Mustansiriyah Journal of Pure and Applied Sciences

www.MJPAS.com

chaotic image encryption method based on three-dimensional nonlinear system

Y. Wanbo[1], Z. Qinwu[2,*], Z. Qingjian[3]

[1,2]Department of Computer Science, School of Information Engineering, Dalian University ([1]yu_wb@126.com, [2]1796585674@qq.com)
[3]Electronic Science and Technology, School of Testing and Optoelectronic Engineering, Nanchang University of Aeronautics and Astronautics ([3]zengqingjian1986@163.com)

A B S T R A C T

Due to the simple structure of one-dimensional chaotic mapping, there may be problems such as lack of security in applying to image encryption; so many researchers have proposed some high-dimensional chaotic mapping to apply to image encryption. However, high-dimensional chaotic mapping may be difficult to be applied in practice because of its complex structure and high cost. So this paper proposes an image encryption method with a relatively simple structure, low cost, and good encryption effect. The Proposed method combines a chaotic system with 3D nonlinear function iterative systems to encrypt images. First, a sequence is generated iteratively according to the given chaotic system, and then the sequence is used to call multiple three-dimensional nonlinear function systems in the group and perform operations in turn; then the iterative operations are performed according to the image size; finally, the chaotic sequence generated by the iteration and the grayscale image is subjected to an XOR operation. The phase diagram and bifurcation diagram determine the iterative sequence to have chaotic solid properties. The algorithm performs only a simple XOR operation, but the experimental results show the scheme is feasible.

Keywords*: image encryption;Three-dimensional nonlinear iterative functions; Chaos; XOR encryption*

1. Introduction

With the rapid development of network communication technology, images are more and more frequently used as network communication carriers. Some images will involve personal private information, even national security secrets, etc., so the confidentiality of images becomes increasingly essential. Since chaos is characterized by randomness, initial value sensitivity, and unpredictability, more and more scholars have proposed chaos-based image encryption schemes. For example, Huang, H., Cheng, D (2022) [1] proposed an algorithm for the joint encryption and compression of discrete cosine transform(DCT) and hyperchaotic systems. In addition, researchers have given a series of image encryption methods for chaotic systems combined with DNA sequences, as documented in the literatures [2-8].

The functions can be classified into linear and nonlinear functions from linear and nonlinear perspectives, and in general, linear functions are less computationally intensive and more efficient to encrypt. There have also been many research works in this area; for example, Yuan, Jouwei, et al. (2010) [11] proposed extending the one-dimensional segmented linear mapping to two-dimensional segmented linear mapping

and encrypting the two sequences generated using the two-dimensional segmented linear mapping. Jing Li et al. (2019) [12] proposed an encryption scheme that enhances image security and extends the critical space by combining a two-dimensional logistic system with a new two-dimensional discrete system; experimental results show that the algorithm has the advantages of easy implementation and sizeable critical space.

From the point of view of chaotic mapping dimension, nowadays, chaotic mapping can be categorized into one-dimensional and high-dimensional chaotic mapping, respectively. Due to the more straightforward structure of one-dimensional chaotic mapping, some researchers have proposed one-dimensional chaotic mapping to be applied to image encryption [9-10]. Ban Dohan et al.(2020) [9] proposed an efficient encryption algorithm for images that reduces the number of rounds of encryption and further improves the efficiency of encryption compared to traditional methods. Pak C et al.(2017) [10] produced a simple and effective chaotic system by utilizing the difference of different output sequences of two identical existing one-dimensional chaotic mappings.

Some researchers also prefer to use high-dimensional chaotic systems to encrypt images. Song Liu et al. (2022)[13] proposed a third-order segmented linear chaotic system based on Shilnikov's theorem for heterodyne rings and then applied the system to generate chaotic data sequences and constructed the dislocation matrix and random diffusion matrix to encrypt images. Yu W et al.(2023) improved the Clifford system and gave a series of trigonometric systems to generate chaotic sequences with better chaotic properties for image encryption [14, 15]; Yu W et al. also designed a series of novel and lightweight systems to generate chaotic sequences, which realize image encryption and various indexes are better than many similar articles [16,17]. Yu F et al. (2021)[18] proposed a new 5D memristive exponential hyperchaotic system (MEHS), and the simulation results show that the algorithm has a good encryption effect and anti-attack ability. Yu F et al. (2023)[19] also proposed a new method to construct a composite hyperbolic tangent-cubic nonlinear function to generate a multivolume chaotic attractor in a canonical Chua circuit. It is easy to find that the structure and parameters of high-dimensional chaotic systems are more complex. However, the critical space is small and increases the complexity and implementation difficulty of the algorithm.

2 Related works

Chaos mapping has a wide range of application value in engineering applications. It can be applied not only in signal processing but also in image compression, image encryption, and other fields. Image encryption mainly relies on the random sequence generated by chaotic mapping iteration to encrypt the image. Because of the randomness and unpredictability of the chaotic sequence, it is challenging to crack the ciphertext image.Reading a large amount of literature reveals that not many chaotic systems are used to encrypt data, such as images. Since it is considering that there are fewer chaotic systems, DNA sequences, some transformation methods, image chunking and layering methods, etc., have been

added. Therefore, constructing new chaotic systems and generating more sequences is a critical problem.

The chaotic sequence construction method of reference [20] is first to generate N groups of M-element linear functions each, and then use a chaotic system (e.g., a Logistic system) to generate a chaotic sequence, adjust the sequence values to integers from 1 to N using an expanded multiplicative remainder method, and iterate based on these integers by calling a particular set of linear functions in the N groups. Starting from an initial value, the iterations are substituted to the set of functions determined by the chaotic logistic sequence to obtain a sequence with powerful chaotic properties when the coefficients of the linear functions satisfy a relatively broad condition. The obtained chaotic sequence is used for image encryption, and the calculated various metrics can reach or approach the ideal value, confirming that this is a simple and efficient method for chaotic sequence generation. This method has yet to be seen in the existing literature.

Based on the literature [20], this paper implements and successfully uses a chaotic sequence generation method based on three-dimensional nonlinear functions for image encryption. The algorithm has the advantages of simple structure and good encryption effect of reference [20] but also has higher security and more difficulty deciphering.

3 Improvement of chaotic systems

3.1 Logistic map

The Logistic map[21] can be expressed as (1) :

$$f(x) = \mu * x * (1 - x) \tag{1}$$

where $\mu \in [0, 4]$, $x \in [0, 1]$. The logistic mapping is chaotic when $\mu \in [3.5699456, 4]$. Because Logistic mapping has advantages such as knot simplicity, it is very widely used in encryption.

In this paper, we take Logistic mapping as an example; any system with strong chaotic properties can be used as this "guided" chaotic system, such as a sinusoidal function chaotic system, Lorentzian chaotic system, and so on.

3.2 Three-dimensional nonlinear functions

A (set of) three-dimensional nonlinear iterative functions can be represented by system (2).

$$\begin{cases} F_{n1} = \sum_{k11=1}^{m1} a_{k11} * x^{k11} + \sum_{k12=1}^{m2} b_{k12} * y^{k12} + \sum_{k13=1}^{m3} c_{k13} * z^{k13} + \sum_{k14=1}^{m1}\sum_{k15=1}^{m2} d_{k14k15} * x^{k14}y^{k15} + \sum_{k1=1}^{m1}\sum_{k3=1}^{m3} e_{k16k17} * x^{k16}z^{k17} \\ \qquad + \sum_{k1=1}^{m2}\sum_{k3=1}^{m3} f_{k18k19} * y^{k18}z^{k19} + g_{n1} \\ F_{n2} = \sum_{k21=1}^{m1} a_{k21} * x^{k21} + \sum_{k22=1}^{m2} b_{k22} * y^{k22} + \sum_{k23=1}^{m3} c_{k23} * z^{k23} + \sum_{k24=1}^{m1}\sum_{k25=1}^{m2} d_{k24k25} * x^{k24}y^{k25} + \sum_{k26=1}^{m1}\sum_{k27=1}^{m3} e_{k26k27} * x^{k26}z^{k27} \\ \qquad + \sum_{k28=2}^{m2}\sum_{k29=1}^{m3} f1_{k28k29} * y^{k28}z^{k29} + g_{n2} \\ F_{n3} = \sum_{k31=1}^{m1} a_{k31} * x^{k31} + \sum_{k32=1}^{m2} b_{k32} * y^{k32} + \sum_{k33=1}^{m3} c_{k33} * z^{k33} + \sum_{k34=1}^{m1}\sum_{k35=1}^{m2} d_{k34k35} * x^{k34}y^{k35} + \sum_{k36=1}^{m1}\sum_{k37=1}^{m3} e_{361k37} * x^{k36}z^{k37} \\ \qquad + \sum_{k38=2}^{m2}\sum_{k39=1}^{m3} f_{38k39} * y^{k38}z^{k39} + g_{n3} \end{cases} \quad (2)$$

Where $m1, m2, m3$ are integers in [1, 6]; x, y, z are variables; $a_{kij}, b_{kij}, c_{kij}, d_{kij}$, $e_{kij}, f_{kij}, i = 1, 2, 3, j = 1,2\cdots,8,9$ are random parameters belonging to [0, 1]; $g_{ni}, i = 1, 2, 3$ is a random constant belonging to [0, 1], respectively. Let n be a natural number, then there are a total of n sets of three-dimensional nonlinear functions, and in this paper, n is 3.

In this paper, the algorithm firstly iterates the Logistic chaotic system; then, according to the random sequence generated after the iteration, it selects the corresponding three-dimensional nonlinear function expression and carries out the operation. and performs the operations such as remainder and integer on the values after the operation, and stores the values after the operation into the arrays **A1**, **B1**, and **C1** respectively; then iterates according to the size of the image; after the iteration, the three chaotic After the iteration, the three chaotic arrays **A1**, **B1**, **C1** are converted into a three-dimensional array **H.** Finally, the three-dimensional array **H** is used to encrypt the image differently.

