Received 19 May 2022, accepted 13 June 2022, date of publication 22 June 2022, date of current version 30 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185229

Color Image Encryption Using 2D Sine-Cosine Coupling Map

ZEZONG ZHANG[1], JIANENG TANG[1,2], FENG ZHANG[3], HUI NI[3], JINYUAN CHEN[2], AND ZHONGMING HUANG[2]

[1]College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China
[2]College of Engineering, Huaqiao University, Quanzhou 362021, China
[3]Fujian MM Electronics Company Ltd., Quanzhou 362000, China

Corresponding author: Jianeng Tang (jn_tang@hqu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61573004, in part by the Foundation of Quanzhou under Grant 2018C106R, and in part by the Pilot Project of Fujian Province under Grant 2022H0017.

ABSTRACT In this paper, a new two-dimensional sine-cosine coupling chaos map(2D-SCCM) is proposed. The performance of the proposed two-dimensional chaotic system is analysed by using trajectory distribution maps, Lyapunov exponents, sample entropy and sequence sensitivity, etc. The results show that the 2D-SCCM has better randomness and ergodicity, as well as a wider hyperchaotic range than the existing partial two-dimensional chaos maps. To verify its application in practice, a 2D-SCCM-based color image encryption algorithm is proposed. First, the plain image is used to combine with hash function to generate the key. Then, we construct S-Boxes using the random sequence generated by 2D-SCCM combined with Arnold map. Finally, a color image encryption algorithm is proposed by using the constructed S-Boxes and the chaotic map combined with hash function. Experimental simulations and security tests show that the proposed encryption algorithm has high encryption efficiency and strong security, and can effectively protect images from various attacks.

INDEX TERMS Image encryption, chaos, color image encryption, S-Box, Arnold map.

I. INTRODUCTION

With the rapid development of modern network technology, the use of images for information transmission has become more and more frequent. The security of images has received extensive attention from researchers because of the widespread use of images in the Internet. In order to protect the information of images, many schemes have been proposed by researchers, including data hiding [1], watermarking [2], and image encryption [3]–[5]. Among all the schemes, image encryption is one of the most straightforward and effective schemes [4].

Due to the large redundancy and strong correlation between image pixels, image encryption is different from text encryption [5]. Traditional data encryption schemes mainly include Advanced Encryption Standard (AES) and Data Encryption Standard (DES). When they are applied to image encryption, it takes a lot of time resulting in low encryption efficiency [4]. Therefore, researchers have proposed many different image encryption techniques, which contain image filtering [3], DNA coding [5], frequency domain transform [6], elliptic curves [7], compressive sensing [8], S-Box [9], fractional Fourier transform [10], neural networks [11], chaotic systems [12]–[14], etc. However, among these technologies, because of the special nature of chaotic systems, cryptography based on chaotic systems has become one of the most popular cryptographic schemes.

There are many unique characteristics of chaotic systems, including high sensitivity to initial conditions and parameters, internal randomness and ergodicity, which make them suitable for image encryption. Since J. Fridrich first applied chaotic system to image encryption [14], the image encryption schemes based on chaotic systems have been developed rapidly, and various chaos-based image encryption schemes have been proposed. For example, Wu *et al.* proposed an image encryption algorithm based on 2D-HSM and DNA coding techniques, where they performed a diffusion operation on the image by DNA and scrambled the image by chaotic sequences [5]. Mansouri *et al.* proposed an image encryption algorithm based on a one-dimensional chaotic system, where they added chaotic sequences to the original image during the diffusion process to improve the security of

The associate editor coordinating the review of this manuscript and approving it for publication was Jun Wang.

the algorithm [15]. J. Arif *et al.* proposed an image encryption algorithm based on Logistic map and S-Box [9]. They used S-Box to perform pixel substitution in the pixel diffusion stage to enhance the security of the algorithm, but this algorithm can only be used for grayscale images. Similarly, there are many other image encryption schemes based on chaotic systems that have been proposed by researchers [16]–[19].

Although many image encryption algorithms based on chaotic systems have been proposed, some of them have been proven to be insecure [5]. Because the one-dimensional chaotic map has the characteristics of small key space and simple chaotic orbit, the attacker can easily deduce the initial conditions and chaotic orbit of the system [12]. While high-dimensional chaotic systems have relatively complex chaotic behavior and larger key space, the computational complexity and space complexity of high-dimensional chaos are much larger than those of one-dimensional chaotic systems. Thus, it is relatively difficult to implement in practical hardware, and it takes a long time. Therefore, it becomes particularly important to propose a chaotic system with complex chaotic behavior and low computational complexity [19].

In this paper, a new two-dimensional discrete chaotic map is proposed, called 2D-SCCM. On the basis of studying the classical one-dimensional discrete chaotic map, a new two-dimensional discrete chaotic map is constructed by coupling the output variables of multiple one-dimensional discrete systems. At the same time, because the cosine function is introduced in the system, the output of the entire chaotic system is in the range of $[-1,1]$. To verify the chaotic properties of the proposed system, first the system is analyzed by using trajectory distribution diagrams, Lyapunov exponents(LE), sample entropy, and sequence sensitivity. Then the chaotic properties are compared with those of other 2D chaotic maps. Finally, the randomness of the system-generated sequences were tested by using the National Institute of Standards and Technology (NIST) statistical test tool. The results show that the proposed two-dimensional chaotic map has more complex chaotic behavior and wider hyperchaotic interval. Based on the proposed chaotic map, we further propose a color image encryption algorithm. The algorithm can be divided into two major parts: scrambling and diffusion. In the image scrambling stage, the chaotic sequences are used to perform overall scrambling. At the same time, pixels are replaced one by one by introducing S-Boxes. In the diffusion stage, XOR and the modular diffusion are used for pixel diffusion, so that the three components of the color image interact with each other to achieve better encryption performance. The experimental simulation results and security analysis show that the proposed algorithm can encrypt color images of arbitrary size and has strong resistance to attacks.

This paper is organized as follows. Section II introduces the mathematical model of 2D-SCCM. Section III analyzes dynamic behavior of the 2D-SCCM. Section IV details the proposed color image encryption algorithm. Section V introduces the simulation and safety analysis. And section VI is a brief summary about the paper.

II. 2D SINE-COSINE COUPLING MAP

This section first reviews two traditional one-dimensional discrete chaotic maps, and then introduces the mathematical model of the two-dimensional sine-cosine coupling chaotic map proposed in this paper.

A. LOGISTIC MAP

Logistic map is a one-dimensional discrete chaotic map with the mathematical definition of Eq(1).

$$x_{i+1} = 4\mu x_i(1 - x_i) \tag{1}$$

where its control parameter $\mu \in (0, 1]$. The system is in a chaotic state when the parameter $\mu \in [0.89, 1]$.

B. SINE MAP

Sine map is also a one-dimensional discrete chaotic map with the mathematical definition of Eq(2), which is derived from the sine function.

$$x_{i+1} = \rho \sin(\pi x_i) \tag{2}$$

where its control parameter $\rho \in (0, 1]$. The system is in a chaotic state when the parameter $\rho \in [0.87, 1]$.

C. 2D-SCCM

The traditional one-dimensional discrete chaotic map has many disadvantages such as narrow chaotic interval and simple system structure. To solve these problems, a new two-dimensional sine-cosine coupling chaotic system is proposed with the mathematical definition of Eq(3).

$$\begin{cases} x_{i+1} = \sin(4\alpha\pi y_i(1 - x_i)) \\ y_{i+1} = \cos(8\beta\pi \sin(\pi + y_i) + x_{i+1}) \end{cases} \tag{3}$$

where α, β are both system control parameters with $\alpha \neq 0$, $\beta \neq 0$. It can be seen from the mathematical definition of Eq(3) that the 2D-SCCM is mainly derived from Logistic and Sine maps. Its output signals are finally modulated with the nonlinear cosine function, which further improves the randomness of the output signals. Meanwhile, because of the boundedness of the cosine function, the final output signals x and y of the system are in the range of $[-1,1]$.

III. BEHAVIOR ANALYSIS

In this section, the dynamical behavior of the proposed chaotic system is analyzed by chaos trajectory, Lyapunov exponent, sample entropy, and sequence sensitivity. Meanwhile, the randomness of the system output sequences are tested by using the NIST statistical test tool. In addition, we compare the relevant properties with three existing 2D discrete chaotic systems, including 2D Sine Logistic modulation map (2D-SLMM) [19], 2D Logistic-Sine-coupling map (2D-LSCM) [12], 2D hyperchaotic map (2D-HM) [17].

The 2D-SLMM has two parameters γ and η. When the control parameter $\eta = 3$, it has a better chaotic performance.

