but also has an increasingly divergent attraction domain.

For dynamical behaviour that varies with the initial state, as shown in Fig. 5, the UWPR-MCS is chaotic if $w_0\in(-60,60)$.

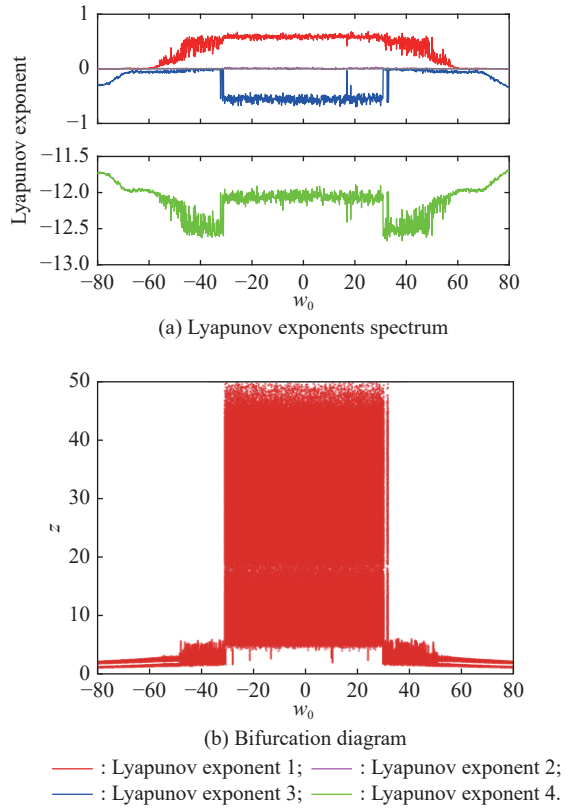


Fig. 5 Lyapunov exponents spectrum and bifurcation diagram for increasing parameter w_0

2.4 Utility of the proposed memristive chaotic system in image encryption

Thanks to the incorporation of a memristor in this system, the chaotic state is not only influenced by the parameters and initial values of the system itself but also by the internal parameters of the memristor, which makes the dynamical behaviour of the chaotic system quite enriched. In comparison with the common chaotic systems in image encryption, the proposed memristor-based chaotic system not only offers a large set of parameters, but also maintains a stable chaotic state over a super wide range of parameters. The robustness of the chaotic cipher stream is ensured even when the parameters are slightly changed. Moreover, the proposed system moves over a broad space and is able to generate pseudo-random sequences sufficient to satisfy the requirements during the encryption of images with large amounts of data. These features give UWPR-MCS outstanding practical value in image encryption.

3. Image encryption scheme

We focus on the super-wide parameter range and excellent robustness of the UWPR-MCS and use it to design an

image encryption scheme. First of all, set the secret key of this scheme as $\text{Key}=\{k_{32}, x_0^{(1)}, y_0^{(1)}, z_0^{(1)}, w_0^{(1)}, x_0^{(2)}, y_0^{(2)}, z_0^{(2)}, w_0^{(2)}\}$, where k_{32} represents 32 integers in the range of [0,255], and the other eight float numbers mean two sets of initial values of the UWPR-MCS.

The size of the plain image $\boldsymbol{P}$ (i, j, k) is identified as $m\times n\times h$, with $h=3$ for color images and $h=1$ for gray images. In this case, the total number of pixels in the plain image is noted as N, and the sum of pixel values S is calculated as

$$S=\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{h}\boldsymbol{P}(i,j,k). \tag{12}$$

where i, j, and k represent the coordinates of an image in three dimensions.

Once these basics have been defined, the whole encryption process begins.

3.1 External key protection based on Grain-128a

The Grain-128a [19] is a highly secure lightweight encryption method and is the latest version of the Grain series of encryption algorithms [20,21] proposed by Hell et al. in 2011. Grain-128a system consists of a 128-level linear feedback shift register (LFSR), a 128-level non-linear feedback shift register (NFSR) and a non-linear filter function, which supports a 128-bit key and a 96-bit initial vector. In this paper, the Grain-128a algorithm is used to generate the plaintext-associated internal key.

Step 1 The first portion of the secret key, k_{32}, is decomposed into a 256-bit binary sequence, noted as $\{k_0, k_1, \cdots, k_{255}\}$. Make $\{k_0, k_1, \cdots, k_{127}\}$ and $\{k_{128}, k_{129}, \cdots, k_{255}\}$ as the initial states of LFSR and NFSR respectively. Then consider that $\{p_i, p_{i+1}, \cdots, p_{i+127}\}$ are the contents of the LFSR and $\{q_i, q_{i+1}, \cdots, q_{i+127}\}$ are the contents of the NFSR.

Step 2 A new binary sequence is obtained by the update functions, the filter function and the output function.

The update function of the LFSR is

$$p_{i+128}=p_i+p_{i+7}+p_{i+38}+p_{i+70}+p_{i+81}+p_{i+96}. \tag{13}$$

Unlike the LFSR, the update function of the NFSR contains the state bits of the LFSR, as shown in

$$\begin{aligned}q_{i+128}=&\,p_i+q_i+q_{i+26}+q_{i+56}+q_{i+91}+q_{i+96}+\\&q_{i+3}q_{i+67}+q_{i+11}q_{i+13}+q_{i+17}q_{i+18}+q_{i+27}q_{i+59}+\\&q_{i+40}q_{i+48}+q_{i+61}q_{i+65}+q_{i+68}q_{i+84}+q_{i+22}q_{i+24}q_{i+25}+\\&q_{i+70}q_{i+78}q_{i+82}+q_{i+88}q_{i+92}q_{i+93}q_{i+95}.\end{aligned} \tag{14}$$

Before the final output, a non-linear filter function is displayed as

$$\begin{aligned}h(x)=&\,q_{i+12}p_{i+8}+p_{i+13}p_{i+20}+q_{i+95}p_{i+42}+\\&p_{i+60}p_{i+79}+q_{i+12}q_{i+95}p_{i+94}.\end{aligned} \tag{15}$$

Finally, an update binary sequence H_i ($i\in[0,255]\in\mathbf{N}$) is obtained from

$$\begin{aligned}H_i=&\,h(x)+p_{i+93}+q_{i+2}+q_{i+15}+q_{i+36}+\\&q_{i+45}+q_{i+64}+q_{i+73}+q_{i+89}.\end{aligned} \tag{16}$$

Step 3 The 256-bit H_i is divided into eight 32-bit subsequences. Each sub-sequence is converted into a decimal number, denoted as $H1$, $H2$, $\cdots$, $H8$. To obtain the updated initial values for the chaotic system, the calculation is completed as

$$\begin{cases}x_0^{(3)}=\text{rem}\left(x_0^{(1)}\cdot\dfrac{(H1+S)}{256m},400\right)\\ y_0^{(3)}=\text{rem}\left(y_0^{(1)}\cdot\dfrac{(H2+S)}{32mn},40\right)\\ z_0^{(3)}=\text{rem}\left(z_0^{(1)}\cdot\dfrac{(H3+S)}{256n},400\right)\\ w_0^{(3)}=\text{rem}\left(w_0^{(1)}\cdot\dfrac{(H4+S)}{2^{18}},40\right)\end{cases}, \tag{17}$$

$$\begin{cases}x_0^{(4)}=\text{rem}\left(x_0^{(2)}\cdot\dfrac{(H5+S)}{256m},400\right)\\ y_0^{(4)}=\text{rem}\left(y_0^{(2)}\cdot\dfrac{(H6+S)}{32mn},40\right)\\ z_0^{(4)}=\text{rem}\left(z_0^{(2)}\cdot\dfrac{(H7+S)}{256n},400\right)\\ w_0^{(4)}=\text{rem}\left(w_0^{(2)}\cdot\dfrac{(H8+S)}{2^{18}},40\right)\end{cases}, \tag{18}$$

where $\text{rem}(\alpha, \beta)$ represents the function that makes the value of a within the range $(-\beta, \beta)$.