Compared with reference [20], the 3D surface images of nonlinear functions are more curved than the 3D images of linear functions, such that the image encryption algorithm based on nonlinear functions is more difficult to decipher than the image encryption algorithm based on linear functions. For example, Figure 1 shows the 3D function maps of nine different nonlinear functions, where the values of m1, m2, and m3 are 3, 3, and 0, respectively; the x, y, and z axes are the values of the independent variables x, y, and the response variable F, respectively. A 3D point iteratively "walks" through these surfaces (in groups of three), and its trajectory would be more challenging to track and recover if more information were unknown.

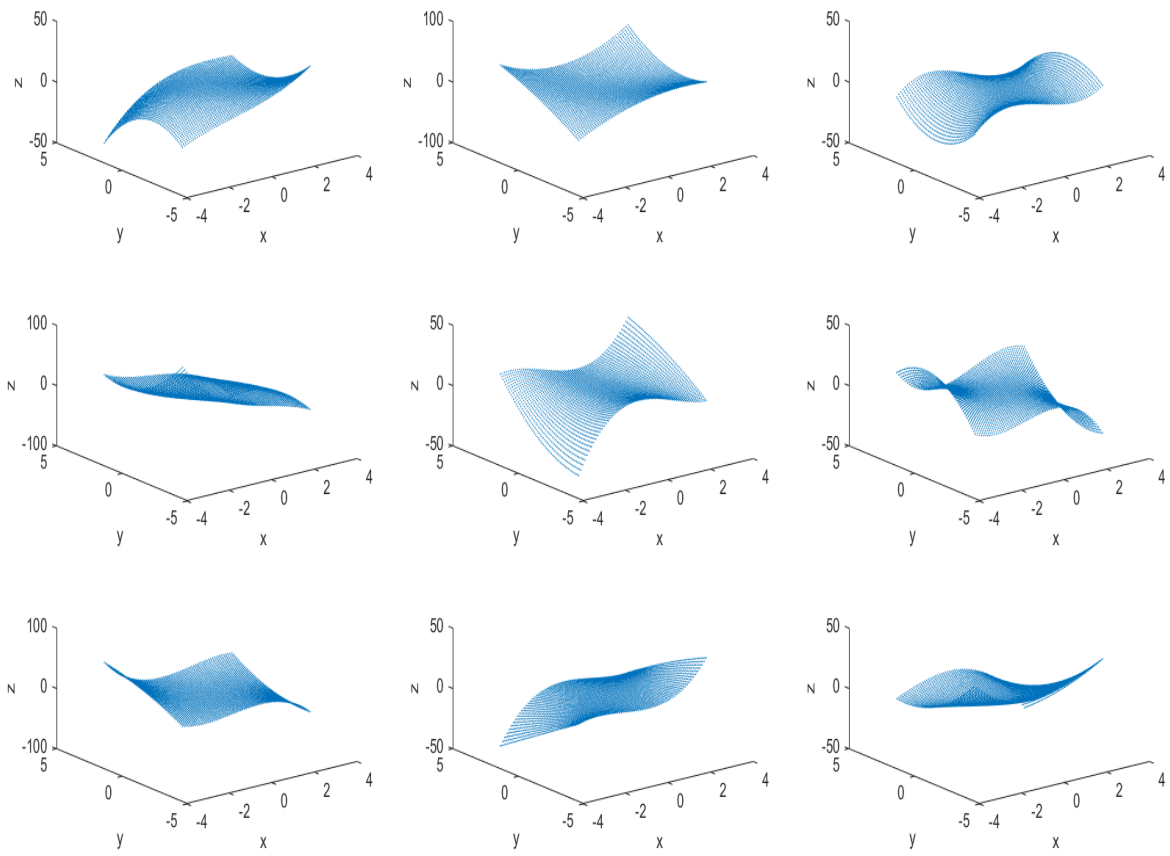
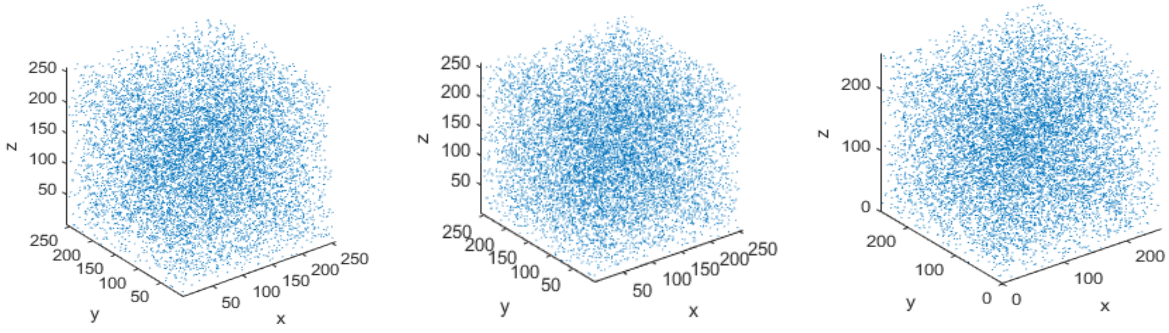


Figure 1: 3D graphs of 9 different nonlinear

4 New chaotic sequence generation method and analysis

4.1 Phase diagram of the system

If the chaotic system occupies ample space and is dense in the phase diagram, the chaotic system will generally have good chaotic properties.Figure 2 shows the phase diagrams for different indices $m1, \mathrm{m2}$, and $m3$ in expression (2), representing the variation of the variable z in expression (2) with the variables x, y. among them, the values of indices $m1, m2$, and $m3$ in Figure 2(a) are 2,2,1, respectively; the values of indices $m1, m2$, and $m3$ in Figure 2(b) are 3,3,1, respectively; and the values of indices $m1, m2$, and $m3$ in Figure 2(c) are 5,4,1, respectively. From Figure 2, it can be seen that the phase diagram is more spatial, showing that the system has good chaotic properties.



(a) (b) (c)

Figure 2:Phase diagrams for different indices m1, m2, m3

4.2 Bifurcation diagram of the system

Bifurcation diagrams usually show the state of motion of a chaotic system by controlling the parameters. The specific steps are as follows:

Step 1: Set the system parameter $a_{kij}, b_{kij}, c_{kij}, d_{kij}$, e_{kij}, f_{kij}, i $= 1, 2, 3,$ j $= 1,2 \cdots 8,9$ of expression (2) to the system parameter corresponding to the phase diagram (4.1 Parameters for Plotting the Phase Diagram); the initial value of the independent variable x in expression (1) is also set to the value corresponding to the phase diagram, and set the values of the indices m1,m2,m3 of expression (2).

Step 2: According to the initial value x of the chaotic mapping in equation (1), the operation is performed to obtain $f(x)$, and after the operation, the value of f(x) is multiplied by several times (e.g., 100), then the ceil() function is used to round up, and then the Mod() function is used to carry out the remainder of the operation on n, and the value obtained after the set of operations is noted as *sign1*.

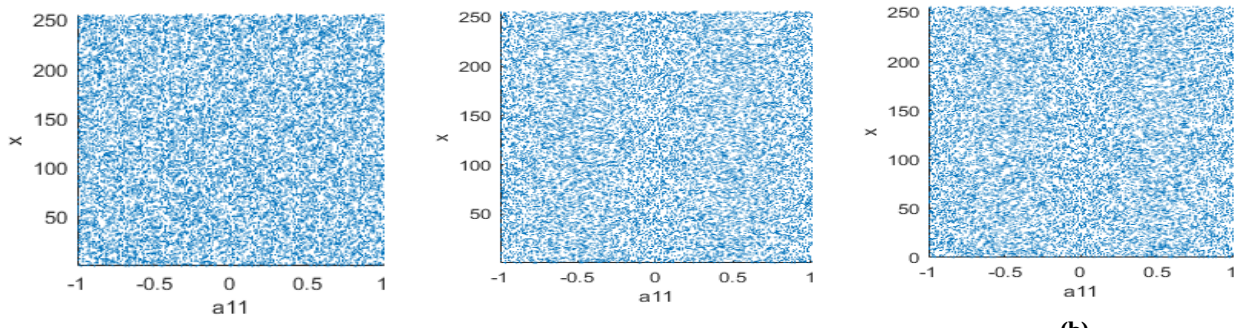
Step 3: A specific ternary nonlinear function expression for the iterative system is selected based on the value of *sign1*, e.g., if the value of sign1 is k, then the kth expression is used, and the value of the function respondent F_{n1}、F_{n2}、F_{n3} is computed.

Step 4 Plot the two-dimensional point plots of the variables a11, x (independent variable x of expression (2)) using Matlab's line() function.

Step 5: First $f(x)$ as x (expression (1) of the independent variable x) substitution, F_{n1}、F_{n2}、F_{n3} as *x, y,* z respectively (expression (2) of the independent variable x, y, z) substitution (the specific code is,$x = f(x), x = F_{n1}$,y=F_{n2}, $z = F_{n3}$); finally, the independent variable x, y, z expanded by several times and then carry out the remainder of the operation.

Step 6: Set the interval and range of system parameter iterations, and let *h1* be a natural number. Perform *h1* x $\frac{\text{range}}{\text{interval}}$ iterations for steps 2-step 5.

Figure 3 shows the bifurcation diagrams for different indices $m1, m2$, and $m3$ in expression (2); it represents the variation of the variable x in expression (2) with the system parameter $a11$. Among them, the values of indices $m1, m2$, and $m3$ in Fig. 3(a) are 2,2,1, respectively; the values of indices $m1, m2$, and $m3$ in Fig. 3(b) are 3,3,1, respectively; and the values of indices $m1, m2$, and $m3$ in Fig. 3(c) are 5,4,1, respectively. It can be seen from the plots that the system has an extensive output range of indices in different ranges of values and better traversal rows. This indicates that the system has a solid chaotic property.



(a) **(b)** **(b)**

Figure 3:Bifurcation diagrams for different indices m1, m2,

5. Chaotic Sequences for Image Encryption

This paper proposes an encryption method based on a chaotic system of iterative three-dimensional nonlinear functions. The algorithm mainly uses the sequence iteratively generated by the three-dimensional nonlinear function system to perform a single dissimilarity operation on the image's pixel values. This iterative sequence generation method is equivalent to compounding a classically chaotic system (e.g., logistic mapping) with a three-dimensional nonlinear iterative system, which is more "chaotic," harder to break, and more effective in encryption.