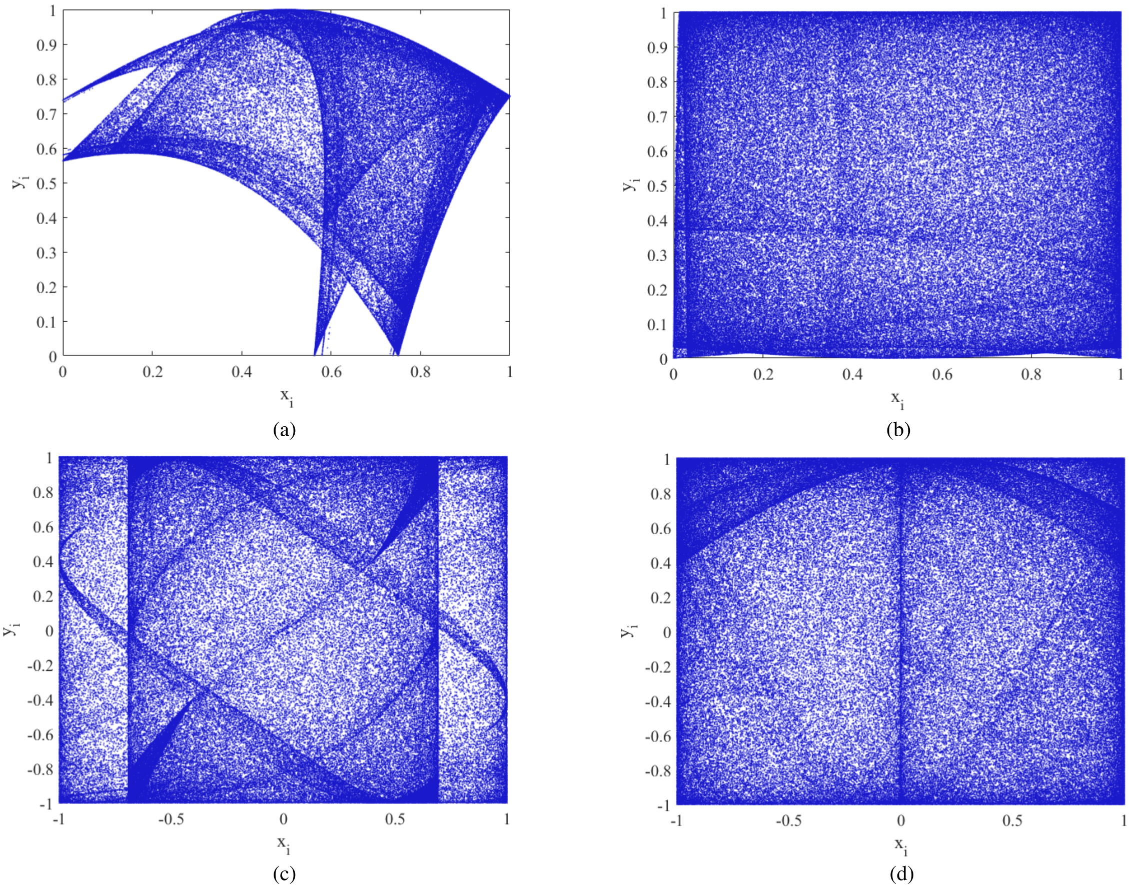


FIGURE 1. Trajectories of four 2D chaotic maps: (a) the 2D-SLMM with parameter $\gamma = 1$; (b) the 2D-LSCM with parameter $\theta = 0.99$; (c) the 2D-HM with parameter $\kappa = 2$ and $\lambda = 1$; (d) the 2D-SCCM with parameter $\alpha = 4$ and $\beta = 4$.

Thus, it is simply defined as follow [19]:

$$\begin{cases} x_{i+1} = \gamma(\sin(\pi y_i) + 3)x_i(1 - x_i) \\ y_{i+1} = \gamma(\sin(\pi x_{i+1}) + 3)y_i(1 - y_i) \end{cases} \tag{4}$$

The 2D-LSCM only has one parameter θ and the definition is written as follow [12]:

$$\begin{cases} x_{i+1} = \sin(\pi(4\theta x_i(1 - x_i) + (1 - \theta)\sin(\pi y_i))) \\ y_{i+1} = \sin(\pi(4\theta y_i(1 - y_i) + (1 - \theta)\sin(\pi x_{i+1}))) \end{cases} \tag{5}$$

The 2D-HM has two parameters κ and λ. its mathematical expression is defined as follow [17]:

$$\begin{cases} x_{i+1} = \sin(\dfrac{\kappa}{\sin(y_i)}) \\ y_{i+1} = \lambda\sin(\pi(x_i + y_i)) \end{cases} \tag{6}$$

A. CHAOS TRAJECTORY

The trajectory distribution diagram of a chaotic system describes the variation of the output of the system over time. Systems with complex chaotic behavior usually have complex trajectory distribution maps. And it also occupies a larger phase space [17].

Fig.1 shows the trajectory distributions of four different 2D chaotic systems, including (a) 2D-SLMM, (b) 2D-LSCM, (c) 2D-HM, (d) 2D-SCCM. When plotting these trajectory distributions, the parameters are set that can make all corresponding chaotic maps attain a completely chaotic state [12]. At the same time, the first 5000 terms of the system output are discarded in order to make the system reach a stable chaotic state. It can be seen from the Fig.1 that the trajectory distribution diagram of 2D-SLMM only occupies a small part of the phase plane, and the trajectory distribution diagram of 2D-LSCM does not completely occupy the entire phase plane, while the trajectory distribution diagrams of 2D-HM and 2D-SCCM completely occupies the entire phase plane. In addition, the trajectory distribution of 2D-SCCM is more uniform than that of 2D-HM, indicating that the sequences generated by 2D-SCCM are more random than the other three chaotic systems.

B. LYAPUNOV EXPONENT

The initial state sensitivity is one of the main features to determine whether a dynamic system exists chaotic behavior. A good chaotic system should gradually output two distinct sequences of iterations after inputting two initial iterations with small differences. We usually use the Lyapunov Exponent (LE) [12] to characterize the sensitivity of a chaotic system. Assuming that a one-dimensional discrete chaotic system is defined by $x_{i+1} = F(x_i)$, where $F(x)$ is piecewise differentiable [12], the LE of the system is defined as

$$LE = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln \left|F'(x_i)\right| \tag{7}$$

LE is a numerical feature that describes the average separation rate and divergence degree of adjacent phase space trajectories [12]. A positive LE means that the system exist chaotic behavior. And the larger the value of LE is, the more complex behavior the chaotic system has. For high-dimensional chaotic systems, there are generally several different LE values in the system. When the system has two or more positive LE values, we call this system a hyperchaotic system. Compared with the general chaotic system, the hyperchaotic system has more complex chaotic behavior.

A two-dimensional chaotic system usually has two LE values, and Fig.2 shows the LE of different two-dimensional chaotic systems under their respective control parameters. When the parameter range is set to [0,1], the LE values of each system behave as follows: the 2D-SLMM shows hyperchaotic behavior when $\gamma \in [0.85, 1]$, the 2D-LSCM shows hyperchaotic behavior when $\theta \in [0.57, 1]$, the 2D-HM shows hyperchaotic behavior when $\lambda \in [0.2, 0.34]$ or $\lambda \in [0.36, 0.58]$, and the 2D-SCCM shows hyperchaotic behavior when $\alpha \in [0.28, 1]$ or $\beta \in [0.15, 1]$. The result shows that the 2D-SCCM has a wider range of hyperchaotic intervals compared with the other three chaotic maps. In order to better compare the magnitude of LE, we compare the two LE values with the four chaotic systems under certain parameters in Fig.3. The results show that the LE values of 2D-SCCM are larger than the other three existing chaotic systems in a large interval, which further indicates that the proposed system have more complex chaotic behavior.

C. SAMPLE ENTROPY

Sample Entropy (SE) is a new evaluation method for testing the complexity of time series based on approximate entropy [22], which is used to describe the output similarity of dynamic systems. The SE of a given time series $\{z_1, z_2, \cdots, z_N\}$ of m dimension is defined as [23]

$$SE(m, r, N) = -\log \frac{K_1}{K_2} \tag{8}$$

where m is the reconstruction dimension and r is the calculated threshold, which we usually set $m = 2$, $r = 0.2 \times std$ (the std is the standard deviation of the tested time series). The K_1 represents the number of vectors whose Chebyshev distance is less than or equal to r between $\boldsymbol{Z_{m\mathcal{C}1}}(i)$ and $\boldsymbol{Z_{m\mathcal{C}1}}(j)$, and the K_2 represents the number of vectors whose Chebyshev distance is less than or equal to r between $\boldsymbol{Z_m}(i)$ and $\boldsymbol{Z_m}(j)$, where $\boldsymbol{Z_m}(i) = \{z_i, z_{i+1}, \cdots, z_{i+m-1}\}$. The larger the calculated SE value, the lower the level of regularity of the system and the more complex the chaotic behavior of the system [22]. Fig.4 shows the comparison of SE tests for Sine, Logistic and four 2D chaotic systems under different control parameter conditions. It can be seen that the SE values of the proposed 2D-SCCM system are larger than those of the other five chaotic systems, indicating that the proposed chaotic system has more complex chaotic characteristics.

D. SEQUENCE SENSITIVITY ANALYSIS

A good chaotic system should be extremely sensitive to changes in the initial values. In other words, if the system's inputs are slightly different, its output sequence should be significantly different. In order to test the sensitivity of the system, we input initial values with a difference of 10^{-15}, and the changes of the output sequences are shown in Fig.5. They can be seen that the system's outputs are significantly different after 14 iterations, indicating that our proposed system is extremely sensitive to the initial values.