Step 4 Iterate the proposed chaotic system with initial values $x_0^{(3)}$, $y_0^{(3)}$, $z_0^{(3)}$, $w_0^{(3)}$, $x_0^{(4)}$, $y_0^{(4)}$, $z_0^{(4)}$ and $w_0^{(4)}$ to generate eight chaotic sequences of length $N/4+3\,000$. For eliminating transient effects, the first 3 000 elements of the chaotic sequences are removed and the resulting sequences are denoted as s_1, s_2, $\cdots$, s_8. They will be handled in different ways to obtain the chaotic stream cipher for each of the processes.

3.2 Dynamic pixel masking relying on chaotic random selection

The values of image pixels always present visually meaningful information in a certain pattern. The work in this subsection is to mask such a particular pattern, as shown in Fig. 6.

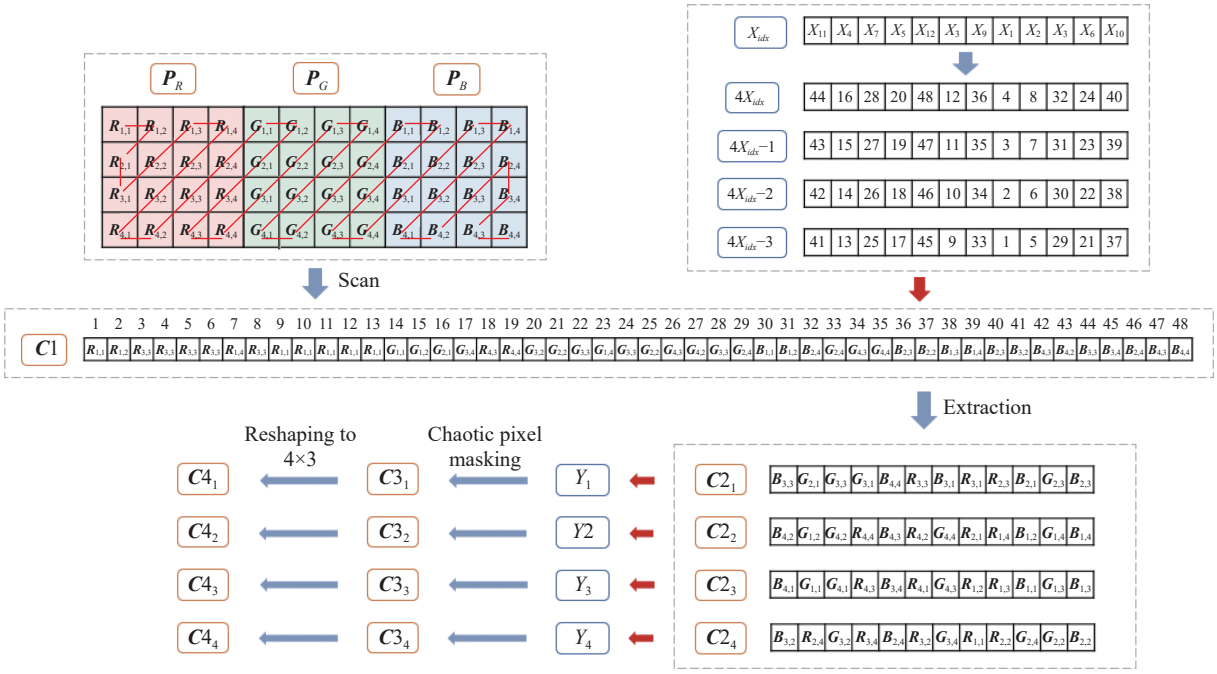


Fig. 6 An example of dynamic pixel masking

Step 1 The R, G, and B channels of the original color image $\boldsymbol{P}(i, j, k)$ are spliced side-by-side and the pixels are scanned in diagonal order to get $C1$.

Step 2 The sequence X prepared by (19) is then sorted in the ascending order, and the index sequence of elements is noted as X_{idx}.

$$X = \left\lfloor (s_1 + s_2 + s_7 + s_8) \times 10^5 + 2^{16} \right\rfloor \bmod N \tag{19}$$

where $\lfloor \cdot \rfloor$ means taking an integer to $-\infty$.

Four sub-vectors are extracted from $\boldsymbol{C}1$ with indexes $4X_{idx}$, $4X_{idx}-1$, $4X_{idx}-2$, and $4X_{idx}-3$ respectively and are recorded as $\boldsymbol{C}2_i$ (i=1,2,3,4).

Step 3 The way to mask each value in $\boldsymbol{C}2_i$ is chosen by the elements in the cipher streams Y_i.

$$Y_i = \left\lceil s_i \cdot (\max\ s_{i+4} + \min\ s_{i+4}) \times 10^6 \right\rceil \bmod 256 \tag{20}$$

where $\lceil \cdot \rceil$ represents taking an integer to $+\infty$.

As given in (21), when $Y_i(j) \bmod 3 = 0$, a bit-level XOR operation $\oplus$ is acted.

$$\boldsymbol{C}3_i(j) = \boldsymbol{C}2_i(j) \oplus Y_i(j) \tag{21}$$

If $Y_i(j) \bmod 3 = 1$, the value of $\boldsymbol{C}2_i(j)$ is executed as a bit-cyclic shift left operation, as

$$\boldsymbol{C}3_i(j) = \boldsymbol{C}2_i(j) \lll (Y_i(j) \bmod 6 + 2). \tag{22}$$

Besides, if $Y_i(j) \bmod 3 = 2$, the value of $\boldsymbol{C}2_i(j)$ is computed by a bit cyclic shift left operation.

$$\boldsymbol{C}3_i(j) = \boldsymbol{C}2_i(j) \ggg (Y_i(j) \bmod 6 + 2) \tag{23}$$

where $j = 1, 2, \cdots, N/4$.

Step 4 The four one-dimensional sequences $\boldsymbol{C}3_i$ are reshaped into four two-dimensional matrices $\boldsymbol{C}4_i$ with m rows and $N/4m$ columns.

3.3 Permutation based on Brownian motion matrix

Brownian motion is a continuous irregular random motion of particles in a fluid medium, named after the British botanist Robert Brown, who first discovered it. Fig. 7 shows the trajectory of a particle moving 200 times in a fixed space as Brownian motion, which is simulated by Monte Carlo method.

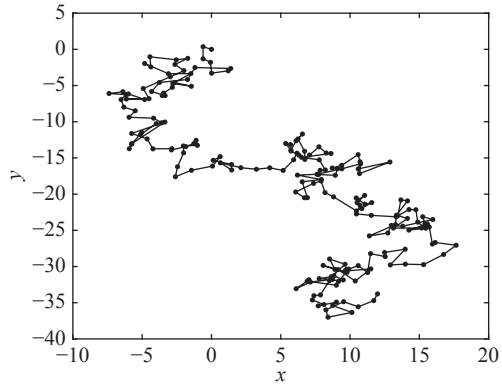


Fig. 7 An example of simulating two-dimensional Brownian motion using Monte Carlo method

The movement direction of the particle is represented by the two polar angles in polar coordinates, i.e., a and b, from

$$\begin{cases} a = a_i \cdot \pi, & a_i \in [0, 1] \\ b = b_i \cdot 2\pi, & b_i \in [0, 1] \end{cases}. \tag{24}$$

Then, the position after a single move in Cartesian coordinates (x, y) is

$$\begin{cases} x = r\sin a \cos b \\ y = r\sin a \sin b \end{cases} \tag{25}$$

where the polar radius $r\in[0,+\infty]$.

a_i and b_i in the classical Monte Carlo method are generated by the ordinary random number function rand(·). However, if each pixel of the image is considered as a Brownian particle, the number of iterations would be so large that the rand(·) function exhibits periodicity. In this scheme, two chaotic cipher streams with excellent randomness are used instead of rand(·) to provide a_i and b_i. This results in each pixel moving on a different trajectory from each other.