5.1 Encryption Process

The specific steps of the encryption process are as follows:

Step 1: According to system(2), set up n different three-dimensional nonlinear functions of the chaotic system; n is 3 in this paper.

Set the indices $m1, m2, m3$to random integers of [1,6]. The system parameters $a_{kij}, b_{kij}, c_{kij}, d_{kij}$, $e_{kij}, f_{kij}, i = 1, 2, 3, j = 1,2 \cdots 8, 9$ are set to a random number within [0, 1]; the system constant $g_{ni}, i = 1, 2, 3$ is set to a random number within [0, 1]. Moreover, set the initial values of the variables x, y, and z to random numbers within [0, 1], respectively (the initial values of x, y, and z in this paper are 0.33, 0.678, and 0.976(x, y, z is a random number from 0 to 1, not fixed. (This article just take $x = 0.33; y = 0.678; z = 0.976$ as an example)), respectively).

Step 1 pseudo code: m1,m2,m3<--randi([1,6]);

$a_{kij}, b_{kij}, c_{kij}, d_{kij}$， e_{kij}, f_{kij}, (i = 1, 2, 3, j = 1,2 ⋯,8,9)<--rand(0,1)

g_{ni}, (i = 1, 2, 3)<--rand(0,1);

x,y,z<--rand(0,1);

Step 2: Set the logistic mapping in expression (1): $y = \mu * x * (1 - x)$

Set the value of μ to a random number of μ ∈ [3.5699456, 4] (the value of μ in this paper is 3.8). The initial value of the variable x is a random number from 0 to 1. After the logistic mapping operation, the variable y is expanded a few times, and the remainder operation is performed on n. (In this paper, the variable y is multiplied by 100, and then the value of y is set to an integer using the ceil()(Rounding upwards, towards positive infinity) function, and finally, the remainder operation is performed on y using the mod() position on 3.); Finally, the value after the operation is recorded as $ch1$.

Step 2 pseudo code: μ <--rand(3.5699456, 4);

n<--4;

x<--rand(0,1);

y<--μ ∗ x ∗ (1 − x); //After the Logotic chaotic mapping operation is completed, the rounding operation is performed.

y<--ceil(100*y)

y<--mod(y,n);

ch1<--y;

Step 3: Select different 3D nonlinear functions in expression (2) according to the value of $ch1$. After the operation of the selected three-dimensional nonlinear function, the values of $Fn1, Fn2$, and $Fn3$ are expanded several times. Then the remainder operation is performed (in this paper, the importance of $Fn1, Fn2$, and $Fn3$ are grown 1000 times, and then the remainder operation is completed). After the procedure, the $Fn1, Fn2$, and $Fn3$ values are stored in the arrays **A1**, **A2**, and **A3**, respectively.

Step 3 pseudo code: If ch1<--n //The value of n is 1,2,3,4; the corresponding three-dimensional nonlinear function system is selected according to the value of n.

$$F_{n1} <-- \sum_{k11=1}^{m1} a_{k11} * x^{k11} + \sum_{k12=1}^{m2} b_{k12} * y^{k12} + \sum_{k13=1}^{m3} c_{k13} * z^{k13} + \sum_{k14=1}^{m1}\sum_{k15=1}^{m2} d_{k14k15} * x^{k14}y^{k15} + \sum_{k16=1}^{m1}\sum_{k17=1}^{m3} e_{k16k17} * x^{k16}z^{k17} + \sum_{k18=1}^{m2}\sum_{k19=1}^{m3} f_{k18k19} * y^{k18}z^{k19} + g_{n1}$$

$$F_{n2} <-- \sum_{k21=1}^{m1} a_{k21} * x^{k21} + \sum_{k22=1}^{m2} b_{k22} * y^{k22} + \sum_{k23=1}^{m3} c_{k23} * z^{k23} + \sum_{k24=1}^{m1}\sum_{k25=1}^{m2} d_{k24k25} * x^{k24}y^{k25} + \sum_{k26=1}^{m1}\sum_{k27=1}^{m3} e_{k26k27} * x^{k26}z^{k27} + \sum_{k28=1}^{m2}\sum_{k29=1}^{m3} f_{k28k29} * y^{k28}z^{k29} + g_{n2}$$

$$F_{n3} <-- \sum_{k31=1}^{m1} a_{k31} * x^{k31} + \sum_{k32=1}^{m2} b_{k32} * y^{k32} + \sum_{k33=1}^{m3} c_{k33} * z^{k33} + \sum_{k34=1}^{m1}\sum_{k35=1}^{m2} d_{k34k35} * x^{k34}y^{k35} + \sum_{k36=1}^{m1}\sum_{k37=1}^{m3} e_{361k37} * x^{k36}z^{k37} + \sum_{k38=1}^{m2}\sum_{k39=1}^{m3} f_{k38k39} * y^{k38}z^{k39} + g_{n3}$$

F_{n1}<--mod(F_{n1}*1000,256); //After the operation is completed, the remainder and rounding operation is carried out.

F_{n2}<--mod(F_{n1}*1000,256);

F_{n3}<--mod(F_{n1}*1000,256);

A1<--F_{n1}; A2<--F_{n2}; A3<--F_{n3}; //Store the chaotic sequences after the operation in the arrays A1, A2, and A3, respectively.

Step 4: Assume the image size is *M*×*N*, and perform *M*×*N* iterations for steps 2 and step 3.

The specific method of step 2 iteration is to make $y = x1$. The process of step 3 iteration is to make x = $Fn1, y = Fn2, z = Fn3$.

Step 4 pseudo code:for i<--1 to M //Iterate based on image size

for j<--1 to N do

y<--x //Step 2 iterative approach

x<--F_{n1}; y<--F_{n2}; z<--F_{n3}; //Step 3 iterative approach

end;

end;

Step 5 : XOR encryption

After the iteration is completed, the image to be encrypted is invoked using the imread ( ) function. Then, the chaotic array **A1**, **A2**, and **A3** is converted into a two-dimensional array mi _ a. Then, the three-dimensional chaotic array mi _ a performs an XOR encryption on the transferred image.

Step 5 pseudo code: P1<--imread('Path to the image file to be encrypted') //Load the image to be encrypted

mi_a<--zero(M,N,3);//Converting the produced chaotic sequence into a three-dimensional matrix mi_a

mi_a(;;1)<--A1; // Also assign **A2**,**A3** chaotic arrays to mi_a(;;1)

mi_a(;;2)<--A2; //Also assign **A1**,**A3** chaotic arrays to mi_a(;;2)

for i<--1 to M //Encrypting an image with image size **M**×**N**

for j<--1 to N do

encrypted(i,j)<--bitxor(P1(i,j),mi_a(i,j)); //The bitxor function is used to XOR encrypt the image once.

end;

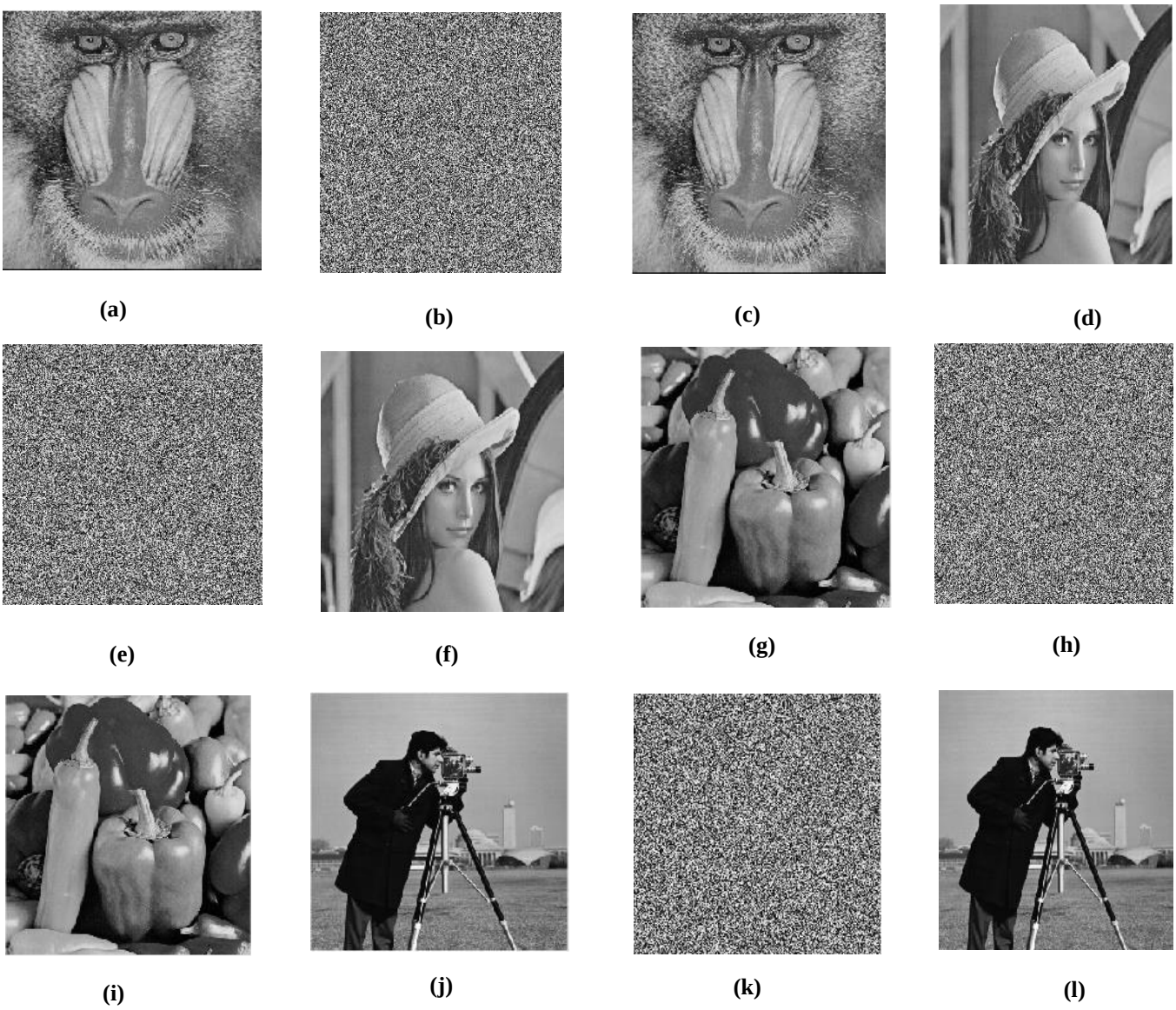
end;

5.2 Decryption process

The image's decryption algorithm is the encryption algorithm's reverse process. It mainly re-utilizes the 'bitxor()' function to perform a 'Xor' operation on the 3D chaotic sequence m and the ciphertext image mi_a.