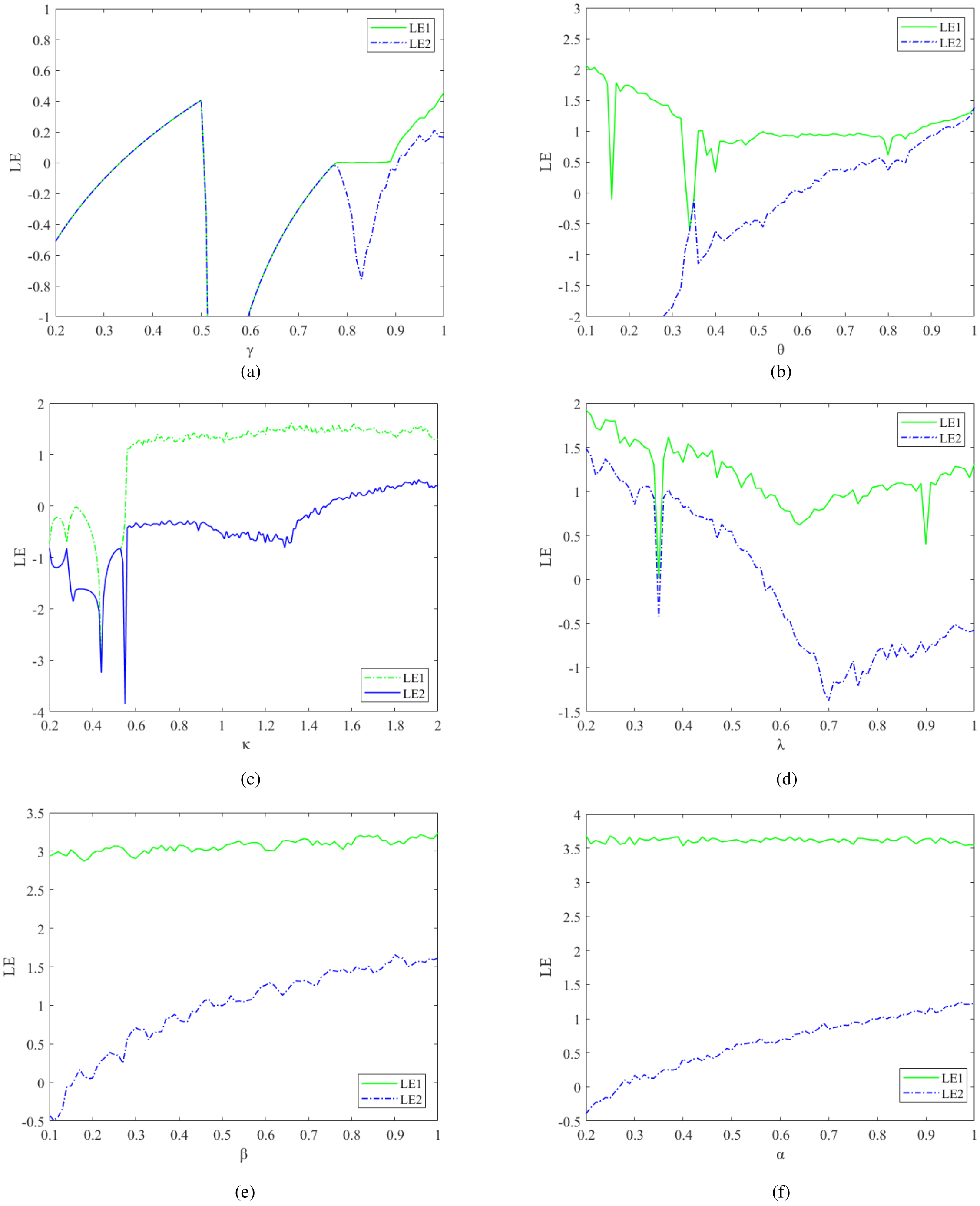
E. NIST

To further investigate the randomness of the system output sequences, we used the National Institute of Standards and Technology (NIST) SP800-22 test tool [24]. The NIST statistical test tool includes 15 statistical tests. It is used for evaluating the randomness of the sequences generated by 2D-SCCM [34]. After the NIST statistical test tool finishes testing, each test item will generate a corresponding test value $P \in [0, 1]$. And when the corresponding generated P-value is greater than 0.01, it indicates passing the test item. When all test items pass, it indicates that the sequence has strong randomness, while a larger P-value indicates stronger randomness [12]. The computational statistics of the NIST test are shown in Table 1. From the results, the sequences generated by the system can pass all tests of 15 items, indicating that they have strong pseudo-randomness.

TABLE 1. **P-value of NIST statistical test.**

Sub-Tests	P-value(x) $\geq$0.01	Result	P-value(y) $\geq$0.01	Result
Approximate Entropy(m = 10)	0.949602	Pass	0.299251	Pass
Block Frequency(M = 128)	0.534146	Pass	0.054199	Pass
Cumulative-Forward test	0.066882	Pass	0.911413	Pass
Cumulative-Reverse test	0.100508	Pass	0.804337	Pass
FFT	0.100508	Pass	0.407091	Pass
Frequency	0.862344	Pass	0.468595	Pass
Linear Complexity(M = 500)	0.213309	Pass	0.148094	Pass
Longest Run	0.862344	Pass	0.602458	Pass
Non-overlapping template(m = 9)*	0.458847	Pass	0.420940	Pass
Overlapping template(m = 9)	0.299251	Pass	0.100508	Pass
Random Excursions*	0.147839	Pass	0.142809	Pass
Random Excursions Variant*	0.183107	Pass	0.135168	Pass
Rank	0.299251	Pass	0.253551	Pass
Runs	0.911413	Pass	0.178278	Pass
Serial1 test(m = 16)	0.602458	Pass	0.976060	Pass
Serial2 test(m = 16)	0.804337	Pass	0.407091	Pass
Universal	0.534146	Pass	0.299251	Pass

*The average values of multiple tests.



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

FIGURE 2. The two LEs of four 2D chaotic maps: (a) the 2D-SLMM; (b) the 2D-LSCM; (c) the 2D-HM with fixed parameter $\lambda = 1$; (d) the 2D-HM with fixed parameter $\kappa = 2$; (e) the 2D-SCCM with fixed parameter $\alpha = 4$; (f) the 2D-SCCM with fixed parameter $\beta = 4$.

IV. ENCRYPTION ALGORITHM

This section introduces a color image encryption algorithm based on 2D-SCCM combined with S-Box scrambling, and the overall encryption structure is shown in Fig.6. The hash value is generated by the hash function and plain image, and the key is updated by the hash value to generate the finally key. First, the chaotic sequences generated by 2D-SCCM are used to diffuse the three color components of the original image to change the pixel values. Then, the sequences are used to generate S-Boxes combined with Arnold mapping to confuse the diffused image, and the confusing method can change the position and value of the image pixels at the same time. Finally, combined with the hash value and chaotic sequences, the scrambled image is diffused again to improve

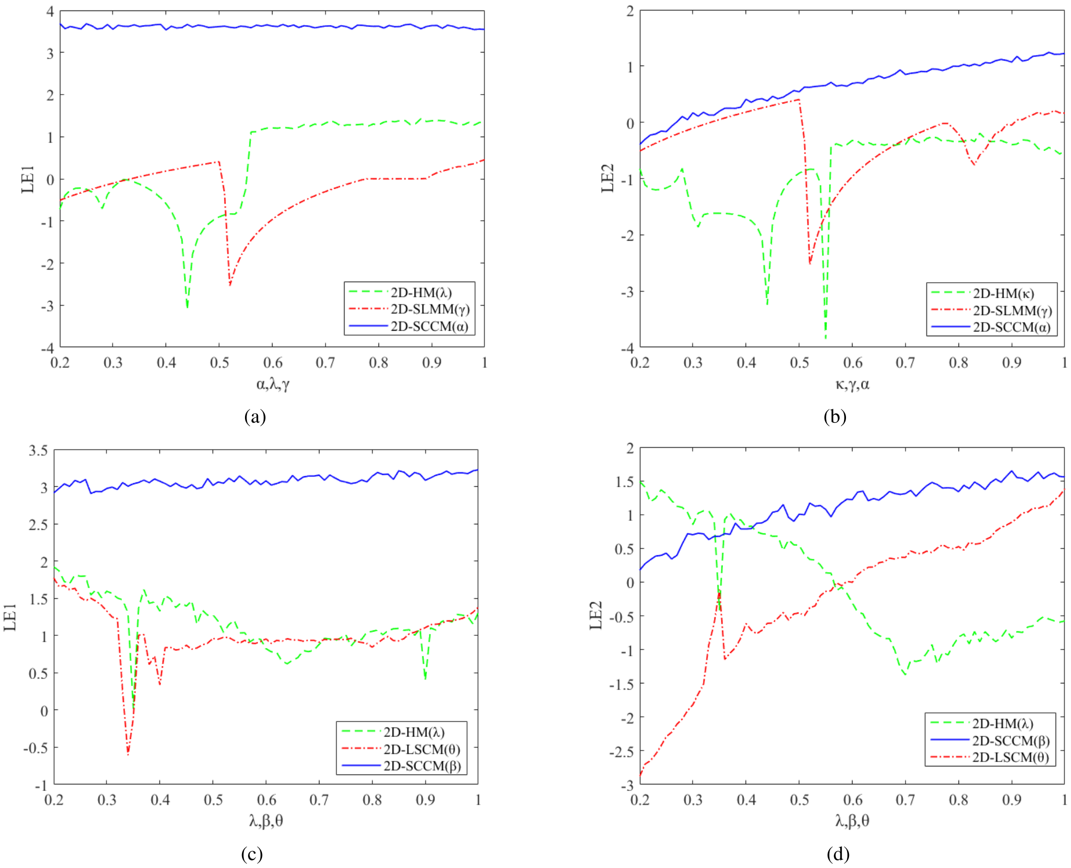


FIGURE 3. The comparison of LE: (a) the comparison of 2D-HM(λ), 2D-SLMM(γ) and 2D-SCCM(α) for LE1; (b) the comparison of 2D-HM(κ), 2D-SLMM(γ) and 2D-SCCM(α) for LE2; (c) the comparison of 2D-HM(λ), 2D-LSCM(θ) and 2D-SCCM(β) for LE1; (d) the comparison of 2D-HM(λ), 2D-LSCM(θ) and 2D-SCCM(β) for LE2.