The chaotic sequences used to generate the Brownian motion matrices are given by

$$Z_i = (s_i \times 10^4) \bmod 1, \quad i = 1,2,\cdots,8. \tag{26}$$

Make Z_i and $Z_{i+4}(i=1,2,3,4)$ correspond to a_i and b_i respectively, and iterate (25) for R times to obtain the four chaotic Brownian motion matrices $\boldsymbol{BM}_i(l_x,l_y)$ $(i=1,2,3,4)$. The elements in $\boldsymbol{C}4_i$ are relocated according to $\boldsymbol{BM}_i(l_x,l_y)$ to get $\boldsymbol{C}5_i$ as

$$\boldsymbol{C}5_i(x,y) = \boldsymbol{C}4_i(l_x,l_y). \tag{27}$$

3.4 Multiplicative diffusion over GF(2^8)

To extend the impact of a single pixel, $\boldsymbol{C}5_i$ are first reconstructed as a sequence $\boldsymbol{C}6$, and one more multiplicative diffusion operation over GF(2^8) is performed according to (28). In particular, multiplication "⊗" over the finite field GF(2^8) is calculated in advance and stored in a table, so that diffusion operations can be executed by looking up the table, effectively increasing the computational speed.

$$\begin{cases} \boldsymbol{C}7(1) = \boldsymbol{C}6(1) \\ \boldsymbol{C}7(i) = \boldsymbol{C}6(i) \otimes \boldsymbol{C}7(i-1) \otimes V(i) \end{cases} \tag{28}$$

where $i = 2, 3, \cdots, m\times n\times h$, and $V=\{W_1, W_2, W_3, W_4\}$ of length N. W_i is given by

$$W_i = \left\lfloor s_i \times 10^7 + \left\lceil s_{i+4} \times 2^{18} \right\rceil \right\rfloor \bmod 256 \tag{29}$$

where $i=1,2,3,4$.

3.5 Overall process of the encryption and decryption scheme

The image encryption scheme proposed in this paper can be summarised according to the above mentioned components and illustrated in Fig. 8.

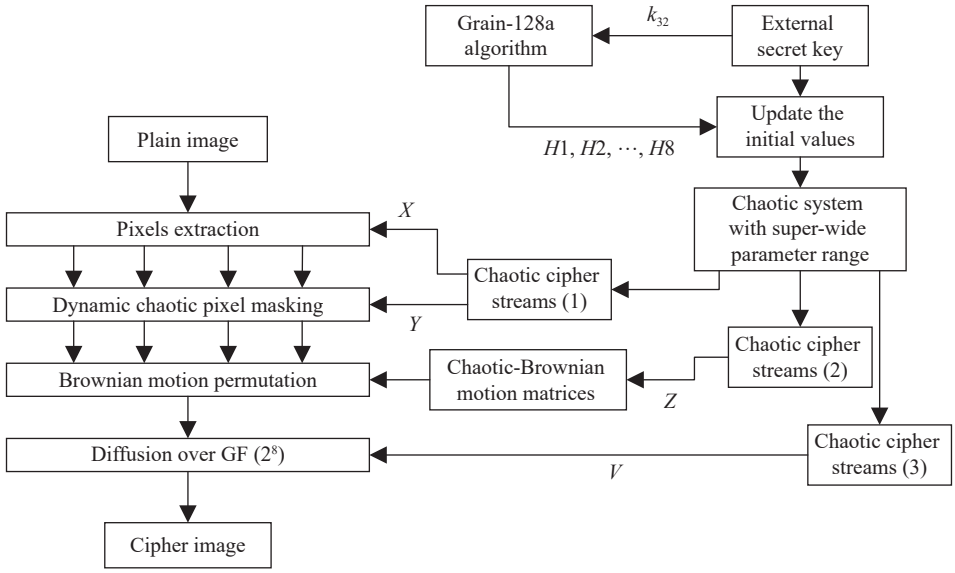


Fig. 8 Flowchart of the proposed image encryption scheme

Step 1 Enter the external secret key and the image to be encrypted.

Step 2 Update the external key to the internal key via the Grain-128a algorithm and the plain image as described in Subsection 3.1. Eight initial chaotic pseudo-random sequences are generated based on the new internal key and the proposed memristive chaotic system.

Step 3 From Fig. 6, the subsequences $\boldsymbol{C}2_i(i=1,2,3,4)$ are extracted from the original image in the order determined by the chaotic cipher streams. Then, the values of the pixels are masked to get $\boldsymbol{C}4_i$ according to Subsection 3.2.

Step 4 Firstly, we generate the Brownian motion matrices using the chaotic cipher streams as the directional control parameters. Afterward, change the positions of the elements in $\boldsymbol{C}4_i$ one by one based on the resulting Brownian motion matrices, as explained in Subsection 3.3.

Step 5 The resulting submatrices $\boldsymbol{C}5_i$ are converted into a one-dimensional sequence $\boldsymbol{C}6$ in the order in which they are extracted. Each pixel in $\boldsymbol{C}6$ is diffused by a multiplicative look-up table method over a finite field, as in Subsection 3.4.

Step 6 Finally, the sequence $\boldsymbol{C}7$ is reshaped to size $m\times n\times h$ to obtain the encrypted image.

The decryption of a cipher image is defined as the reverse of the encryption process.

Step 1 Enter the external key and the cipher image.

Step 2 Update the external key and get the chaotic sequence for decryption.

Step 3 Make the image into a one-dimensional sequence $\boldsymbol{C}7'$ and reversely diffuse the elements of the sequence by the division lookup table method on a finite field GF(2^8).

Step 4 The resulting sequence $\boldsymbol{C}6'$ is decomposed into subvectors by the method explained in Fig. 6 and reshaped into sub-matrices $\boldsymbol{C}5'_i(i=1,2,3,4)$. Generate the chaotic Brownian motion matrices and recover the positions of the permuted elements in $\boldsymbol{C}5'_i$.

Step 5 The obtained matrices $\boldsymbol{C}4'_i$ are turned into one-dimensional sequences $\boldsymbol{C}3'_i$, which is then processed into $\boldsymbol{C}2'_i$ with chaotic key streams.

If $Y_i(j)$mod3=0, $\boldsymbol{C}2'_i$ are derived from

$$\boldsymbol{C}2'_i(j)=\boldsymbol{C}3'_i(j)\oplus Y_i(j) \tag{30}$$

where $j=1,2,\cdots,N/4$, and $\oplus$ is bitwise XOR.

Unlike the encryption process, if $Y_i(j)$mod3=1, then

$$\boldsymbol{C}2'_i(j)=\boldsymbol{C}3'_i(j)\ggg(Y_i(j)\bmod 6+2), \tag{31}$$

and if $Y_i(j)$mod3=2,

$$\boldsymbol{C}2'_i(j)=\boldsymbol{C}3'_i(j)\lll(Y_i(j)\bmod 6+2). \tag{32}$$

Step 6 These sequences are recomposed and reconstructed into the decrypted image.

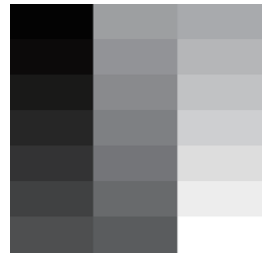
3.6 Simulation results and security analysis of the encryption scheme

As shown in Fig. 9(a) to Fig. 9(d), the classical Baboon, Peppers, Girl, and Black-to-White are used to test the encryption and decryption of the proposed image encryption algorithm, and to further security analysis. The cipher images after encryption by this cryptosystem are displayed in Fig. 9(i) to Fig. 9(l). The decrypted images are shown in Fig. 9(q) to Fig. 9(t). It is clear from these examples that the cipher images are all noise-like and the decrypted images cannot be visually identified as different from the original images, implying that the proposed cryptosystem is capable of a great encryption.