5.3 Encryption results

Although the encryption algorithm in this paper is straightforward, with only one XOR operation, the encryption effect is excellent. In this paper, we test the encryption effect of images with different values of indices $m1, m2$, and $m3$ in expression (2). Among them, Figure 4 shows the test results of different indices in 256×256 size; Figure 5 shows the test results of different indices in 512×512 size; Fig. 6 shows the test results of different indices in all-black and all-white images. The results show that the method shows a good encryption effect regardless of the all-black image, all-white image, or different sizes.

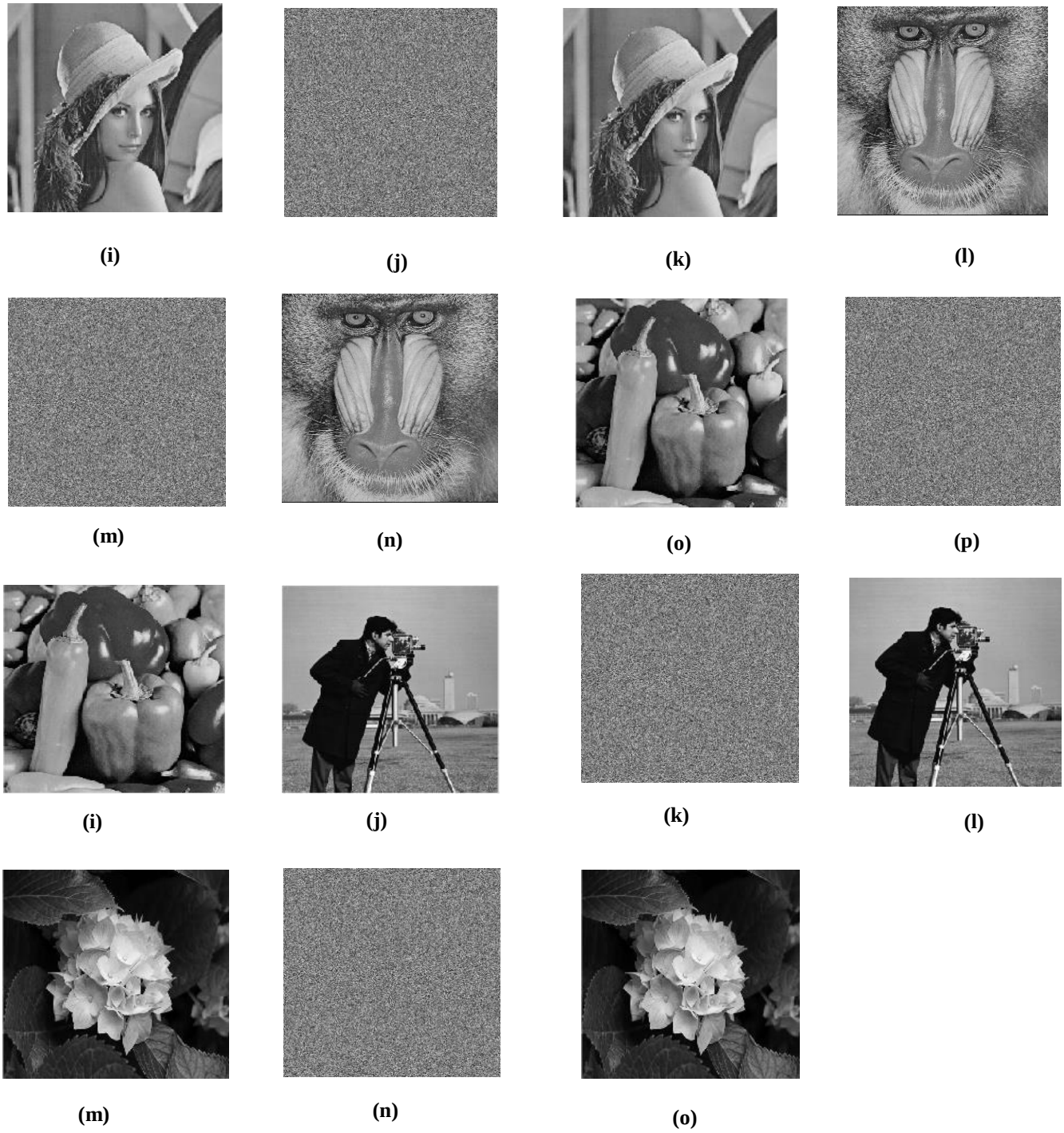


(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

(i) **(j)** **(k)** **(l)**

Figure 4: Test results for image size of 256×256



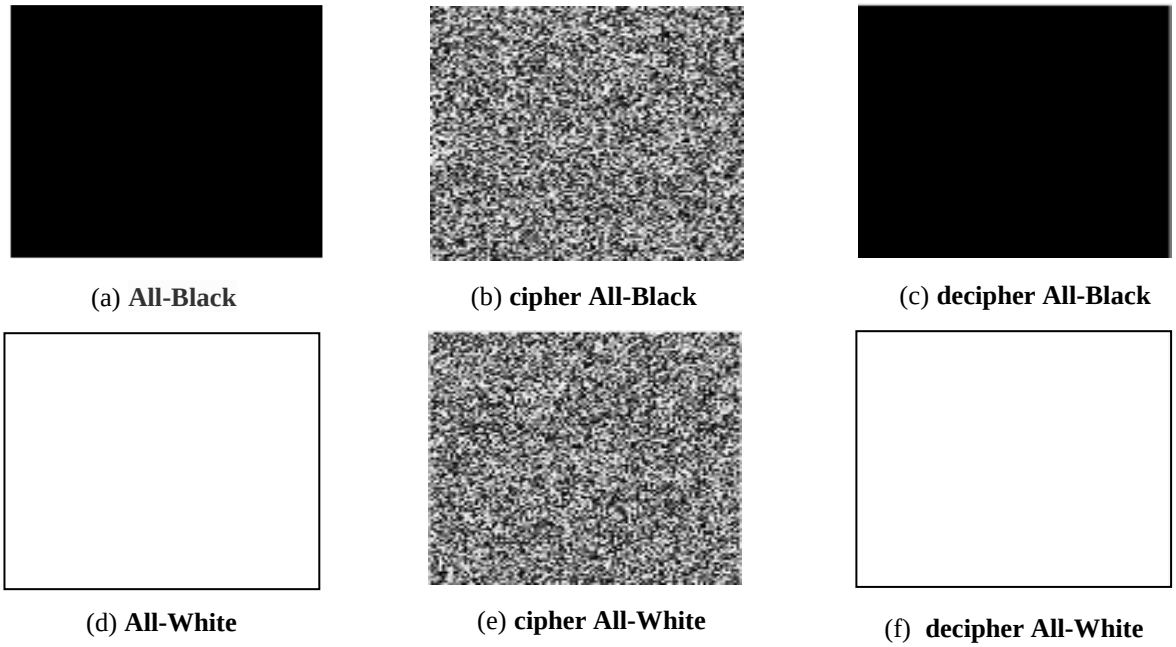
(i) **(j)** **(k)** **(l)**

(m) **(n)** **(o)** **(p)**

(i) **(j)** **(k)** **(l)**

(m) **(n)** **(o)**

Figure 5: Test results for an image size of 512 x 512



(a) **All-Black** (b) **cipher All-Black** (c) **decipher All-Black**

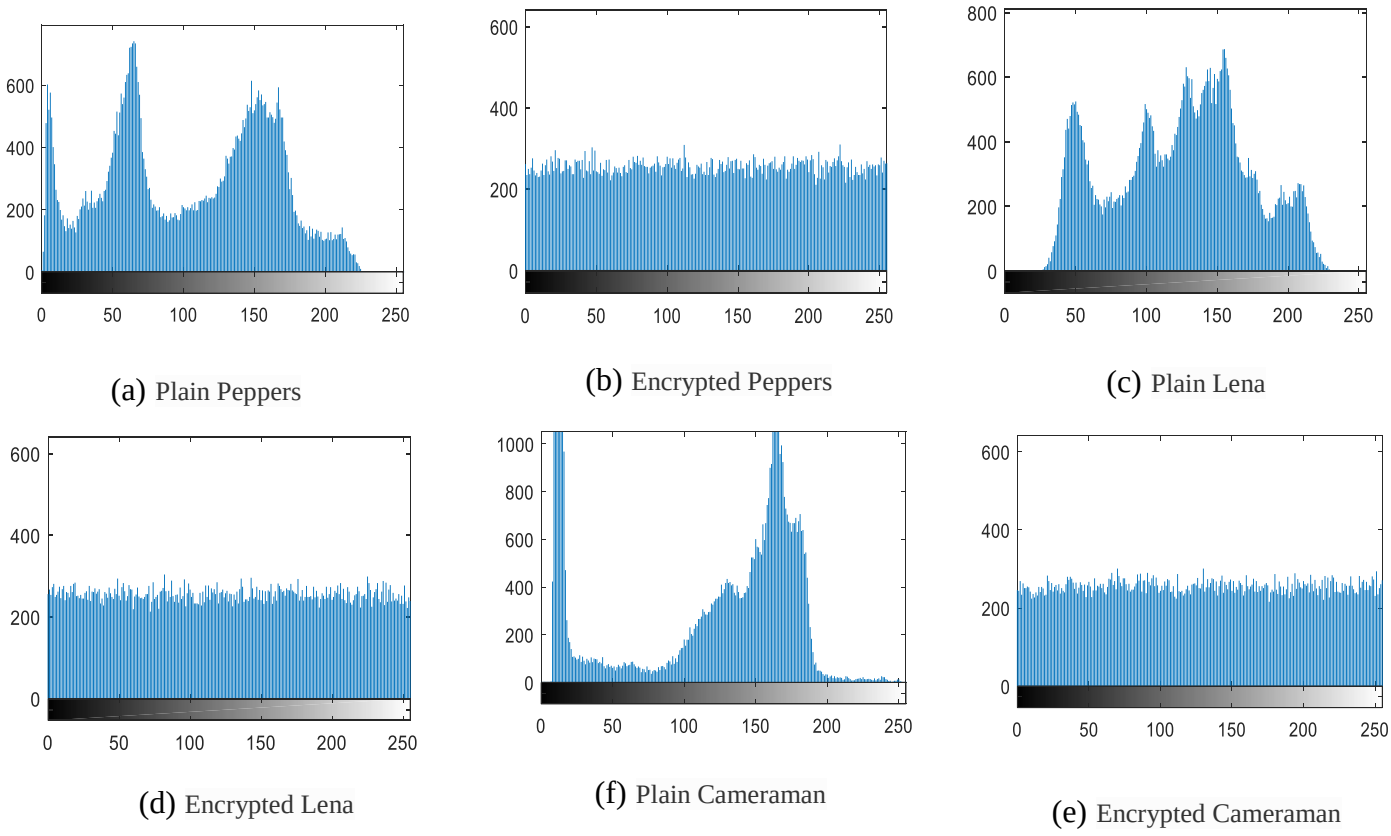
(d) **All-White** (e) **cipher All-White** (f) **decipher All-White**