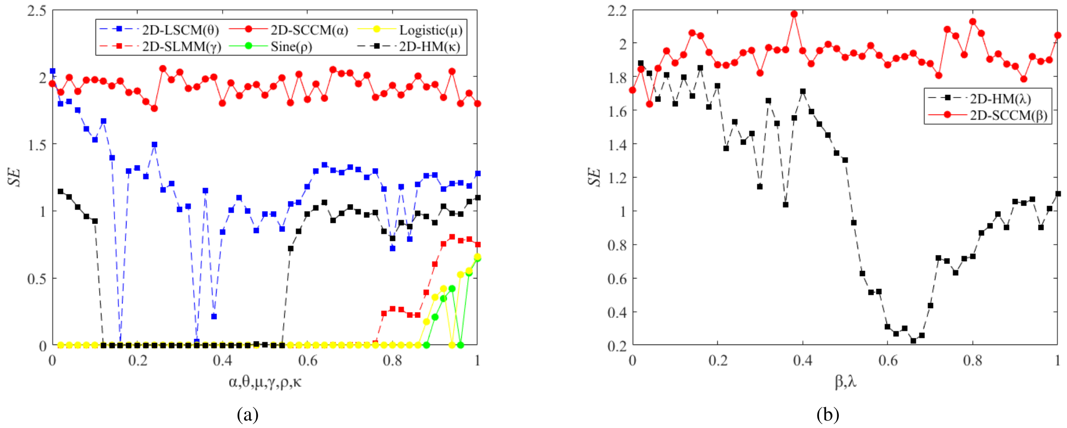
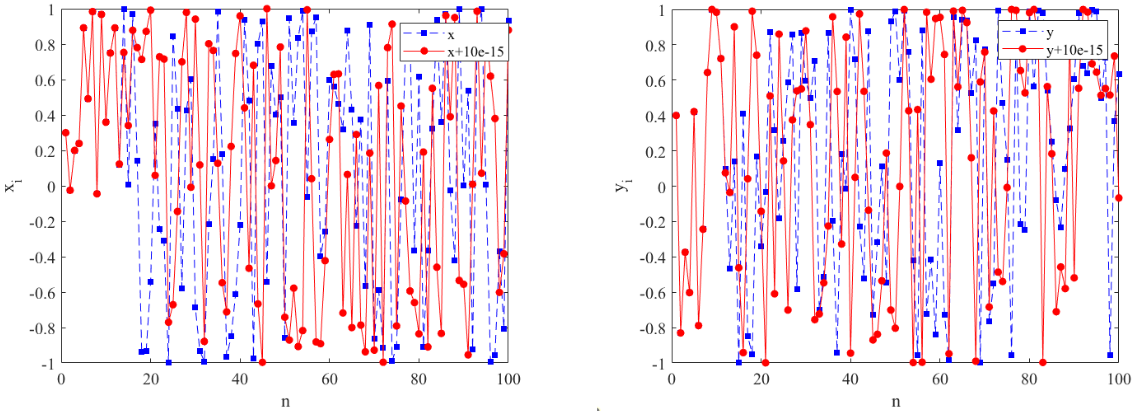


FIGURE 4. The SE comparison of different chaotic systems: (a) the Logistic, Sine, 2D-LSCM, 2D-SLMM($\eta = 3$), 2D-HM($\lambda = 1$) and 2D-SCCM(β=4); (b) 2D-HM($\kappa = 1$) and 2D-SCCM($\alpha = 4$).

the security performance of image encryption. The specific encryption steps are described as follows.

A. KEY GENERATION

A separate system key increases the possibility of the system being attacked to a chosen plaintext attack. The hash value obtained from the plain image is used to update the initial key, which reduces the possibility of the system being attacked by the chosen plaintext. Here we choose the SHA-256 function, the generated hash value is divide into 64 blocks, and converted each block to a decimal number, denoted as k_1, $k_2,\cdots,k_{64}$. Then, the initial key is set to $\mathbf{K} = \{x_0, y_0, a_0, b_0\}$,



(a)

(b)

FIGURE 5. Sequence sensitivity analysis: (a)x (b)y.

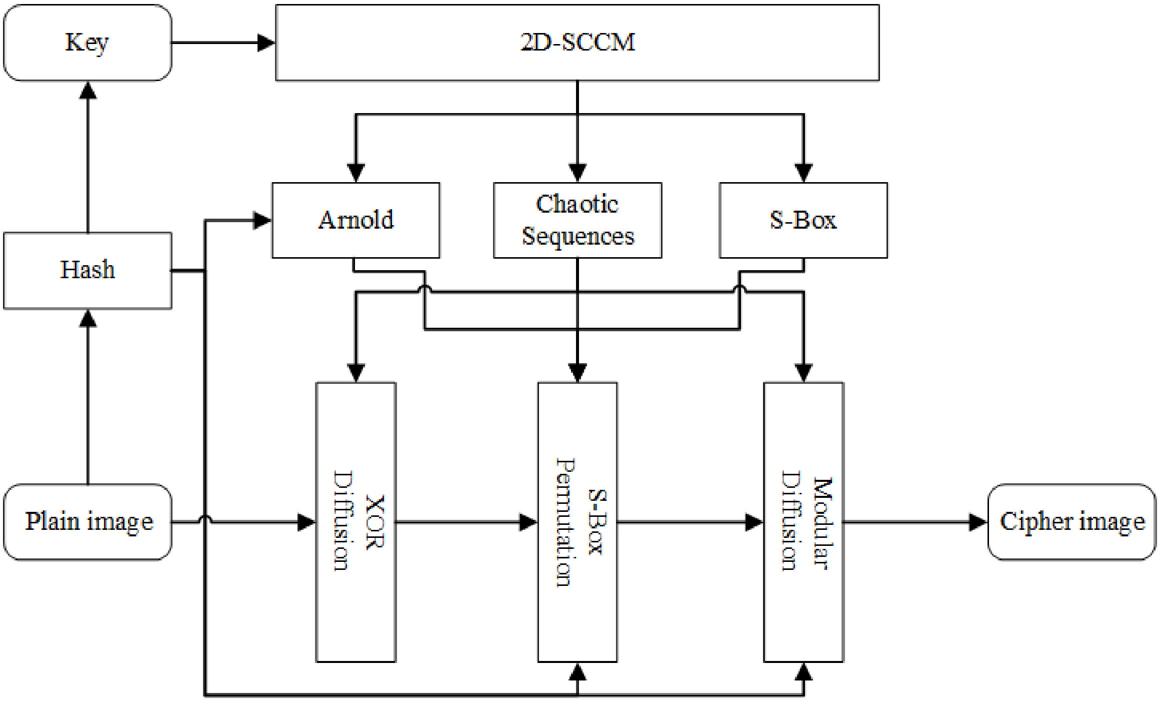


FIGURE 6. Block diagram of the overall encryption algorithm.

and the set initial key is updated by using Eq(9).

$$\begin{cases} x_1 = (\dfrac{(k_1 + k_2 + \cdots + k_{16})}{(k_{17} + k_{18} + \cdots + k_{32})}) \; mod \; 1 + x_0 \\ \alpha = ((k_1 + k_2 + \cdots + k_{32})/64) \; mod \; 1 + a_0 \\ \beta = ((k_{32} + k_{33} + \cdots + k_{64})/16) \; mod \; 1 + b_0 \\ iter = floor(mean(k_1, k_2, \cdots, k_{64})) + 40 \\ SQ = (k_1 + k_2 + \cdots + k_{64}) \times 2 \\ CZ = (k_1 + k_2 + \cdots + k_{64}) \; mod \; 256 \\ y_1 = (\dfrac{mean(k_{33}, k_{34}, \cdots, k_{48})}{mean(k_{49}, k_{50}, \cdots, k_{64})}) \; mod \; 1 + y_0 \end{cases} \tag{9}$$

where $\{x_1, y_1\}$ is the updated initial condition and $\{\alpha, \beta\}$ is the updated control parameters of the chaotic system. The iter is the number of iterations of the Arnold map. The SQ is the number of iteration terms discarded in the encryption process to bring the chaotic system to full chaos. The CZ is the preset value used in taking the modal diffusion. The floor(Q) means taking the largest integer not larger than Q. The mean represents taking the average, and the mod means the modulo operation.

B. XOR-DIFFUSION

Step 1: Input a color plain image **P** with a size of $M \times N$, and decompose it into three components of R, G and B with a size of $M \times N$, which are denoted as **Pr**, **Pg** and **Pb** respectively.

Step 2: Substitute $\{x_1, y_1, \alpha, \beta\}$ into 2D-SCCM, and iterate the system $n = SQ + M \times N$ times. Then the first SQ items are discard. Finally we can get the sequence **X1** and **Y1** according to Eq(10).

$$\begin{cases} \boldsymbol{X1} = \{x_{SQ+1}, x_{SQ+2}, \cdots, x_{SQ+MN}\} \\ \boldsymbol{Y1} = \{y_{SQ+1}, y_{SQ+2}, \cdots, y_{SQ+MN}\} \end{cases} \tag{10}$$

Step 3: Let $y_1' = x_{SQ+MN}$, $x_1' = y_{SQ+MN}$, then substitute the updated $\{x_1', y_1', \alpha, \beta\}$ into the chaotic system. Iterate the system $n = SQ + M \times N$ times with discarding the first SQ items. Finally we can get the sequence **X2** and **Y2** according to Eq(11).

$$\begin{cases} \boldsymbol{X2} = \{x'_{SQ+1}, x'_{SQ+2}, \cdots, x'_{SQ+MN}\} \\ \boldsymbol{Y2} = \{y'_{SQ+1}, y'_{SQ+2}, \cdots, y'_{SQ+MN}\} \end{cases} \tag{11}$$

Step 4: Convert the obtained sequences **X1** into matrices **A1** with a size of $M \times N$ according to Eq(12). Similarly, we can get **B1**, **A2**. At this time, the element values of **A1**, **B1**, and **A2** are all in the range of [0, 255], where the abs represents taking the absolute value.