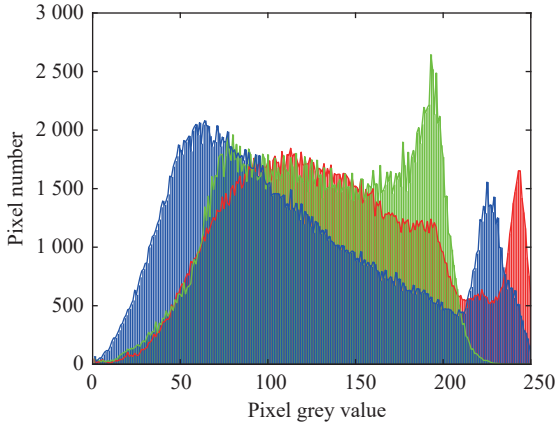
(a) Baboon

(b) Peppers

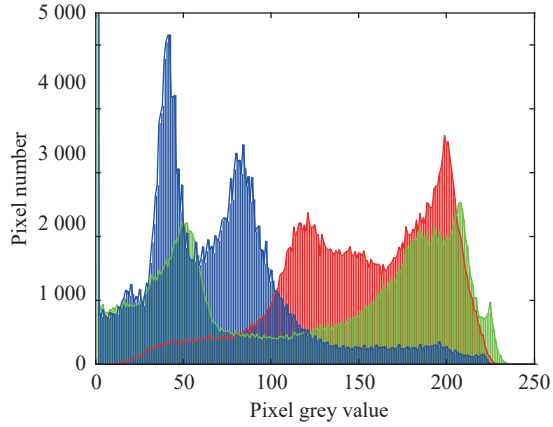
(c) Girl



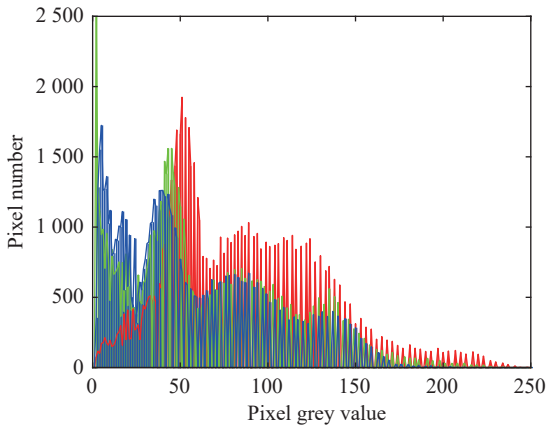
(d) Black-to-White



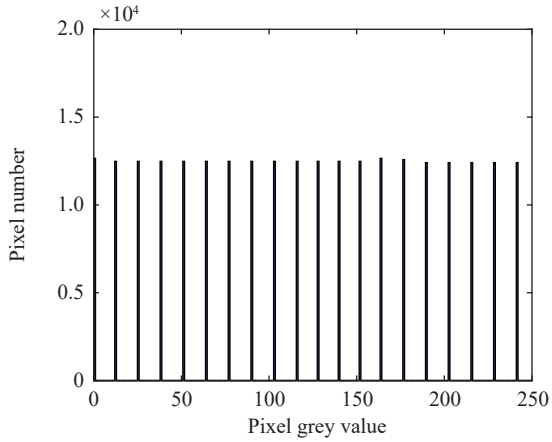
(e) Histogram of the Baboon



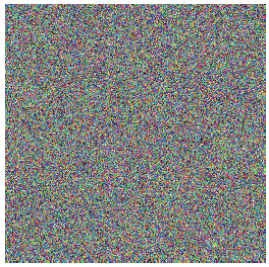
(f) Histogram of the Peppers



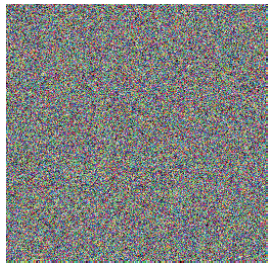
(g) Histogram of the Girl



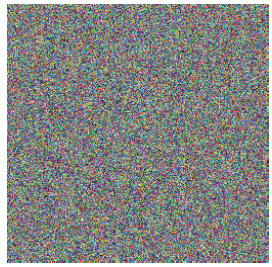
(h) Histogram of the Black-to-White



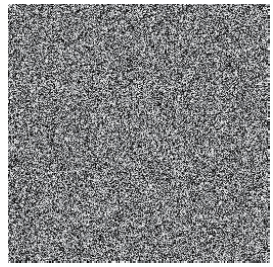
(i) Encrypted Baboon image



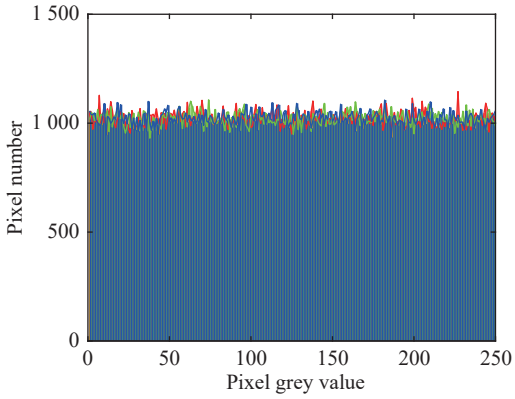
(j) Encrypted Peppers image



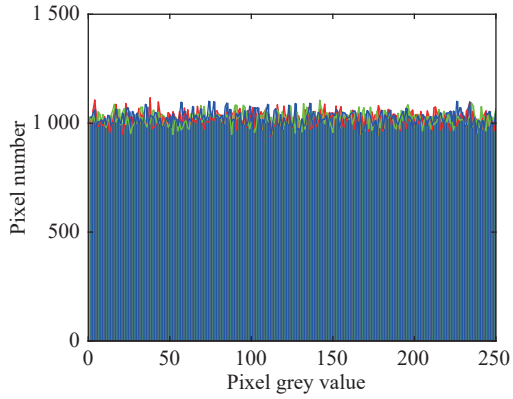
(k) Encrypted Girl image



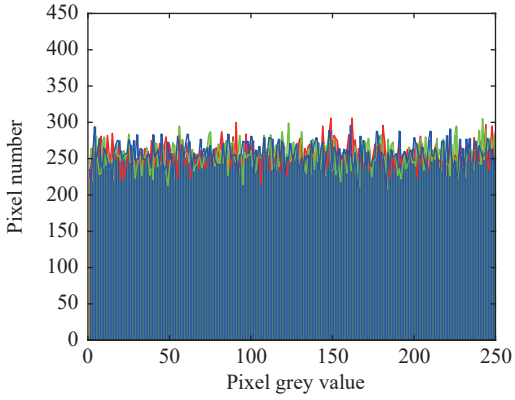
(l) Encrypted Black-to-White image



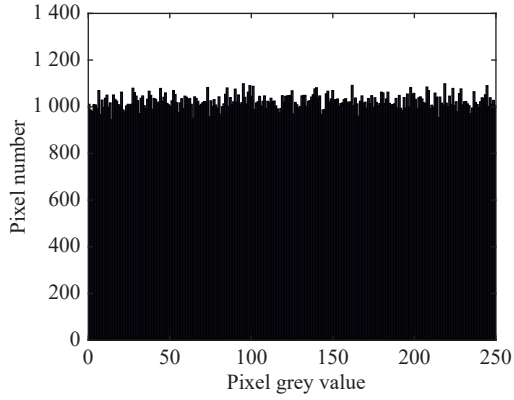
(m) Histogram of the encrypted Baboon



(n) Histogram of the encrypted Peppers



(o) Histogram of the encrypted Girl



(p) Histogram of the encrypted Black-to-White

(q) Decrypted Baboon image (r) Decrypted Peppers image (s) Decrypted Girl image (t) Decrypted Black-to-White image

Fig. 9 Encryption, decryption experiments and histograms

3.6.1 Histogram analysis

The histogram describes the distribution of each grayscale in the image. The longer a grayscale rectangle is, the more frequently it appears in the image.