Figure 6: All-black and all-white images in 256×256 size

6 Encryption effect analysis

6.1 Histogram Characterization

Histogram is an important metric for evaluating the effectiveness of image encryption Figure 7 shows the histograms of different indexes $m1, m2$, and $m3$ in different images, and the image size is 256 × 256. Among them, Figure 7 (a) and (b) show the histograms of the original and encrypted images of gray Peppers, respectively, and the values of the exponents $m1, m2$, and $m3$ are 2, 2, and 1, respectively. Figures 7 (c) and (d) show the histograms of the original and encrypted images of the gray Lena, respectively. The values of the exponents $m1, m2$, and $m3$ are 3, 3, and 1, respectively. Figures 7 (e) and (f) show the histograms of the original and encrypted gray Cameraman images, respectively. The values of the exponents $m1, m2$, and $m3$ are 5, 4, and 1, respectively. From the experimental results, it can be seen that the histograms are more balanced after encryption. This indicates that the pixel values of the ciphertext are evenly distributed. Hence, it shows that the encryption scheme is effective against statistical attacks [16].



(a) Plain Peppers (b) Encrypted Peppers (c) Plain Lena

(d) Encrypted Lena (f) Plain Cameraman (e) Encrypted Cameraman

Figure 7: Histograms of the plain and cipher images

Neighboring pixel correlation is an essential metric for image encryption performance; the closer the correlation of its cipher image is to 0, the better its encryption is. Its calculation formula is as follows [14]:

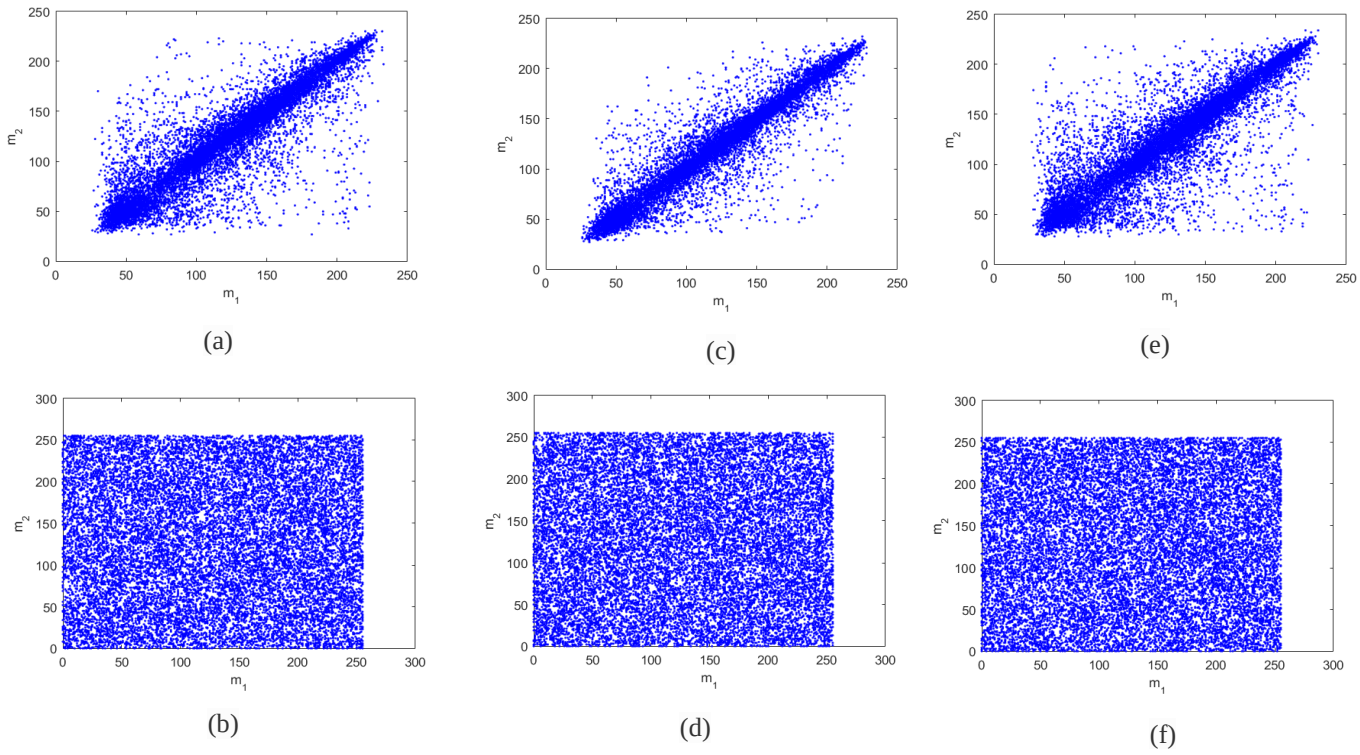
$$R_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{3}$$

$$\mathrm{E}(x) = \frac{1}{N}\sum_{i=1}^{N} xi \tag{4}$$

$$\mathrm{D}(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - E(x)\right)^2 \tag{5}$$

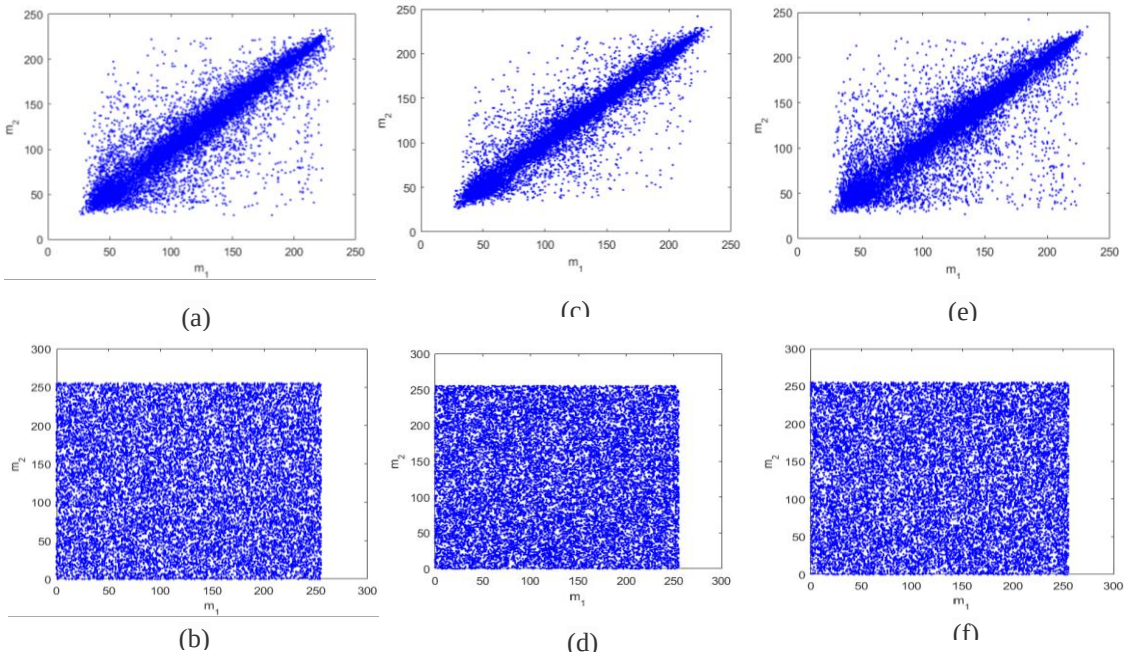
$$\operatorname{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{6}$$

In the above equation, y is the neighboring pixels of x, and N calculates the total number of pixels. From the original Lena gray image and the encrypted Lena gray image, 20,000 pairs of adjacent pixels in diagonal, horizontal, and vertical directions are randomly selected, with an image size is 256×256. Figure 8 shows the correlation distribution of expression (2) with indices $m1, m2$, and $m3$ of 2,2,1, respectively. Figure 9 shows the correlation distribution of expression (2) with indices $m1, m2$, and $m3$ of 3, 3, and 1, respectively. Figure 10 shows the correlation distribution of the expression (2) with indices $m1, m2$, and $m3$ of 5, 4, and 1, respectively. Table 1 lists the experimental results of the algorithm for the correlation analysis of different images. Table 2 compares the correlation coefficients of the gray Lena encrypted images with other literature. As can be seen from Table 2, the correlation coefficients of the ciphertext images in this paper are closer to zero.