$$\begin{cases} \boldsymbol{X11}(i) = (abs(floor(\boldsymbol{X1}(i) \times 10^{12}))) \text{ mod } 256 \\ \boldsymbol{A1} = reshape(\boldsymbol{X11}, [M, N]) \\ \boldsymbol{Y11}(i) = (abs(floor(\boldsymbol{Y1}(i) \times 10^{12}))) \text{ mod } 256 \\ \boldsymbol{B1} = reshape(\boldsymbol{Y11}, [M, N]) \\ \boldsymbol{X22}(i) = (abs(floor(\boldsymbol{X2}(i) \times 10^{12}))) \text{ mod } 256 \\ \boldsymbol{A2} = reshape(\boldsymbol{X22}, [M, N]) \\ i = 1, 2 \cdots MN \end{cases} \tag{12}$$

Step 5: The generated **A1**, **A2** and **B1** are used to perform bitwise XOR with the three image components of the color image. The image components **PR**, **PG** and **PB** are obtained respectively after the first diffusion encryption.

$$\begin{cases} \boldsymbol{PR}(i,j) = \boldsymbol{Pr}(i,j) \oplus \boldsymbol{A1}(i,j) \\ \boldsymbol{PG}(i,j) = \boldsymbol{Pg}(i,j) \oplus \boldsymbol{A2}(i,j) \\ \boldsymbol{PB}(i,j) = \boldsymbol{Pb}(i,j) \oplus \boldsymbol{B1}(i,j) \end{cases} \quad \begin{matrix} i = 1, 2, \cdots, M \\ j = 1, 2, \cdots, N \end{matrix} \tag{13}$$

C. CONFUSION

Generally, each pixel of natural image has strong correlation, and a good encryption algorithm is to eliminate the correlation between elements [15]. Since we only change the value of each pixel in the diffusion stage without modifying the position of the pixel, the correlation between the original pixels cannot be changed. In this section, we use a combination of chaotic sequences and S-Boxes to perform the scrambling operation on the diffused image, and the specific scrambling steps are as follows.

Step 1: Using the chaotic sequences **X1**, **Y1**, **X2**, **Y2** from the subsection B in section IV, generate **S-Box1** according to Eq(14). And we can obtain the initial parameters of the Arnold map according to Eq(15) and Eq(16), respectively.

$$\begin{cases} [-, \boldsymbol{S}] = sort(\boldsymbol{X1}(1:256) + \boldsymbol{Y1}(1:256) \\ \qquad + \boldsymbol{X2}(1:256) + \boldsymbol{Y2}(1:256)) \\ \boldsymbol{S\text{-}Box1} = reshape(\boldsymbol{S}, [16, 16]) - ones(16, 16) \end{cases} \tag{14}$$

$$\begin{cases} a1 = (floor(abs(\boldsymbol{Y2}(iter) \times 10^{12}))) \; mod \; 20 + 1 \\ b1 = (floor(abs(\boldsymbol{Y2}(iter + 1) \times 10^{12}))) \; mod \; 20 + 1 \end{cases} \tag{15}$$

$$\begin{cases} a2 = (floor(abs(\boldsymbol{Y2}(iter + 2) \times 10^{12}))) \; mod \; 20 + 1 \\ b2 = (floor(abs(\boldsymbol{Y2}(iter + 3) \times 10^{12}))) \; mod \; 20 + 1 \end{cases} \tag{16}$$

Step 2: Scramble **S-Box1** according to the initial parameters obtained in the Step1. Two new S-Boxes are generated, namely **S-Box2** and **S-Box3**.

$$\begin{cases} \boldsymbol{S\text{-}Box2} = Arnold(\boldsymbol{S\text{-}Box1}, iter, a1, b1) \\ \boldsymbol{S\text{-}Box3} = Arnold(\boldsymbol{S\text{-}Box2}, iter, a2, b2) \end{cases} \tag{17}$$

Step 3: Sort the chaotic sequences **X1**, **X2**, **Y1**, **Y2** to get the index sequences **SC1**, **SC2**, **SC3**, **SC4**.

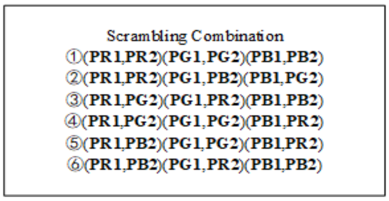
$$\begin{cases} [-, \boldsymbol{SC1}] = sort(\boldsymbol{X1}) \\ [-, \boldsymbol{SC2}] = sort(\boldsymbol{X2}) \\ [-, \boldsymbol{SC3}] = sort(\boldsymbol{Y1}) \\ [-, \boldsymbol{SC4}] = sort(\boldsymbol{Y2}) \end{cases} \tag{18}$$

Step 4: Perform bit plane decomposition on the encrypted image components obtained by the diffusion in the subsection B. Then recombine the upper four bits and the lower four bits respectively. Finally obtain the decomposed image components **PR1**, **PR2**, **PG1**, **PG2**, **PB1**, **PB2**.

$$\begin{cases} \boldsymbol{PR1} = \boldsymbol{PR} \; mod \; 16 + 1 \\ \boldsymbol{PR2} = floor(\boldsymbol{PR}/16) + 1 \\ \boldsymbol{PG1} = \boldsymbol{PG} \; mod \; 16 + 1 \\ \boldsymbol{PG2} = floor(\boldsymbol{PG}/16) + 1 \\ \boldsymbol{PB1} = \boldsymbol{PB} \; mod \; 16 + 1 \\ \boldsymbol{PB2} = floor(\boldsymbol{PB}/16) + 1 \end{cases} \tag{19}$$

Step 5: Use the chaotic sequence **Y2** to generate the parameter H for controlling the scrambling combination according to Eq(20), in which Fig.7 shows the six ways of the set scrambling combination. The final scrambling image components **WR**, **WG**, **WB** are obtained according to the selected scrambling combination using Eq(21). In this paper, Eq(21) represents scrambling combination ①, and other combinations are similarly represented.

$$H = (floor(abs(\boldsymbol{Y2}(iter) \times 10^{12}))) \; mod \; 6 + 1 \tag{20}$$



Scrambling Combination

①(PR1,PR2)(PG1,PG2)(PB1,PB2)

②(PR1,PR2)(PG1,PB2)(PB1,PG2)

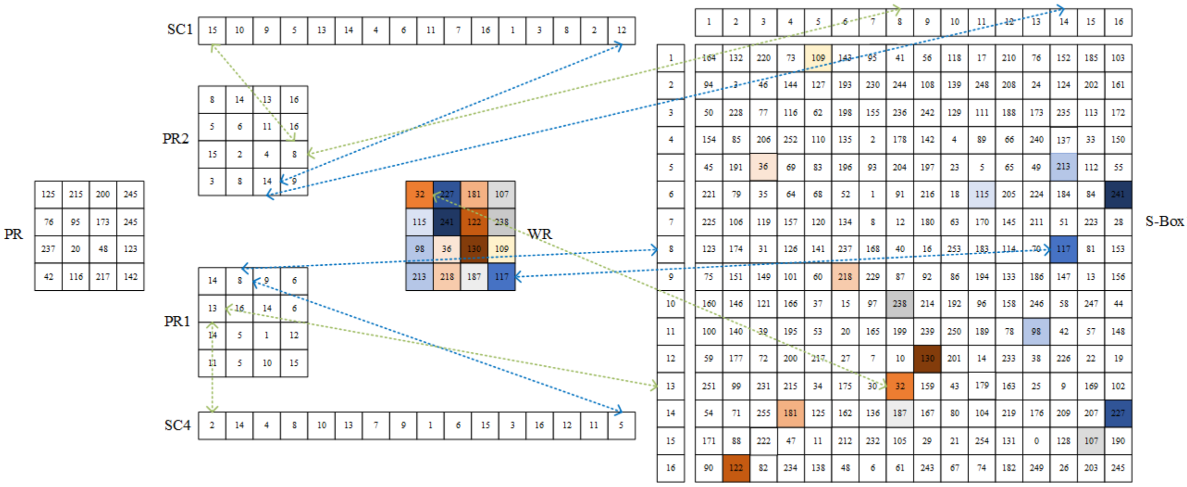
③(PR1,PG2)(PG1,PR2)(PB1,PB2)

④(PR1,PG2)(PG1,PG2)(PB1,PR2)

⑤(PR1,PB2)(PG1,PG2)(PB1,PR2)

⑥(PR1,PB2)(PG1,PR2)(PB1,PB2)

FIGURE 7. The scrambled combination.