Histograms of the plain images are exhibited in Fig. 9(e) to Fig. 9(h), where the R channel, the G channel and the B channel of the color image are differentiated by red, green, and blue.

Distinct distributional features are presented in the histogram of the plain image because the frequency of the grayscales makes up a variable and smooth curve. Fig. 9(m) to Fig. 9(p) show the corresponding histograms for the cipher images. Obviously, the variation in the frequencies of the grayscales of the cipher image is so slight that no valid information is available from the histogram. It illustrates that the proposed image encryption scheme has excellent scrambling performance and is effective in resisting known/chosen plaintext attack.

3.6.2 Information entropy analysis

Image information entropy is a statistic related to the probability of occurrence for each gray level in an image, and is defined by (33). Consider the grayscale $i(i\in[0,255])$ and its probability as m_i and $p(m_i)$, respectively. Then, $H(m)$ is given by

$$H(m)=-\sum_{i=0}^{255}p(m_i)\log_2 p(m_i) \tag{33}$$

where $p(m_0)+p(m_1)+\cdots+p(m_{255})=1$. According to (33), the ideal value is 8 for an image containing 256 gray levels.

Table 2 lists the information entropy results for this encryption scheme. Compared to the plain image, the information entropy of the cipher image increases significantly and is very close to the ideal value. The uniform distribution of grayscales in the cipher image is proven numerically.

Table 2 Results of information entropy

Image	Information entropy		
	R	G	B
Girl	6.4200	6.4457	6.3807
Cipher of Girl	7.9974	7.9975	7.9973
Baboon	7.7067	7.4744	7.7522
Cipher of Baboon	7.9994	7.9994	7.9992
Peppers	7.3388	7.4963	7.0583
Cipher of Peppers	7.9994	7.9994	7.9993
Lena	7.2531	7.5940	6.9684
Cipher of Lena	7.9993	7.9993	7.9994
Lena in [22]	7.9993	7.9993	7.9993
Lena in [23]	7.9994	7.9994	7.9993
Lena in [24]	7.9896	7.9885	7.9899
Lena in [25]	7.9972	7.9972	7.9975

3.6.3 Key space analysis

The set that contains all the secret keys of an encryption system is known as the key space. On the one hand, the capacity of the key space measures whether the encryption algorithm provides the user with sufficient changeable keys and, on the other hand, whether the encryption system is robust against exhaustive attacks.

We set the secret key as $\text{Key}=\{k_{32},x_0^{(1)},y_0^{(1)},z_0^{(1)},w_0^{(1)},x_0^{(2)},y_0^{(2)},z_0^{(2)},w_0^{(2)}\}$. If the initial values of the chaotic system have an effective precision of 10^{-14}, the total key space of the introduced scheme is $2^{256}\times10^{14\times8}\approx10^{189}$. We effectively extend the time required for brute-force attacks by setting the key space much larger than 2^{100}, thus achieving outstanding performance against exhaustive attacks, as shown in Table 3.

Table 3 Key space comparison of several methods

Method	Space
Proposed method	$2^{256}\times10^{112}$
Method in [5]	10^{64}
Method in [16]	2^{267}
Method in [22]	2^{128}
Method in [23]	$2^{144}\times10^{126}$

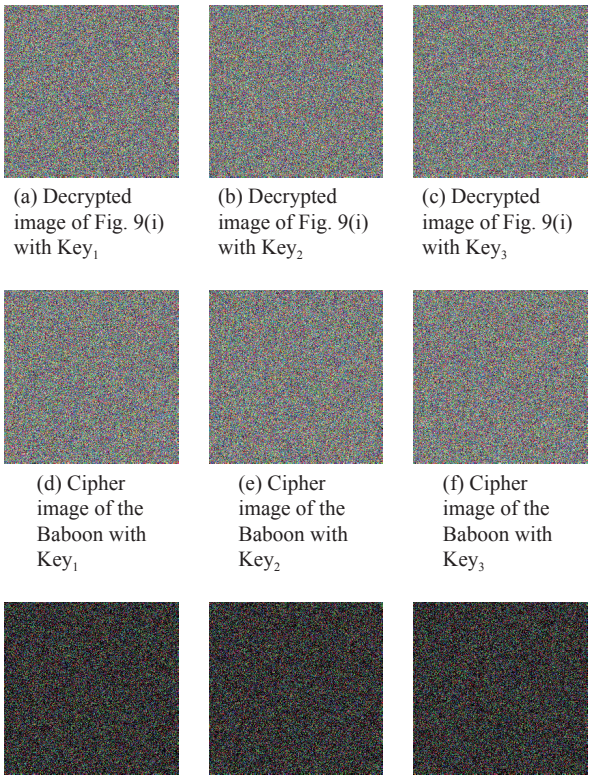
3.6.4 Key sensitivity analysis

An image encryption system which is extremely sensitive to the secret key is effective and practical. The values of the three representative keys, Key_1, Key_2, and Key_3, are listed in Table 4 and they are used as control groups to analyze the key sensitivity.

Table 4 Several keys for key sensitivity analysis

Secret key	Value
Key_1	$\{k_1+1, k_2, \cdots, k_{32}, x_0^{(1)}, y_0^{(1)}, z_0^{(1)}, w_0^{(1)}, x_0^{(2)}, y_0^{(2)}, z_0^{(2)}, w_0^{(2)}\}$
Key_2	$\{k_{32}, x_0^{(1)}+10^{-14}, y_0^{(1)}, z_0^{(1)}, w_0^{(1)}, x_0^{(2)}, y_0^{(2)}, z_0^{(2)}, w_0^{(2)}\}$
Key_3	$\{k_{32}, x_0^{(1)}, y_0^{(1)}, z_0^{(1)}, w_0^{(1)}, x_0^{(2)}+10^{-14}, y_0^{(2)}, z_0^{(2)}, w_0^{(2)}\}$

It is necessary that only the exactly correct secret key can reconstruct the original image. Fig. 10(a)−Fig. 10(c) show the images obtained by decrypting Fig. 9(i) with Key_1, Key_2, and Key_3 respectively. None of them succeed in restoring the correct plain image.



(a) Decrypted image of Fig. 9(i) with Key_1 (b) Decrypted image of Fig. 9(i) with Key_2 (c) Decrypted image of Fig. 9(i) with Key_3

(d) Cipher image of the Baboon with Key_1 (e) Cipher image of the Baboon with Key_2 (f) Cipher image of the Baboon with Key_3

(g) Pixel-by-pixel difference between Fig.10(d) and Fig. 9(i) (h) Pixel-by-pixel difference between Fig.10(e) and Fig. 9(i) (i) Pixel-by-pixel difference between Fig.10(f) and Fig. 9(i)

Fig. 10 Key sensitivity analysis

Additionally, the secret key has an impact on the generation of the cipher for the encryption process, which further influences the encrypted image. We encrypt Baboon with Key_1, Key_2, and Key_3 respectively, and subtract the resulting cipher image (as in Fig. 10(d)−Fig. 10(f)) from Fig. 9(i) pixel by pixel. If the pixel values at the same position in both images are the same, then this position in the difference image is marked by black, and conversely is colored. It can be seen that the difference image displayed by Fig. 10(g)−Fig. 10(i) has a large number of colored dots. Thus, the two keys with a slight variation lead to quite different results.

Both experiments reveal that the proposed encryption scheme is equipped with a high level of key sensitivity.

3.6.5 Correlation analysis

One of the significant properties of images is the high correlation between the adjacent pixels. In order to keep the information in the plain image confidential, the image encryption operation should serve to break this correlation.