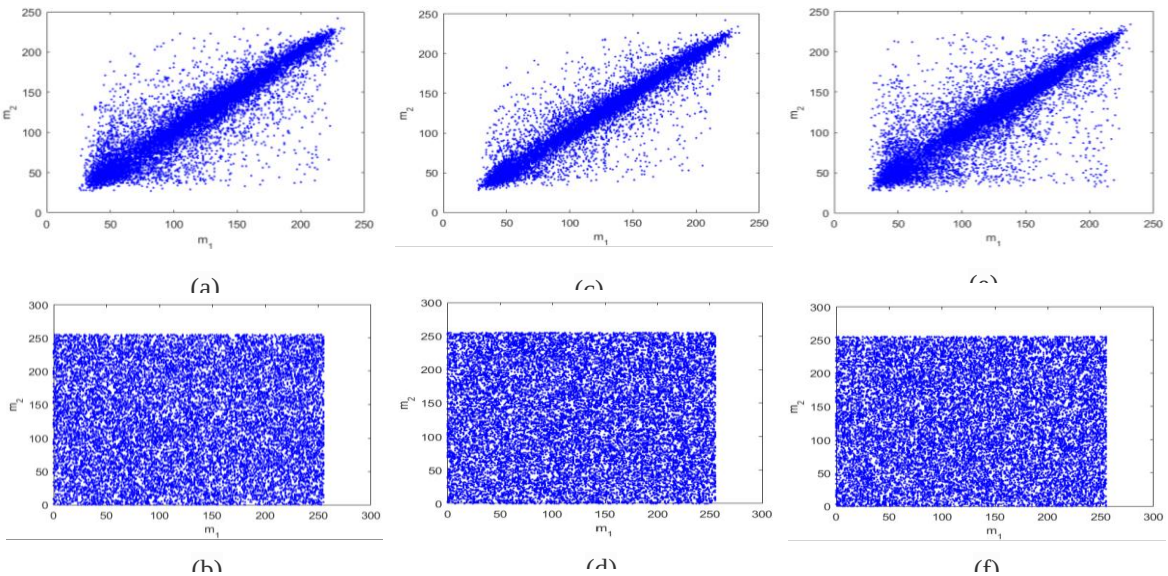
Correlation analysis diagrams: Horizontal direction of (a) plain Lena, (b) cipher Lena;Vertical direction of (c) plain Lena, (d) cipher Lena, Diagonal direction of (e) plain Lena, (f) cipher Lena

Figure 8: Correlation distribution of gray Lena images with indices m1, m2, and m3 of 2, 2, and 1, respectively



Correlation analysis diagrams: Horizontal direction of (a) plain Lena, (b) cipher Lena;Vertical direction of (c) plain Lena, (d) cipher Lena, Diagonal direction of (e) plain Lena, (f) cipher

Figure 9: Correlation distribution of gray Lena images with indices m1, m2, and m3



Correlation analysis diagrams: Horizontal direction of (a) plain Lena, (b) cipher Lena;Vertical direction of (c) plain Lena, (d) cipher Lena, Diagonal direction of (e) plain Lena, (f) cipher

Figure 10 Correlation distribution of gray Lena images with indices m1, m2, and m3 of 5, 4, and 1, respectively

Table 1: Distribution of correlation coefficients of different indices m1,m2, and m3 in different gray images

Different indices	*Image*	*direction*	**original images**	**Cipher Image**
The indices m1, m2, and m3 are 2, 2, 1 respectively	Lena	Level	0.9126	-0.000193
		Vertical	0.9573	0.000691
		Diagonal	0.8967	0.000356
	Peppers	Level	0.9526	-0.001
		Vertical	0.9609	0.0013
		Diagonal	0.9258	-0.000147
The indices m1, m2, and m3 are 3, 3, 1 respectively	Lena	Level	0.925	-0.000485
		Vertical	0.9588	-0.000674
		Diagonal	0.8940	0.000972
	Peppers	Level	0.9573	0.000952
		Vertical	0.9585	0.0012
		Diagonal	0.9268	-0.0016
The indices m1, m2, and m3 are 5, 4, 1 respectively	Lena	Level	0.9280	0.000651
		Vertical	0.9578	0.0021
		Diagonal	0.8994	-0.0012
	Peppers	Level	0.9568	-0.0015
		Vertical	0.9606	0.0013
		Diagonal	0.9257	0.0014

Table 2: Comparison of correlation coefficients of different indices m1,m2,m3 in gray Lena ciphertext images and other literature

Different indices	**Correlation direction**		
	Horizontal	**Vertical**	**Diagonal**
indices m1=2, m2=2, m3=1	-0.000193	0.000691	0.000356
indices m1=3 , m2=3 , m3=1	-0.000485	-0.000674	0.000972
indices m1=5 , m2=4 , m3=1	0.000651	0.0021	-0.0012
Relevance distribution of references	**Correlation direction**		
	Horizontal	**Vertical**	**Diagonal**
References [1]	0.0010	0.0013	0.0010
References [2]	-0.0057	0.0034	-0.0073
References [13]	0.0081	0.0316	0.0234
References [10]	-0.0038	-0.0026	0.0017

6.3 Differential Attack Analysis

Table 1: Distribution of correlation coefficients of different indices m1,m2, and m3 in different gray images

Different indices	*Image*	*direction*	**original images**	**Cipher Image**
The indices m1, m2, and m3 are 2, 2, 1 respectively	Lena	Level	0.9126	-0.000193
		Vertical	0.9573	0.000691
		Diagonal	0.8967	0.000356
	Peppers	Level	0.9526	-0.001
		Vertical	0.9609	0.0013
		Diagonal	0.9258	-0.000147
The indices m1, m2, and m3 are 3, 3, 1 respectively	Lena	Level	0.925	-0.000485
		Vertical	0.9588	-0.000674
		Diagonal	0.8940	0.000972
	Peppers	Level	0.9573	0.000952
		Vertical	0.9585	0.0012
		Diagonal	0.9268	-0.0016
The indices m1, m2, and m3 are 5, 4, 1 respectively	Lena	Level	0.9280	0.000651
		Vertical	0.9578	0.0021
		Diagonal	0.8994	-0.0012
	Peppers	Level	0.9568	-0.0015
		Vertical	0.9606	0.0013
		Diagonal	0.9257	0.0014

Table 2: Comparison of correlation coefficients of different indices m1,m2,m3 in gray Lena ciphertext images and other literature

Different indices	**Correlation direction**		
	Horizontal	**Vertical**	**Diagonal**
indices m1=2, m2=2, m3=1	-0.000193	0.000691	0.000356
indices m1=3 , m2=3 , m3=1	-0.000485	-0.000674	0.000972
indices m1=5 , m2=4 , m3=1	0.000651	0.0021	-0.0012
Relevance distribution of references	**Correlation direction**		
	Horizontal	**Vertical**	**Diagonal**
References [1]	0.0010	0.0013	0.0010
References [2]	-0.0057	0.0034	-0.0073
References [13]	0.0081	0.0316	0.0234
References [10]	-0.0038	-0.0026	0.0017

6.3 Differential Attack Analysis

Table 3: Distribution of correlation coefficients of different indices m1,m2, and m3 in different gray images

Different indices	*Image*	*direction*	**original images**	**Cipher Image**
The indices m1, m2, and m3 are 2, 2, 1 respectively	Lena	Level	0.9126	-0.000193
		Vertical	0.9573	0.000691
		Diagonal	0.8967	0.000356
	Peppers	Level	0.9526	-0.001
		Vertical	0.9609	0.0013
		Diagonal	0.9258	-0.000147
The indices m1, m2, and m3 are 3, 3, 1 respectively	Lena	Level	0.925	-0.000485
		Vertical	0.9588	-0.000674
		Diagonal	0.8940	0.000972
	Peppers	Level	0.9573	0.000952
		Vertical	0.9585	0.0012
		Diagonal	0.9268	-0.0016
The indices m1, m2, and m3 are 5, 4, 1 respectively	Lena	Level	0.9280	0.000651
		Vertical	0.9578	0.0021
		Diagonal	0.8994	-0.0012
	Peppers	Level	0.9568	-0.0015
		Vertical	0.9606	0.0013
		Diagonal	0.9257	0.0014

Table 4: Comparison of correlation coefficients of different indices m1,m2,m3 in gray Lena ciphertext images and other literature

Different indices	**Correlation direction**		
	Horizontal	**Vertical**	**Diagonal**
indices m1=2, m2=2, m3=1	-0.000193	0.000691	0.000356
indices m1=3 , m2=3 , m3=1	-0.000485	-0.000674	0.000972
indices m1=5 , m2=4 , m3=1	0.000651	0.0021	-0.0012
Relevance distribution of references	**Correlation direction**		
	Horizontal	**Vertical**	**Diagonal**
References [1]	0.0010	0.0013	0.0010
References [2]	-0.0057	0.0034	-0.0073
References [13]	0.0081	0.0316	0.0234
References [10]	-0.0038	-0.0026	0.0017

6.3 Differential Attack Analysis

An image encryption algorithm with good encryption results should be able to resist differential attacks very effectively. Generally, Numberof Pixels Change Rate (NPCR) and unified averaged changed intensity (UACI) indexes are used to evaluate differential attacks, and the ideal values of Numberof Pixels Change Rate (NPCR) and unified averaged changed intensity (UACI) are 99.6094%, and 33.4635%, respectively. Table 3 shows the experimental results of different indices m1, m2, and m3 in gray images; Table 4 compares different indices m1, m2, and m3 in gray Lena images and other algorithms. Table 4 shows that the encryption algorithm proposed in this paper is closer to the ideal value than some other algorithms in the literature.