SC1

15	10	9	5	13	14	4	6	11	7	16	1	3	8	2	12

PR2

8	14	13	16
5	6	11	16
15	2	4	8
3	8	14	9

PR

125	215	200	245
76	95	173	245
237	20	48	123
42	116	217	142

WR

32	227	181	107
115	241	122	238
98	36	130	109
213	218	187	117

PR1

14	8	9	6
13	16	14	6
14	5	1	12
11	5	10	15

SC4

2	14	4	8	10	13	7	9	1	6	15	3	16	12	11	5

S-Box1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	164	132	220	73	109	143	95	41	56	118	17	210	76	152	185	103
2	94	3	46	144	127	193	230	244	108	139	248	208	24	124	202	161
3	50	228	77	116	62	198	155	236	242	129	111	188	173	235	113	172
4	154	85	206	252	110	135	2	178	142	4	89	66	240	137	33	150
5	45	191	36	69	83	196	93	204	197	23	5	65	49	213	112	55
6	221	79	35	64	68	52	1	91	216	18	115	205	224	184	84	241
7	225	106	119	157	120	134	8	12	180	63	170	145	211	51	223	28
8	123	174	31	126	141	237	168	40	16	253	183	114	70	117	81	153
9	75	151	149	101	60	218	229	87	92	86	194	133	186	147	13	156
10	160	146	121	166	37	15	97	238	214	192	96	158	246	58	247	44
11	100	140	39	195	53	20	165	199	239	250	189	78	98	42	57	148
12	59	177	72	200	217	27	7	10	130	201	14	233	38	226	22	19
13	251	99	231	215	34	175	30	32	159	43	179	163	25	9	169	102
14	54	71	255	181	125	162	136	187	167	80	104	219	176	209	207	227
15	171	88	222	47	11	212	232	105	29	21	254	131	0	128	107	190
16	90	122	82	234	138	48	6	61	243	67	74	182	249	26	203	245

FIGURE 8. **A scrambling example of a 4 × 4 size image, with the PR component of the color image taking the scrambling of combination ①.**

$$\begin{cases} \boldsymbol{WR}(k) = \boldsymbol{S\text{-}Box1}(\boldsymbol{PR1}(\boldsymbol{SC4}(k)), \boldsymbol{PR2}(\boldsymbol{SC1}(k))) \\ \boldsymbol{WG}(k) = \boldsymbol{S\text{-}Box2}(\boldsymbol{PG1}(\boldsymbol{SC4}(k)), \boldsymbol{PG2}(\boldsymbol{SC2}(k))) \\ \boldsymbol{WB}(k) = \boldsymbol{S\text{-}Box3}(\boldsymbol{PB1}(\boldsymbol{SC4}(k)), \boldsymbol{PB2}(\boldsymbol{SC3}(k))) \\ k = 1, 2, \cdots, MN \end{cases} \tag{21}$$

In order to better describe the whole process of scrambling, we present a simple 4 × 4 image scrambling example in Fig.8, which takes the **WR** generation process in scrambling combination ① as an example. The methods of other components and scrambling combinations are similar and will not be described in detail here.

D. MODULAR DIFFUSION

To further improve the security of the cryptosystem, we introduce the sequential mod-taking diffusion of elements after the XOR diffusion and the S-Box-based permutation. In the mod-taking diffusion, each newly generated element is related to the previous element, and the elements of each color component are also related to the other components. Therefore, the mod-taking diffusion can effectively improve the security of the encryption algorithm, and the detailed steps are as follows.

Step 1: Use the chaotic sequence **Y2** from the subsection B to generate the required sequence **B2** in the mod-taking diffusion according to Eq(22).

$$\boldsymbol{B2} = (abs(floor(\boldsymbol{Y2} \times 10^{12}))) \ mod \ 256 \tag{22}$$

Step 2: Perform modular diffusion operation for **WR**, **WG** and **WB** according to Eq(23).

$$\begin{cases} \boldsymbol{CR}(1) = (\boldsymbol{WR}(1) + \boldsymbol{B2}(1) + CZ) \ mod \ 256 \\ \boldsymbol{CR}(i) = (\boldsymbol{CR}(i-1) + \boldsymbol{B2}(i) + \boldsymbol{WR}(i)) \ mod \ 256 \\ \boldsymbol{CG}(1) = (\boldsymbol{WG}(1) + \boldsymbol{B2}(1) + \boldsymbol{CR}(MN)) \ mod \ 256 \\ \boldsymbol{CG}(i) = (\boldsymbol{CG}(i-1) + \boldsymbol{B2}(i) + \boldsymbol{WG}(i)) \ mod \ 256 \\ \boldsymbol{CB}(1) = (\boldsymbol{WB}(1) + \boldsymbol{B2}(1) + \boldsymbol{CG}(MN)) \ mod \ 256 \\ \boldsymbol{CB}(i) = (\boldsymbol{CB}(i-1) + \boldsymbol{B2}(i) + \boldsymbol{WB}(i)) \ mod \ 256 \\ i = 2, 3, \cdots, MN \end{cases} \tag{23}$$

Step 3: The obtained encrypted components **CR**, **CG** and **CB** are recombined to form the final color encrypted image **C**.

The decryption algorithm is the inverse of the encryption algorithm, which is not described in detail here.

V. EXPERIMENTAL SIMULATION AND SAFETY ANALYSIS

In this section, the simulation test of the proposed algorithm is carried out, and the encryption performance of the algorithm is analyzed at the same time. Our selected test images include 512*512*3 Lena, all black image, all white image and some color images from USC-SIPI [4]. Simulations are carried out on Windows10 environment using MATLAB 2020b [34]. The processor of the computer used is Intel(R) Core(TM) i5-7200 CPU @ 2.50 GHz, and the running memory of the computer is 12 GB.

A. EXPERIMENTAL SIMULATION

The essence of the image encryption algorithm is to transform the natural image to be transmitted into an unrecognizable noise-like image. Only the recipient can recover information of the original image by applying the correct key and the cipher image. Others cannot recover any usable information of the original image without the correct key even if they get the cipher image.

The simulation test results are shown Fig.9. It can be seen from Fig.9(a) that the original image has rich information. And from Fig.9(c), it can be seen that the encrypted image has no valid information. When the key is used, the decrypted image consistent with the original image can be obtained as shown in Fig.9(e). The simulation results show that our proposed algorithm can convert a rich color image into an unrecognizable ciphertext image.

B. KEY ANALYSIS

1) KEY SPACE ANALYSIS

An ideal image encryption algorithm should have as large a key space as possible so that it can resist possible brute-force exhaustive attacks. Reference [25] gives a standard for an

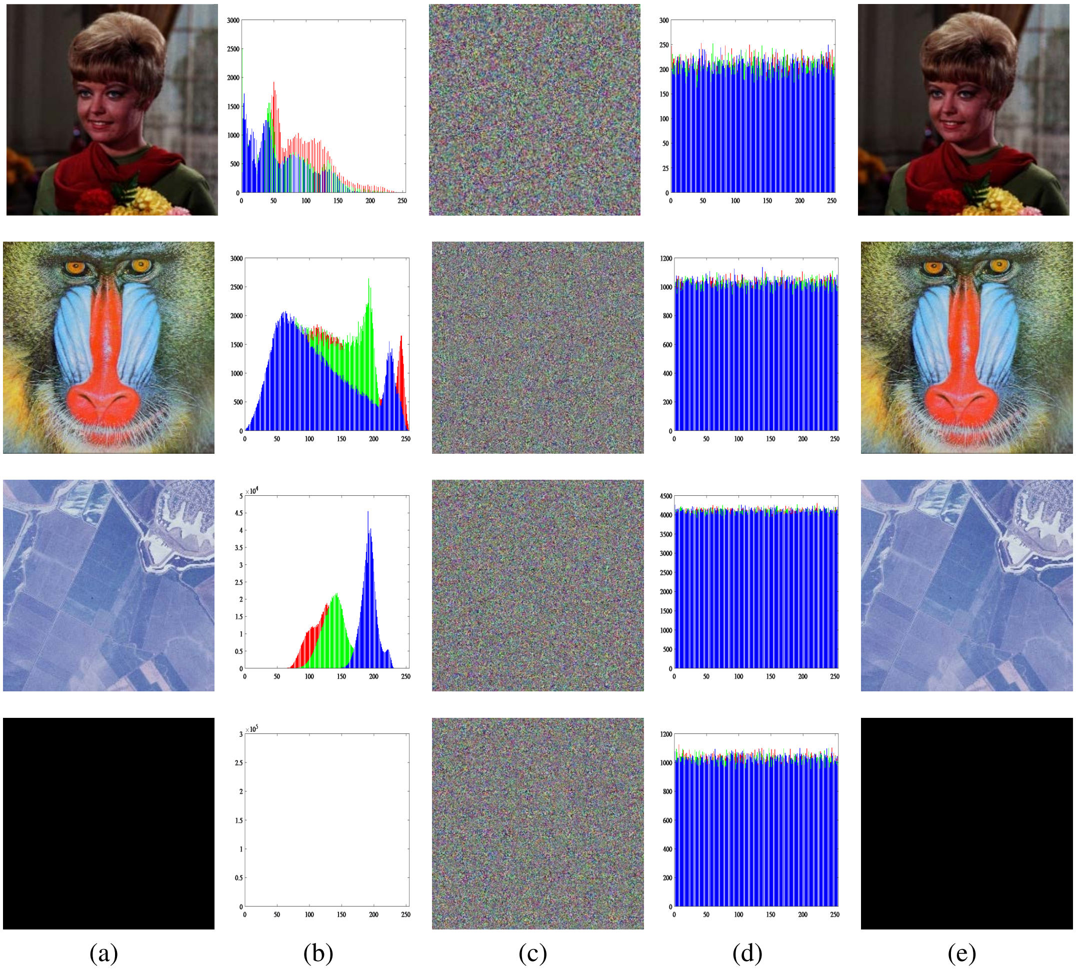


FIGURE 9. Simulation test of encryption algorithm: (a) plain images; (b) histogram of the plain images; (c) cipher Images; (d) histogram of the cipher images; (e) decrypted images.

ideal algorithm key space, which should be larger than 2^{100}. The key of the encryption algorithm proposed in this paper is divided into two parts: (1) initial conditions $\{x_0, y_0\}$ and control parameters $\{a_0, b_0\}$; (2) the number of chaotic sequences discards SQ. In this paper, we assume that the computational precision of the computer is 10^{-15}. Thus, the key space of this paper is $10^{60} \times 1921 \approx 2^{209}$, which is much larger than the key space standard mentioned in the reference [25].

2) KEY SENSITIVITY ANALYSIS

For a superior encryption algorithm, the key should not only have a large key space but also be extremely sensitive. In order to test the key sensitivity of our proposed algorithm, we encrypt the image with a key to obtain the ciphertext image Fig.10(b), and then decrypt to obtain the decrypted image Fig.10(e). Because the **Key** is divided into two parts, where the SQ is determined by the hash value. Therefore, we first change *Hash* to *Hash*+1 to test the sensitivity of the SQ in the key. Then, for another part, we add a tiny perturbation of 10^{-15} to each parameter of the key respectively to form four new keys, named **Key2**, **Key3**, **Key4**, and **Key5**. Here we use Lena as a test image.