We analyse the correlation between adjacent pixels in two ways, i.e., the correlation graph and the correlation coefficient. Then, 4 000 pairs of pixels adjacent to each other horizontally (H), vertically (V), diagonally (D), and anti-diagonally (A) are randomly selected from the Peppers.

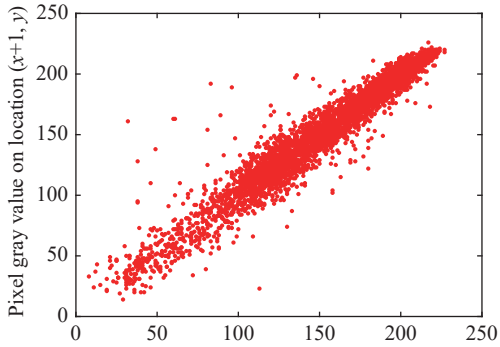
For correlation graphs, the points are plotted with one of the pixel pairs as the coordinate on the x-axis and the other as the coordinate on the y-axis. As shown in Fig. 11(a), Fig. 11(c), Fig. 11(e), and Fig. 11(g), the points of the plain image are clustered around $y=x$, which implies that the values of two adjacent pixels are almost equal. However, the points of the cipher image in Fig. 11(b), Fig. 11(d), Fig. 11(f), and Fig. 11(h) are uniformly distributed over the entire region, confirming the randomness of the adjacent pixels in the encrypted image.

Apart from these graphs, the correlation coefficient (CC) also reflects the correlation of the adjacent pixels. Consider a certain two adjacent pixels as x and y, the CC is given by

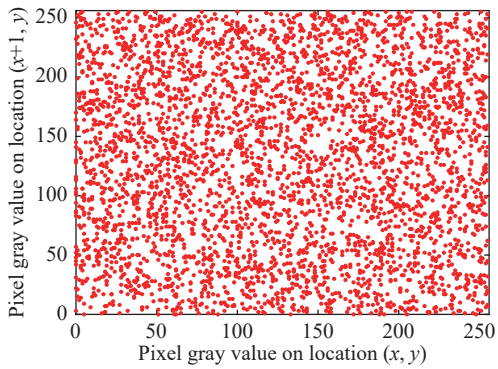
$$r_{xy} = \frac{\text{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{34}$$

where

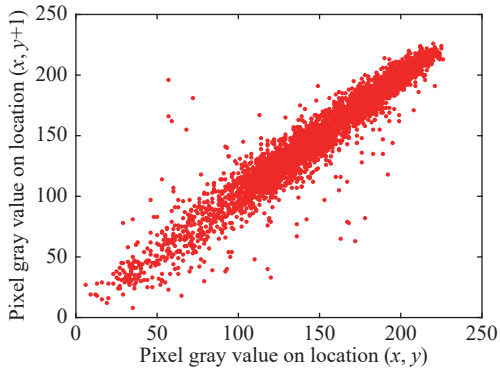
$$\begin{cases} \text{cov}(x,y) = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N}(x_i - \text{E}(x))(y_i - \text{E}(y)) \\ \text{E}(x) = \dfrac{\displaystyle\sum_{i=1}^{N} x_i}{N} \\ D(x) = \dfrac{\displaystyle\sum_{i=1}^{N}(x_i - \text{E}(x))^2}{N} \end{cases}. \tag{35}$$



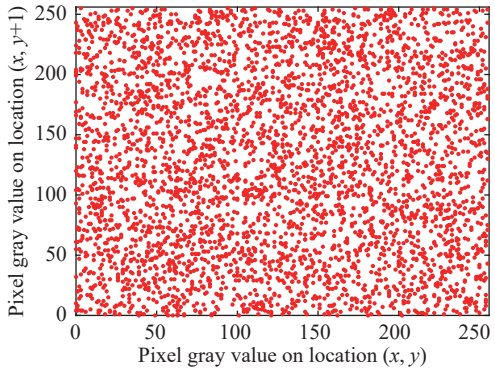
(a) Horizontal adjacent pixels in the Peppers



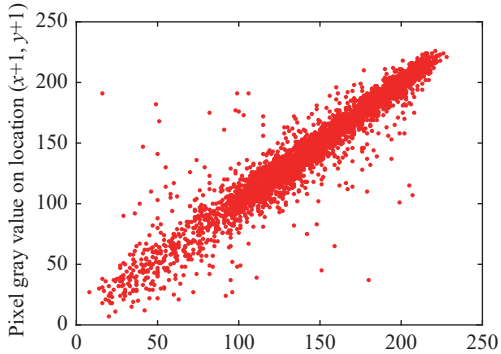
(b) Horizontal adjacent pixels in the encrypted Peppers



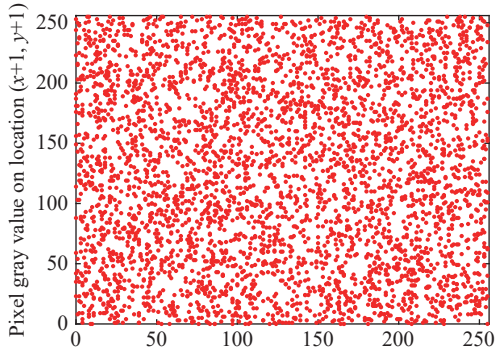
(c) Vertical adjacent pixels in the Peppers



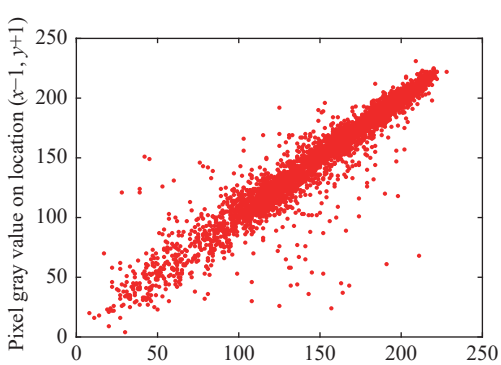
(d) Vertical adjacent pixels in the encrypted Peppers



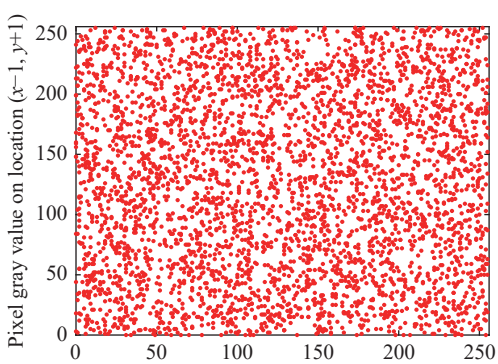
(e) Diagonal adjacent pixels in the Peppers



(f) Diagonal adjacent pixels in the encrypted Peppers



(g) Anti-diagonal adjacent pixels in the Peppers



(h) Anti-diagonal adjacent pixels in the encrypted Peppers

Fig. 11 Correlation distributions of two adjacent pixels in horizontal, vertical, diagonal, and anti-diagonal directions of the R-channel

In Table 5, the numerical results for the CC of the plain image are close to 1, but those for the cipher image are approximately 0, which proves that the proposed image encryption algorithm makes adjacent pixels almost independent.