Table 3: NPCR and UACI indexes of different indices k1,k2,k3 in gray images

Different indices	**Image**	**NPCR(%)**	**UACI(%)**
The indices m1, m2, and m3 are 2, 2, 1 respectively	**Lena**	99.6109	33.4757
	Peppers	99.5987	33.4706
	Baboon	99.6109	33.4729
The indices m1, m2, and m3 are 3, 3, 1 respectively	**Lena**	99.6140	33.4658
	Peppers	99.6002	33.4713
	Baboon	99.6017	33.4773
The indices m1, m2, and m3 are 5, 4, 1 respectively	**Lena**	99.6033	33.4780
	Peppers	99.6109	33.4557
	Baboon	99.5834	33.4956

Table 4: Comparison of NPCR and UACI of different indices m1,m2,m3 in gray Lena cipher images with other literature

Different indices	**NPCR(%)**	**UACI(%)**
The indices m1, m2, and m3 are 2, 2, 1 respectively	99.6109	33.4757
The indices m1, m2, and m3 are 3, 3, 1 respectively	99.6140	33.4658
The indices m1, m2, and m3 are 5, 4, 1 respectively	99.6033	33.4780
NPCR and UACI of references		
References [2]	99.6185	33.6245
References [16]	99.59	33.48
References [3]	99.6552	33.5871

6.4 Speed and Performance

In this paper, images of 256×256 and 512×512 sizes are used to test the usage time of the algorithm in the encryption phase and decryption phase, respectively. The device used is God of War Z7-TA7NP, the system is Windows 11, the running memory is 16G, and the software is Matlab 2016a. Table 5 shows the encryption and decryption times for different indices *m1*, *m2*, and *m3* at 256×256 image size; Table 6 shows the encryption and decryption times for different indices *m1*, *m2*, and *m3* at 512×512 image size. Table 7 shows the comparison between this algorithm and other literature in terms of encryption and decryption time. From the table, we can see that the algorithm has a faster encryption speed.

Table 5: Encryption and decryption times for different indices m1,m2, and m3 at 256×256 image size

Different indices	Image	Times(S)	
The indices m1,m2,m3 are 2,2,1 respectively	Lena(256*256)	encryption time(s)	0.079
		Decryption time(s)	0.078
	Peppers(256*256)	encryption time(s)	0.079
		Decryption time(s)	0.079
The indices m1,m2,m3 are 3,3,1 respectively	Lena(256*256)	encryption time(s)	0.11
		Decryption time(s)	0.125
	Peppers(256*256)	encryption time(s)	0.126
		Decryption time(s)	0.125
The indices m1,m2,m3 are 5,4,1 respectively	Lena(256*256)	encryption time(s)	0.375
		Decryption time(s)	0.39
	Peppers(256*256)	encryption time(s)	0.375
		Decryption time(s)	0.375

Table 6: Encryption and decryption times for different indices m1,m2, and m3 at 512×512 image size

Different indices	Image	Times(S)	
The indices m1,m2,m3 are 2,2,1 respectively	Lena(512*512)	encryption time(s)	0.313
		Decryption time(s)	0.299
	Peppers(512*512)	encryption time(s)	0.298
		Decryption time(s)	0.298
The indices m1,m2,m3 are 3,3,1 respectively	Lena(512*512)	encryption time(s)	0.456
		Decryption time(s)	0.459
	Peppers(512*512)	encryption time(s)	0.455
		Decryption time(s)	0.456
The indices m1,m2,m3 are 5,4,1 respectively	Lena(512*512)	encryption time(s)	1.441
		Decryption time(s)	1.453
	Peppers(512*512)	encryption time(s)	1.421
		Decryption time(s)	1.438

Table 7: Comparison of different indices m1,m2,m3 and other literature in terms of encryption and decryption time

Different indices	**Lena image of 256×256 size**		**Lena image of 512×512 size**	
	encryption time(s)	**Decryption time(s)**	**encryption time(s)**	**Decryption time(s)**
The indices m1,m2,m3 are 2,2,1 respectively	0.079	0.078	0.313	0.299
The indices m1,m2,m3 are 3,3,1 respectively	0.11	0.125	0.456	0.459
The indices m1,m2,m3 are 5,4,1 respectively	0.375	0.39	1.441	1.453
references	**Lena image of 256×256 size**		**Lena image of 512×512 size**	
	encryption time(s)	**Decryption time(s)**	**encryption time(s)**	**Decryption time(s)**
References [3]	0.5274	--	2.1027	--
References [8]	0.548691	--	1.634973	--
References [21]	0.921	--	--	--

6.5 Information entropy analysis

Information entropy is generally used to measure the randomness of an image[16], and its calculation formula is shown in equation (7):

$$H(x) = \sum_{i=0}^{2^N-1} p(x_i) log_2 \frac{1}{p(x_i)} \tag{7}$$

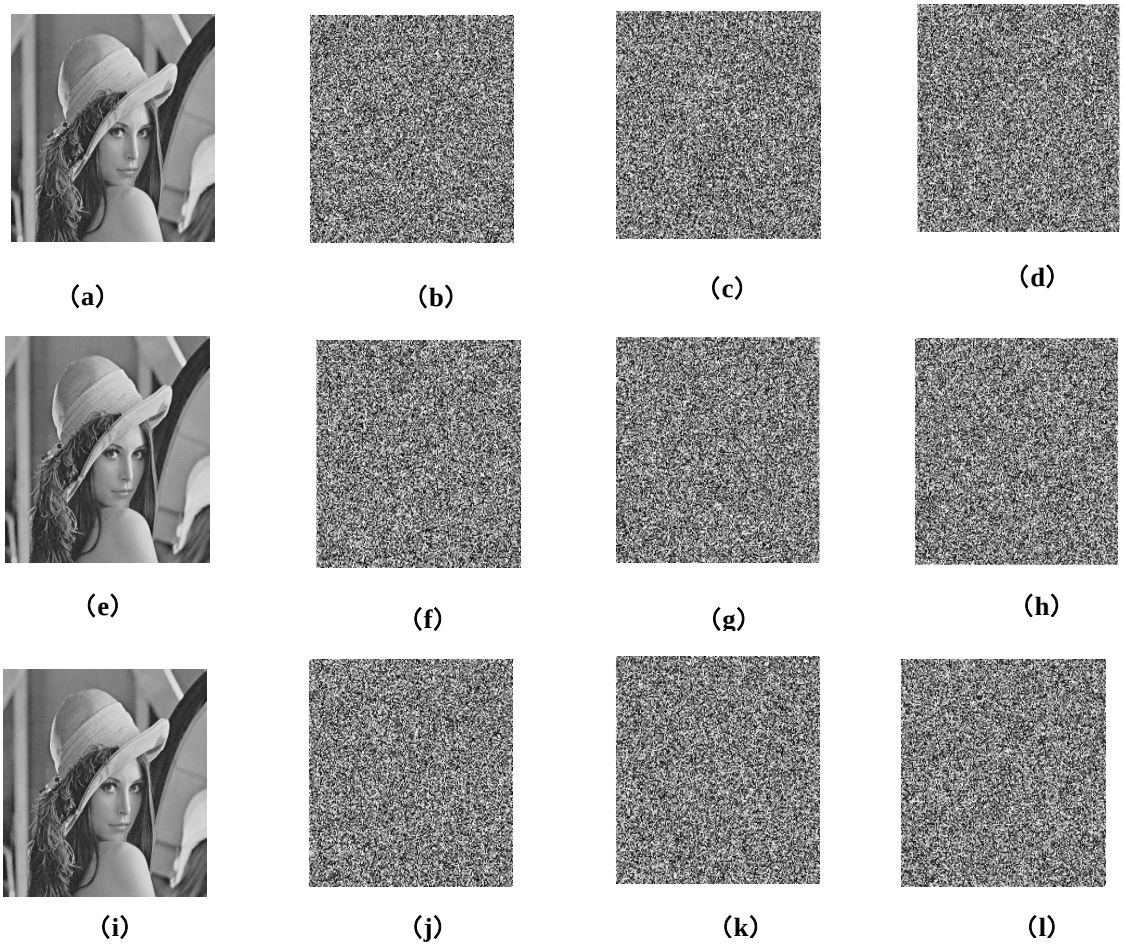
Where x_i is the pixel value, $p(x_i)$ is the probability of occurrence of x_i in the gray level, and *N* is the total value of pixel values. In the experiment, several 256×256 gray images are selected, and the information entropy values of the original and encrypted images are calculated using Eq. Table 8 shows the information entropy values for different indices *m1*, *m2*, and *m3* in the gray images. The table shows that the entropy values of the algorithm in the ciphertext images are all close to 8, which indicates that the algorithm proposed in this paper can effectively resist the information entropy attack.

Table 8: Information entropy index of gray images with different indices m1, m2, m3

Different indices	Image	Plain image	Cipher image
The indices m1,m2,m3 are 2,2,1 respectively	**Lena**	7.444015	7.98929
	Peppers	7.569734	7.989622
	Baboon	7.364862	7.989804
The indices m1,m2,m3 are 3,3,1 respectively	**Lena**	7.444015	7.989501
	Peppers	7.569734	7.990165
	Baboon	7.364862	7.989849
The indices m1,m2,m3 are 5,4,1 respectively	**Lena**	7.444015	7.989470
	Peppers	7.569734	7.989257
	Baboon	7.364862	7.989441

6.6 Key sensitivity analysis

This paper's key includes the initial values x=0.33, y=0.678, z=0.976, the system parameters $a11$, $a12$, $a13 \in [0, 1]$, etc; The constants $g11 \in [0, 1]$, etc; Using the gray image made by Lena as the original image, minor changes are made to the variable x and the system parameters $a11$ and $g11$, respectively. In contrast, the other independent variables and parameters remain unchanged, and finally, they are decrypted separately. Figure 11 shows the experimental results for different indices $m1$, $m2$, and $m3$. The first column of Fig. 11 shows the results of decrypting the image with the correct key, the second column tests the results of decrypting the image with a small change in variable x, and the third and fourth columns test the results of decrypting the image with a small change in the system parameters $a11$ and $g11$, respectively. From the experimental results, it can be concluded that the algorithm is highly sensitive [15].