Hash='6eca336ff63320b496048f0bc37edd1a8a5d764eefa85e54862984db7f301258';

Hash+1= '6eca336ff63320b496048f0bc37edd1a8a5d764eefa85e54862984db7f301259';

$$\boldsymbol{Key} = \{Hash, x_0, y_0, a_0, b_0\},$$
$$\boldsymbol{Key1} = \{Hash + 1, x_0, y_0, a_0, b_0\},$$
$$\boldsymbol{Key2} = \{Hash, x_0 + 10^{-15}, y_0, a_0, b_0\},$$
$$\boldsymbol{Key3} = \{Hash, x_0, y_0 + 10^{-15}, a_0, b_0\},$$
$$\boldsymbol{Key4} = \{Hash, x_0, y_0, a_0 + 10^{-15}, b_0\},$$
$$\boldsymbol{Key5} = \{Hash, x_0, y_0, a_0, b_0 + 10^{-15}\}.$$

In order to test the encryption sensitivity of the key, first we encrypt the original image with **Key2** to get the cipher image Fig.10(c), then we subtract it with the cipher image obtained with **Key** to get Fig.10(d). The result shows that the two cipher images have a significant difference. And the mathematical analysis of the two cipher images shows that there is 99.6105% difference between the two cipher images, indicating that the key is extremely sensitive in the encryption algorithm. Meanwhile, in order to test the decryption sensitivity of the key, we decrypt the ciphertext image

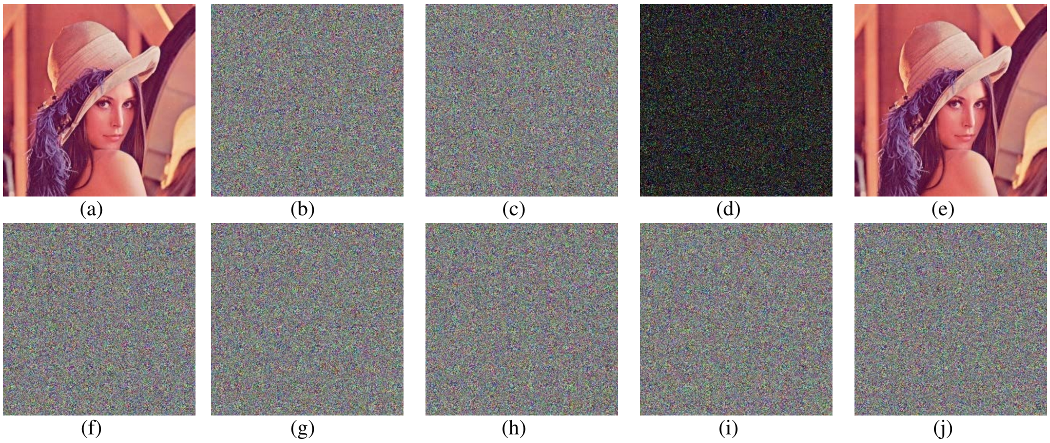


FIGURE 10. Key sensitivity analyses: (a) plain image; (b) the cipher image with Key; (c) the cipher image with Key2; (d) |(b)-(c)|; (e)the decrypted image with Key to (b); (f) the decrypted image with Key1 to (b); (g) the decrypted image with Key2 to (b); (h) the decrypted image with Key3 to (b); (i) the decrypted image with Key4 to (b); (j) the decrypted image with Key5 to (b).

TABLE 2. The calculation results of χ^2.

File name	Plain Image			Cipher Image		
	R	G	B	R	G	B
4.1.01(256 × 256 × 3)	134400.85	121852.27	120147.45	254.6563	252.6484	236.0625
Lena(512 × 512 × 3)	198013.67	93449.70	270610.96	269.5918	284.0137	246.4766
2.2.21(1024 × 1024 × 3)	633391.24	1956109.2	6750462.09	243.2061	258.9600	223.1099
All Black(512 × 512 × 3)	66585600	66585600	66585600	266.7266	241.6504	206.8867
All White(512 × 512 × 3)	66585600	66585600	66585600	279.0117	253.6348	227.9219

Fig.10(b) using the different keys containing **Key1**, **Key2**, **Key3**, **Key4** and **Key5** respectively. The results show that only the correct key can recover the original image, and other slightly altered key cannot recover the image information. It shows that the key has extremely strong decryption sensitivity. In summary, the proposed algorithm key not only has a large key space, but also has an extremely strong sensitivity.

C. STATISTICAL ANALYSIS

1) HISTOGRAM ANALYSIS

The histogram can well show the distribution of pixels in an image. Generally, the histogram of natural images has significant pixel distribution characteristics. An hacker can be able to get related information of the original image by analyzing the histogram information. Therefore, the pixels of the encrypted ciphertext image should be as evenly distributed as possible. The histogram of the original image is shown in Fig.9(b). It can be seen that the histogram distribution of the original image has obvious variability. The histogram of the ciphertext image is shown in Fig.9(d), and the results show that the histogram distribution of the encrypted image is uniform without obvious features. At the same time, in order to quantitatively analyze the distribution of image pixels, we use the chi-square test (χ^2) mentioned in the literature [26] to analyze the images, which is defined as

$$\chi^2 = \sum_{k=0}^{255} \frac{(f_k - f_e)^2}{f_e} \tag{24}$$

where f_k is the frequency of occurrence of pixel value k, $f_e = MN/256$, and M, N are the width and height of the image pixels respectively. When the test degree is 0.05, the standard value $\chi^2_{0.05} = 293.2478$ can be obtained. When the calculated χ^2 result is less than the standard value, it means that the chi-square test passes [26]. The calculation results of χ^2 for each component of different color images are shown in Table 2. The results show that the χ^2 of all encrypted image components is smaller than the standard value, which means that the pixel value distribution of the encrypted image is uniform.

2) CORRELATION ANALYSIS

A good image encryption algorithm must be able to break the correlation about the original images, as there is generally a strong correlation between pixels in natural images. The correlation between image pixels is mainly manifested in three directions including horizontal, vertical and diagonal. In this paper, we use the method of literature [21] to measure the magnitude of correlation by calculating the autocorrelation coefficient (AC), which is defined as

$$AC = \frac{E\left[(X_t - E[X_t])\right](X_{t+1} - E[X_t])}{E\left[(X_t - E[X_t])^2\right]} \tag{25}$$

where $\boldsymbol{X}_t$ and $\boldsymbol{X}_{t+1}$ are two adjacent pixel sequences, and E[•] denotes the mathematical expectation. The calculated value of the autocorrelation coefficient will fall into the range

TABLE 3. Correlation statistics of plain image and cipher image.

File name	Channel	Plain Image			Cipher Image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
4.1.01($256 \times 256 \times 3$)	R	0.9700	0.9525	0.9460	0.0013	0.00027	-0.0051
	G	0.9676	0.9585	0.9477	-0.0061	-0.0024	0.00005
	B	0.9526	0.9477	0.9359	-0.00035	0.0069	0.0034
4.2.03($512 \times 512 \times 3$)	R	0.9218	0.8624	0.8531	-0.0028	0.00028	0.0019
	G	0.8643	0.7591	0.7299	-0.0037	0.00044	0.00060
	B	0.9071	0.8782	0.8411	-0.0015	0.00021	-0.0011
2.2.20($1024 \times 1024 \times 3$)	R	0.9457	0.9441	0.9238	-0.00084	-0.00014	-0.0013
	G	0.9010	0.8976	0.8660	-0.00053	-0.0011	0.00038
	B	0.8253	0.8182	0.7837	-0.0014	0.00085	-0.00045
All Black($512 \times 512 \times 3$)	R	\	\	\	-0.00091	-0.00046	0.0018
	G	\	\	\	-0.0024	-0.0039	-0.00077
	B	\	\	\	-0.0047	-0.00010	-0.00059
All White($512 \times 512 \times 3$)	R	\	\	\	-0.00080	-0.00057	-0.0020
	G	\	\	\	0.00016	0.0026	0.00053
	B	\	\	\	0.0020	0.00099	-0.0023

TABLE 4. Comparison of correlation coefficients under different encryption algorithms (Lena).

Methods in Literature	Channel	Plain Image			Cipher Image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
[27]	R	0.9813	0.9803	0.9668	0.0092	0.0203	-0.0073
	G	0.9691	0.9594	0.9433	0.0002	-0.0025	-0.0131
	B	0.9455	0.9294	0.9099	0.0076	0.0006	0.0111
[28]	R	0.9777	0.9508	0.9259	0.0090	-0.0027	-0.0013
	G	0.9670	0.9370	0.9111	-0.0013	-0.0051	-0.0155
	B	0.9496	0.9171	0.8867	-0.0025	-0.0103	-0.0078
[29]	R	0.9775	0.9880	0.9737	-0.0021	0.0027	-0.00032
	G	0.9662	0.9817	0.9605	0.0017	0.0023	0.0010
	B	0.9304	0.9568	0.9219	0.0012	-0.0011	0.00069
Our	R	0.9775	0.9880	0.9737	0.0011	0.00041	0.00020
	G	0.9662	0.9817	0.9605	-0.0056	0.00050	0.00035
	B	0.9304	0.9568	0.9219	-0.00084	-0.0047	0.000080

of $[-1,1]$. When the absolute value of AC is smaller, it indicates that the correlation between pixels is smaller.