Table 5 CCs of original and cipher images Continued

Image		Correlation coefficient			
		H	V	D	A
Baboon	R	0.8670	0.9203	0.8609	0.8600
	G	0.7651	0.8739	0.7406	0.7362
	B	0.8795	0.9084	0.8430	0.8458
Cipher of Baboon	R	−0.0009	−0.0015	−0.0019	0.0023
	G	0.0019	0.0040	−0.0001	0.0039
	B	−0.0004	0.0027	−0.0079	−0.0017
Peppers	R	0.9693	0.9625	0.9620	0.9598
	G	0.9829	0.9743	0.9685	0.9689
	B	0.9610	0.9670	0.9483	0.9452
Cipher of Peppers	R	−0.0003	−0.0028	−0.0058	0.0066
	G	−0.0025	0.0044	−0.0053	−0.0037
	B	0.0053	−0.0027	−0.0017	−0.0028
Lena	R	0.9892	0.9817	0.9707	0.9780
	G	0.9834	0.9703	0.9580	0.9621
	B	0.9600	0.9306	0.9147	0.9116
Cipher of Lena	R	−0.0005	−0.0072	−0.0049	0.0077
	G	−0.0025	−0.0055	0.0007	0.0046
	B	0.0030	−0.0019	−0.0004	0.0068
Lena in [22]	R	0.0128	−0.0031	−0.0033	−
	G	−0.0170	0.0160	−0.0093	−
	B	0.0001	−0.0190	−0.0130	−
Lena in [23]	R	0.0001	−0.0064	−0.0214	−
	G	0.0010	−0.0314	−0.0662	−
	B	0.0603	0.0005	−0.0019	−
Lena in [25]	R	0.0083	−0.0049	−0.0095	−
	G	−0.0054	0.0100	−0.0017	−
	B	−0.0010	0.0124	−0.0042	−

3.6.6 Resistance to differential attacks analysis

Once the resistance of an image encryption algorithm to differential attacks is discussed, it is important to evaluate how sensitive it is to plain images. That is, even tiny changes occurring in the plain image produce a much different cipher image. The number pixel change rate (NPCR) and the universal average change intensity (UACI) are two metrics defined for this purpose, and are given in (36) and (37). Two images, $\boldsymbol{P}_1(i, j)$ and $\boldsymbol{P}_2(i, j)$, are identical except that the pixel values at a random position (x, y) are different. Using the same secret key and the same cryptosystem to encrypt these two images, the corresponding cipher images $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are obtained. Then, the NPCR and UACI are computed as

$$\mathrm{NPCR} = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} D(i,j)\times 100\%}{MN}, \tag{36}$$

$$\mathrm{UACI} = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \frac{|C_1(i,j)-C_2(i,j)|}{255}\times 100\%}{MN}, \tag{37}$$

where M and N are the length and width of the image, respectively. Besides, if the values of $\boldsymbol{C}_1(i, j)$ and $\boldsymbol{C}_2(i, j)$ are equal, $\boldsymbol{D}(i,j)$=0. If they are not, then $\boldsymbol{D}(i, j)$=1.

The theoretical NPCR and UACI are 99.6094% and 33.4635% respectively. The results in Table 6 and Table 7, which are very close to the theoretical values, demonstrate that the proposed image encryption scheme is equipped with excellent resistance to differential attacks.

Table 6 NPCR of the proposed algorithm for different images %

Image	NPCR		
	R	G	B
Baboon	99.6088	99.6077	99.6070
Peppers	99.6008	99.6152	99.6093
Girl	99.6111	99.6016	99.6022
Lena	99.6110	99.6041	99.6084
Lena in [22]	99.6000	99.6000	99.6000
Lena in [23]	99.6065	99.6147	99.6235
Lena in [24]	99.6399	99.5987	99.6307
Lena in [25]	99.6078	99.667	99.6078

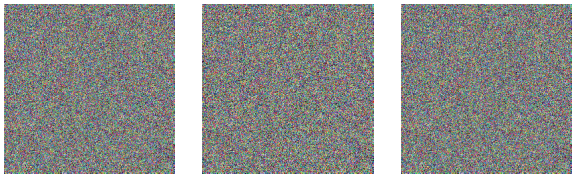
Table 7 UACI of the proposed algorithm for different images %

Image	UACI		
	R	G	B
Baboon	33.4644	33.4690	33.4543
Peppers	33.4560	33.4633	33.4781
Girl	33.4955	33.4847	33.4735
Lena	33.4592	33.4626	33.4548
Lena in [22]	33.2500	33.2800	33.3100
Lena in [23]	33.4108	33.4653	33.4901
Lena in [24]	33.6190	33.5406	33.5727
Lena in [25]	33.5644	33.4458	33.5055

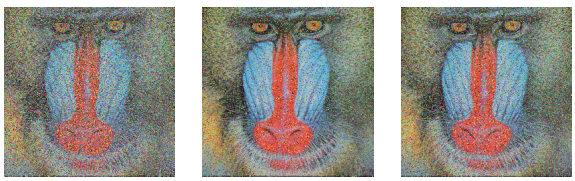
3.6.7 Robustness analysis

Unfortunately, interference is an unavoidable problem when messages are transmitted over open channels. A practical image encryption method should be able to recover images with valid information although the cipher image is affected by noise or corruption.

The most common types of noise include Gaussian white noise, Salt & Pepper noise, and Speckle noise. These three types of noise are overlaid on Fig. 9(i) individually, as shown in Fig. 12(a)−Fig. 12(c). Besides, Fig. 13(a)−Fig. 13(c) present images subjected to different levels of corruption attacks. Decrypt the attacked images and the reconstructed images are exhibited by Fig. 12(d)−Fig. 12(f) and Fig. 13(d)−Fig. 13(f) respectively.

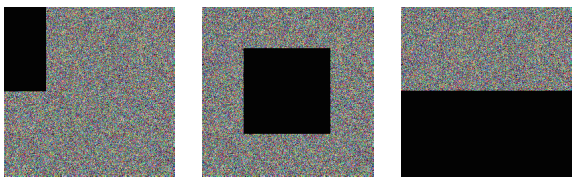


(a) Attacked by Gaussian noise of intensity 3×10^{-6} (b) Attacked by Salt&Pepper noise of intensity 0.1 (c) Attacked by Speckle noise of intensity 5×10^{-6}

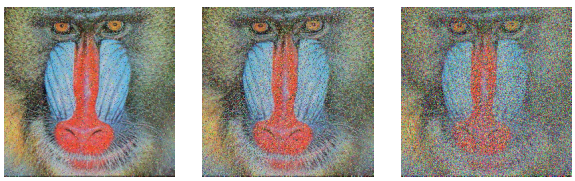


(d) Decrypted image of Fig.12(a) (e) Decrypted image of Fig.12(b) (f) Decrypted image of Fig.12(c)

Fig. 12 Noise attacked images and corresponding decrypted images



(a) 12.5% affected cipher image (b) 25% affected cipher image (c) 50% affected cipher image



(d) Decrypted image of Fig.13(a) (e) Decrypted image of Fig.13(b) (f) Decrypted image of Fig.13(c)

Fig. 13 Cropped cipher images and corresponding decrypted images

To quantitatively measure the quality of the decrypted images under attack, the peak signal to noise ratio (PSNR) is calculated and presented in Table 8 and Table 9. The PSNR is given by

$$\mathrm{PSNR} = 10\lg\left(\frac{(2^8-1)^2}{\mathrm{MSE}}\right) \tag{38}$$

where MSE is the mean square error, and

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(\boldsymbol{P}(i,j)-\boldsymbol{C}(i,j))^2.$$

Table 8 PSNR of images attacked by different noises dB

Image	PSNR		
	Gaussian noise (3×10^{-6})	Salt&Pepper noise (0.1)	Speckle noise (5×10^{-6})
Baboon	12.4589	16.1340	14.5581
Peppers	11.6110	15.2191	13.7482
Girl	10.7942	14.4801	12.9366
Lena	12.1042	15.8221	14.2614

Table 9 PSNR of images attacked by different levels of cropping dB

Image	PSNR		
	12.5% attacked	25% attacked	50% attacked
Baboon	17.9758	14.8644	11.9696
Peppers	17.1096	14.0529	11.0862
Girl	16.1824	13.1821	10.2404
Lena	17.6411	14.5278	11.5540
Image in [25]	17.1595	14.3720	11.3589

It can be clearly seen that meaningful images are restored, reflecting the strong robustness of the present image encryption scheme and its effectiveness against noise and corruption attacks.