(a), (b), (c), (d) are the decryption results for the indices m1=2,m2=2,m3=1 using the correct, x2=x+10^{-4}, a111=a11++10^{-10}, and g111=g11+10^{-6}, respectively

(e), (f), (g), (h) are the decryption results of using the correct exponents m1=3,m2=3,m3=1, x2=x+10^{-4}, a111=a11+10^{-12}, g111=g11+10^{-6}, respectively

(i), (j), (k), and (l) are the decryption results of using correct indices m1=5,m2=4,m3=1, x2=x+10^{-4}, a111=a11+10^{-16}, and g111=g11+10^{-6}, respectively

Figure 11: Sensitivity test of different indices m1, m2, m3 in gray Lena images during decryption

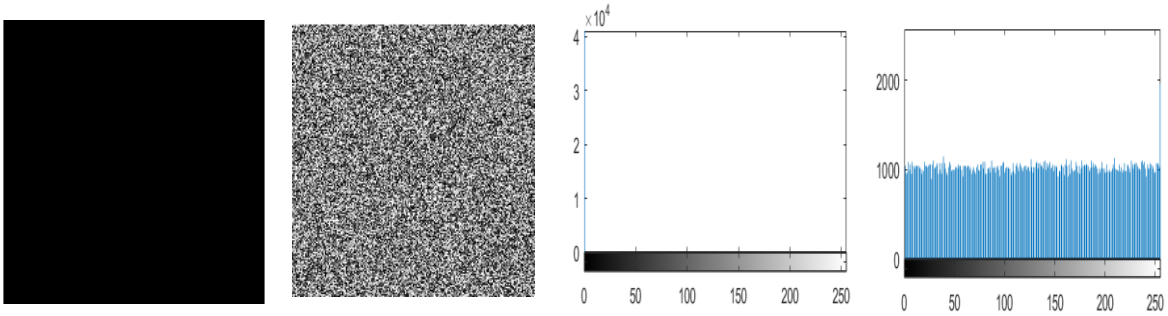
6.7 Key Space Analysis

The size of the key space usually depends on the length of the key, which responds to the cryptographic strength of a system. For this paper, it is assumed that m1, m2, and m3 of the system shown in Eq. (2) are all 3, n is 3, $a_{kij}, b_{kij}, c_{kij}, d_{kij} e_{kij}, f_{kij}, i = 1,2,3, j = 1,2\cdots,8,9$ are a random parameter belonging to [0, 1], respectively; and $g_{ni}, i = 1,2,3$ is a random constant belonging to [0, 1], respectively. There are

$3 \times 3 \times 9 = 81$ parameters in total, whose value interval is [0, 1], and the key space is $(10^{12})^{81} = 10^{972}$ according to the precision of 10^{-12}, which is a rather sizeable necessary space (the initial value, the parameters of the Logistic mapping, are not taken into account).

6.8 randomness analysis

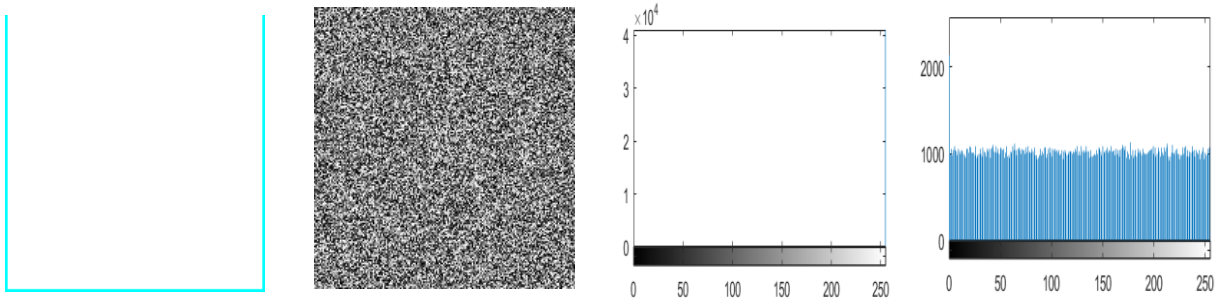
To measure the system's randomness, for this paper, it is assumed that m1, m2, and m3 of the system shown in expression (2) are 2, 2, and 1, respectively. To test the encryption on the all-black and all-white images. Figure 12 shows the case of an all-black appearance. Where, Fig. 12 (a) and (b) show the original and encrypted images of the all-black image, respectively; Fig. 12 (c) and (d) show the histograms of the authentic and encrypted images of the all-black image, respectively. Fig. 13 shows the case of an all-white appearance. In this case, Fig. 13 (a) and (b) show the original and encrypted images of the all-white image; Fig. 13 (c) and (d) show the histograms of the authentic and encrypted images of the all-white image, respectively. From Figs. 12 and 13, it can be seen that the algorithm's key is highly randomized.



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

(a) and (b) show the original and encrypted images of the all-black image, respectively ; (c) and (d) show the histograms of the original and encrypted images of the all-black image, respectively.

Figure 12 : Randomness testing of all-black



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

(a) and (b) respectively show the original and encrypted images of the all-white image ; (c) and (d) show the histogram of the original and encrypted images of the all-white image, respectively.

Figure 13 : Randomness test of all-white images

iteration and the grayscale image is subjected to a dissimilarity operation. From the bifurcation and phase diagrams, we can see that the system has chaotic solid characteristics. From the histogram, correlation, NPCR, and UACI, the algorithm has high security. Also, the low application cost, fast encryption speed, and more straightforward structure of the system is an advantage of the algorithm.

7. References

[1].Huang, H., Cheng, D. A secure image compression-encryption algorithm using DCT and hyperchaotic system. Multimed Tools Appl 81, 31329–31347 (2022). https://doi.org/10.1007/s11042-021-11796-x

[2].De Dieu, N., Ruben, F.S.V., Nestor, T. et al. Dynamic analysis of a novel chaotic system with no linear terms and use for DNA-based image encryption. Multimed Tools Appl 81, 10907–10934 (2022). https://doi.org/10.1007/s11042-022-12044-6

[3].Aouissaoui, I, Bakir, T, Sakly, A. Robustly correlated key-medical image for DNA-chaos based encryption. IET Image Process. 2021; 15: 2770– 2786. https://doi.org/10.1049/ipr2.12261

[4].Azimi, Z., Ahadpour, S. Color image encryption based on DNA encoding and pair coupled chaotic maps. Multimed Tools Appl 79, 1727–1744 (2020). https://doi.org/10.1007/s11042-019-08375-6

[5].Zefreh, E.Z. An image encryption scheme based on a hybrid model of DNA computing, chaotic systems and hash functions. Multimed Tools Appl 79, 24993–25022 (2020). https://doi.org/10.1007/s11042-020-09111-1

[6].Zhang, Q., Han, J. A novel color image encryption algorithm based on image hashing, 6D hyperchaotic and DNA coding. Multimed Tools Appl 80, 13841–13864 (2021). https://doi.org/10.1007/s11042-020-10437-z

[7].Nezhad, Shadi Yoosefian Dezfuli, Naser Safdarian, and Seyed Ali Hoseini Zadeh. "New method for fingerprint images encryption using DNA sequence and chaotic tent map." Optik 224https://doi.org/10.1016/j.ijleo.2020.165661

[8].Yousif, S.F., Abboud, A.J. & Alhumaima, R.S. A new image encryption based on bit replacing, chaos and DNA coding techniques. Multimed Tools Appl 81, 27453–27493 (2022). https://doi.org/10.1007/s11042-022-12762-x

[9].Ban Dohan, Lv Xin, and Wang Xinyuan. "An efficient image encryption algorithm based on one-dimensional chaotic mapping." Computer Science 47.4 (2020): 278-284.

[10].Chanil Pak, Lilian Huang,A new color image encryption using combination of the 1D chaotic map,Signal Processing,Volume 138,2017,Pages 129-137,ISSN 0165-1684.https://doi.org/10.1016/j.sigpro.2017.03.011

[11].Yuan, Y. W., and Jiu-Lun Fan. "An image encryption method based on two-dimensional segmented linear mapping." Microelectronics and Computers 6 (2010): 181-184.DOI : 10.19304 / j.cnki.issn1000-7180.2010.06.048.

[12].Li Jing, Xiang Fei, Zhang Junpeng. A digital image information encryption scheme based on chaos [J].Electronic Design Engineering, 2019,27 (12) : 84-88.DOI : 10.14022 / j.cnki.dzsjgc.2019.12.017.

[13].Liu Song, Zhang Jianqiang, Qiu Da, Liu Jingyi, Zhang Guoping. A third-order piecewise linear chaotic system and its application in image encryption [J / OL].Journal of Central China Normal University (Natural Science Edition) : 1-14 [2022-12-11].http://kns.cnki.net/kcms/detail/42.1178.N.20220114.1214.002.html

[14].Yu Wanbo, Wang Yuxin. Construction of linear piecewise chaotic map and its application in image encryption [J].Computer Engineering and Design, 2023,44 (03) : 707-713.

[15].Yu Wanbo, Huang Rongrong, Wang Wenjin. Three-dimensional composite chaotic system and its application in image encryption [J / OL].Computer engineering and application : 1-11 [2023-04-14].http://kns.cnki.net/kcms/detail/11.2127.tp.20230410.1936.004.html

[16].WanBo Yu, Zhenzhen Hu. Image encryption based on enhanced product trigonometric chaotic sequences[J]. Modern Physics Letters, B. 2022(13):36.

[17]Yu W, Wang H. Analysis of trigonometric chaotic sequence by proposing an index-based bit level scrambling image encryption[J]. Modern Physics Letters B, 2021, 35(24): 2150406. https://doi.org/10.1142/S0217984921504066

[18] Fei Yu, Hui Shen, Zinan Zhang, Yuanyuan Huang, Shuo Cai, Sichun Du, A new multi-scroll Chua's circuit with composite hyperbolic tangent-cubic nonlinearity: Complex dynamics, Hardware implementation and Image encryption application, Integration, Volume 81, 2021, Pages 71-83.

[19] Fei Yu, Si Xu, Xiaoli Xiao, Wei Yao, Yuanyuan Huang, Shuo Cai, Bo Yin, Yi Li. Dynamics analysis, FPGA realization and image encryption application of a 5D memristive exponential hyperchaotic system, Integration, vol. 90, pp. 58-70, 2023

[20]GaoYa. Research on image encryption based on chaotic linear iterated function system [D], Dalian University, 2023.

[21] Wang, X., Chen, S. An image encryption algorithm based on pixel bit operation and nonlinear chaotic system. Vis Comput 39, 3123–3144 (2023). https://doi.org/10.1007/s00371-022-02517-y

.