3.6.8 Time complexity analysis

Time complexity is essential for evaluating image encryption algorithms. For a color image of size $M\times N\times3$, $O(M\times N\times3/2)$ iterations are required to generate the random number sequences in the proposed scheme. During the dynamic pixel masking operation, $O(M\times N\times9)$ iterations are consumed. Besides, the Brownian motion-based permutation process requires $O(M\times N\times6)$ iterations. In the diffusion process, $O(M\times N\times3)$ iterations are taken for the computation over the finite field GF(2^8). In this condition, the total time complexity of the algorithm in this paper is $O(M\times N\times9)$. Specifically, the runtime of encrypting a 256×256 Lena color image is 2.92 s.

The computational complexity of the proposed method is less than $O(M\times N\times24)$ of [7] and the computational time is less than 3.4095 s. However, our scheme is larger than the time complexity $O(2^{2n})$ of the quantum image encryption method [10], and the image encryption of the

orthogonal Latin cube method [25] with a running time of 0.669 s. In the future, we will continue to work on the balance and improvement between encryption effectiveness and operational efficiency.

4. Conclusions

In brief, a novel memristive chaotic system is designed using a classical memristor and its dynamical behaviour is analysed. This memristive chaotic system is possible to generate three different types of chaotic attractors and to maintain a stable chaotic state over a super wide parameter range of $[3\times10^6,9\times10^6]$. Furthermore, based on the proposed memristive chaotic system, we present a chaotic image encryption scheme. The external key is protected with the Grain-128a algorithm before the image is encrypted, and the information from the plain image is also used to update the initial state of the chaotic system. The scheme involves the cryptographic stream generated by the chaotic system in all stages of the encryption process, including the random extraction and scrambling of pixels, the generation of the Brownian motion matrix and the diffusion operation over the finite field $GF(2^8)$. Simulation test results reveal that the image encryption scheme proposed in this paper has a larger key space and better security performance.

References

[1] MATTHEWS R. On the derivation of a “Chaotic” encryption algorithm. Cryptologia, 1989, 13(1): 29–42.

[2] FRIDRICH J. Symmetric ciphers based on two-dimensional chaotic maps. International Journal of Bifurcation and Chaos, 1998, 8(6): 1259–1284.

[3] SU Y G, TANG C, LI B Y, et al. Single-lens Fourier-transform-based optical color image encryption using dual two-dimensional chaotic maps and the Fresnel transform. Applied Optics, 2017, 56(3): 498–505.

[4] KHAN M A, AHMAD J, JAVAID Q, et al. An efficient and secure partial image encryption for wireless multimedia sensor networks using discrete wavelet transform, chaotic maps and substitution box. Journal of Modern Optics, 2017, 64(5): 531–540.

[5] WANG X Y, XU D H. A novel image encryption scheme based on Brownian motion and PWLCM chaotic system. Nonlinear Dynamics, 2014, 75: 345–353.

[6] HUSSAIN I, SHAH T, GONDAL M A, et al. A novel image encryption algorithm based on chaotic maps and $GF(2^8)$ exponent transformation. Nonlinear Dynamics, 2013, 72: 399–406.

[7] ZHANG L M, SUN K H, LIU W H, et al. A novel color image encryption scheme using fractional-order hyperchaotic system and DNA sequence operations. Chinese Physics B, 2017, 26(10): 100504.

[8] NASKAR P K, BHATTACHARYYA S, NANDY D, et al. A robust image encryption scheme using chaotic tent map and cellular automata. Nonlinear Dynamics, 2020, 100(3): 2877–2898.

[9] NJITACKE Z T, ISAAC S D, NESTOR T, et al. Window of multistability and its control in a simple 3D Hopfield neural network: application to biomedical image encryption. Neural Computing and Applications, 2021, 33: 6733–6752.

[10] GONG L H, HE X T, CHENG S, et al. Quantum image encryption algorithm based on quantum image XOR operations. International Journal of Theoretical Physics, 2016, 55: 3234–3250.

[11] NATIQ H, AL-SAIDI N M G, SAID M R M, et al. A new hyperchaotic map and its application for image encryption. European Physical Journal Plus, 2018, 133(1): 6.

[12] WANG X Y, WANG Y, WANG S W, et al. A novel pseudo-random coupled LP spatiotemporal chaos and its application in imageencryption. Chinese Physics B, 2018, 27(11): 110502.

[13] LIU H J, ZHANG Y Q, KADIR A, et al. Image encryption using complex hyper chaotic system by injecting impulse into parameters. Applied Mathematics and Computation, 2019, 360(1): 83−93.

[14] STRUKOV D B, SNIDER G S, STEWART D R, et al. The missing memristor found. Nature, 2008, 453(7191): 80–83.

[15] PENG G Y, MIN F H. Multistability analysis, circuit implementations and application in image encryption of a novel memristive chaotic circuit. Nonlinear Dynamics, 2017, 90(3): 1607–1625.

[16] LI C L, LI Z Y, FENG W, et al. Dynamical behavior and image encryption application of a memristor-based circuit system. International Journal of Electronics and Communications, 2019, 110: 152861.

[17] CHUA L O, KANG S M. Memristive devices and systems. Proccedings of the IEEE, 1976, 64(2): 209–223.

[18] BAO B C, XU J P, ZHOU G H, et al. Chaotic memristive circuit: equivalent circuit realization and dynamical analysis. Chinese Physics B, 2011, 20(12): 120502.

[19] AGREN M, HELL M, JOHANSSON T. Grain-128a: a new version of Grain-128 with optional authentication. International Journal of Wireless and Mobile Computing, 2011, 5(1): 48–59.

[20] HELL M, JOHANSSON T, MEIER W. Grain– a stream cipher for constrained environments. International Journal of Wireless and Mobile Computing, 2006, 2(1): 86–93.

[21] HELL M, JOHANSSON T, MAXIMOV A, et al. A stream cipher proposal: Grain-128. Proc. of the IEEE International Symposium on Information Theory, 2006: 1614–1618.

[22] CHAI X L, GAN Z H, LU Y, et al. A novel color image encryption algorithm based on genetic recombination and the four-dimensional memristive hyperchaotic system. Chinese Physics B, 2016, 25(10): 100503.

[23] HASANZADEH E, YAGHOOBI M. A novel color image encryption algorithm based on substitution box and hyperchaotic system with fractal keys. Multimedia Tools and Applications, 2020, 79: 7279–7297.

[24] LIU H J, KADIR A, XU C B. Color image encryption with cipher feedback and coupling chaotic map. International Journal of Bifurcation and Chaos, 2020, 30(12): 2050173.

[25] ZHOU J, ZHOU N R, GONG L H. Fast color image encryption scheme based on 3D orthogonal Latin squares and matching matrix. Optics & Laser Technology, 2020, 131: 106437.

Biographies

HUANG Lilian was born in 1972. She received her M.E. and Ph.D. degrees in navigation, guidance and control from Harbin Institute of Technology, Harbin, China, in 2002 and 2005, respectively. She is a a professor in Harbin Engineering University. Her research interests are in the study and application of nonlinear chaotic systems.
E-mail: lilian_huang@163.com

SUN Yi was born in 1994. She is currently pursuing her M.S. degree in Harbin Engineering University. Her research interests are the study of nonlinear chaotic systems and their applications in image encryption.
E-mail: sun_yi@hrbeu.edu.cn

XIANG Jianhong was born in 1977. He received his M.E. and Ph.D. degrees in communication and information system from Harbin Engineering University, Harbin, China, in 2006 and 2009, respectively. He is a professor in Harbin Engineering University. His research interests are communication and signal processing, deep learning and artificial intelligence.
E-mail: xiangjianhong@hrbeu.edu.cn

WANG Linyu was born in 1977. She received her M.E. and Ph.D. degrees in communication and information system from Harbin Engineering University, Harbin, China, in 2003 and 2006, respectively. She is a professor in Harbin Engineering University. Her research interests are communications and information processing.
E-mail: wanglinyu@hrbeu.edu.cn