The statistical results of the correlation coefficients of each image component are shown in Table 3. The results show that the correlation coefficients about the encrypted images tend to be close to 0, which indicates that the proposed encryption algorithm can effectively reduce the correlation between pixels. Meanwhile, the comparison data of correlation under different encryption algorithms are shown in Table 4. The results show that the proposed algorithm is more destructive to the original image correlation compared to other encryption algorithms. In order to further visualize the change of correlation before and after encryption, the correlation statistics of Lena as an example are shown in Fig.11. The figures show that there are an obvious linear correlation between adjacent pixels of the original image, while the pixels of the encrypted image are uniformly distributed and almost no correlation. In summary, the proposed encryption algorithm can break the correlation between adjacent pixels of the original image.

D. INFORMATION ENTROPY

The information entropy is used to describe the uncertainty of the signal, and it can also be used to reflect average uncertainty of all pixel value [34]. For an image whose pixel value is in the range of [0, 255], the calculation expression of the information entropy(IE) is as

$$IE = -\sum_{i=0}^{255} f(x_i)\log_2 f(x_i) \tag{26}$$

where $f(x_i)$ denotes the frequency of occurrence of pixel value x_i. From Eq(26), it can be concluded that IE can obtain the theoretical maximum value of 8 when each pixel value has the same probability of occurrence. And the larger the calculated value is, the more uniform the pixel distribution is. The calculation results of IE values for different sizes of images are shown in Table 5. The results show that all plain images have relatively small information entropy. However, the IE values of encrypted images are all close to 8, indicating that the encrypted image pixels are distributed uniformly [18]. In addition, Table 6 shows the comparison results of the IE value for different encryption algorithms. The results show that the information entropy of the proposed algorithm is closer to the theoretical value 8, which indicates that the proposed algorithm has stronger security.

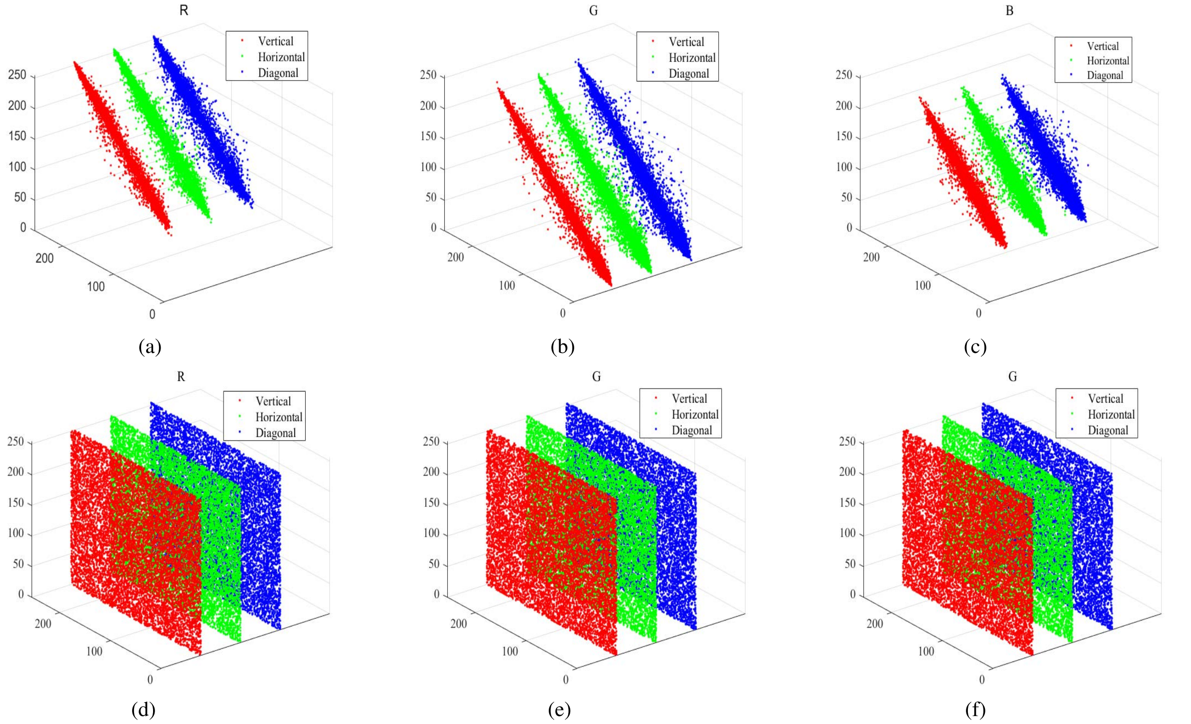


FIGURE 11. Visualize the correlation between pixels(Lena): (a) the correlation of the R component of the plain image in each direction; (b) the correlation of the G component of the plain image in each direction; (c) the correlation of the B component of the plain image in each direction; (d) the correlation of the R component of the ciphertext image in each direction; (e)the correlation of the G component of the ciphertext image in each direction; (f) the correlation of the B component of the ciphertext image in each direction.

TABLE 5. Information entropy of images.

Image size	File name	Plain Image			Cipher Image		
		R	G	B	R	G	B
$256 \times 256 \times 3$	4.1.01	6.4200	6.4457	6.3807	7.9972	7.9972	7.9974
	4.1.02	6.2499	5.9642	5.9309	7.9974	7.9974	7.9971
$512 \times 512 \times 3$	4.2.03	7.7067	7.4744	7.7522	7.9993	7.9993	7.9992
	4.2.05	6.7178	6.7990	6.2138	7.9992	7.9994	7.9993
$1024 \times 1024 \times 3$	2.2.20	6.8206	6.6007	5.6627	7.9998	7.9998	7.9998
	2.2.21	7.3256	6.6329	5.2769	7.9998	7.9998	7.9998
$512 \times 512 \times 3$	All Black	0	0	0	7.9993	7.9993	7.9994
	All White	0	0	0	7.9992	7.9993	7.9994

TABLE 6. Comparison of information entropy of different encryption algorithms (Lena).

Methods in Literature	File name	Plain Image			Cipher Image		
		R	G	B	R	G	B
[17]	Lena	7.2531	7.5940	6.9684	7.9912	7.9913	7.9914
[27]	Lena	7.2531	7.5940	6.9684	7.9980	7.9979	7.9978
[30]	Lena	7.2531	7.5940	6.9684	7.9895	7.9894	7.9894
[31]	Lena	7.2531	7.5940	6.9684	7.9966	7.9972	7.9967
Our	Lena	7.2531	7.5940	6.9684	7.9993	7.9992	7.9993

E. DIFFERENTIAL ATTACK ANALYSIS

Differential attack is a common cryptanalytic attack. Firstly, two plain images with minor differences are encrypted separately by an attacker. Then the attacker establishes an intrinsic connection between the ciphertext and the plain image by comparing the differences between the encrypted images. Finally, the purpose of decryption is achieved in this way [4]. Therefore, a good encryption algorithm should have excellent resistance to differential attacks.

In order to better analyze the ability of encryption algorithms to resist differential attacks, the Number of Pixels Change Rate (NPCR) and the Unified Average Changing

TABLE 7. Test passing criteria for NPCR and UACI.

Image Size	NPCR			UACI		
	$N^*_{0.05}$	$N^*_{0.01}$	$N^*_{0.001}$	$(U^{*-}_{0.05}, U^{*+}_{0.05})$	$(U^{*-}_{0.01}, U^{*+}_{0.01})$	$(U^{*-}_{0.001}, U^{*+}_{0.001})$
256 × 256	99.5693	99.5527	99.5341	(33.2824,33.6447)	(33.2255,33.7016)	(33.1594,33.7677)
512 × 512	99.5893	99.5810	99.5717	(33.3730,33.5541)	(33.3445,33.5826)	(33.3115,33.6156)
1024 × 1024	99.5994	99.5952	99.5906	(33.4183,33.5088)	(33.4040,33.5231)	(33.3875,33.5396)

TABLE 8. Calculation results of NPCR and UACI.

File name	NPCR(%)				UACI(%)			
	R	G	B	Average	R	G	B	Average
4.1.01(256 × 256 × 3)	99.6105	99.6081	99.6072	99.6086	33.4716	33.4772	33.4675	33.4721
4.2.05(512 × 512 × 3)	99.6083	99.6103	99.6105	99.6097	33.4721	33.4746	33.4521	33.4662
2.2.20(1024 × 1024 × 3)	99.6094	99.6083	99.6103	99.6093	33.4575	33.4709	33.4732	33.4672
All Black(512 × 512 × 3)	99.6090	99.6090	99.6107	99.6095	33.4637	33.4576	33.4424	33.4546
All White(512 × 512 × 3)	99.6090	99.6093	99.6083	99.6089	33.4683	33.4809	33.4430	33.4640

TABLE 9. Comparison of NPCR and UACI for different encryption algorithms (Lena).

Methods in Literature	NPCR(%)				UACI(%)			
	R	G	B	Average	R	G	B	Average
[17]	99.6243	99.6433	99.6029	99.6235	33.4686	33.5020	33.4155	33.4620
[18]	99.6479	99.6579	99.6288	99.64487	33.4390	33.4799	33.4833	33.4674
[27]	99.6531	99.6522	99.6518	99.65237	33.4572	33.4715	33.4384	33.4557
[30]	99.6052	99.6060	99.6113	99.6075	33.4280	33.4966	33.3779	33.4341
Our	99.6058	99.6096	99.6105	99.6086	33.4563	33.4783	33.4560	33.4635

Intensity (UACI) are often used as evaluation indicators. They are defined as

$$\begin{cases} NPCR(C_1, C_2) = \displaystyle\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{W(i,j)}{MN} \times 100\% \\ W(i,j) = \begin{cases} 0 & if \;\; C_1(i,j) = C_2(i,j) \\ 1 & if \;\; C_1(i,j) \neq C_2(i,j) \end{cases} \end{cases} \tag{27}$$

$$UACI(C_1, C_2) = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|C_1(i,j) - C_2(i,j)|}{MN \times 255} \times 100\% \tag{28}$$

where C_1 and C_2 denote two different encrypted images. And they are generated by encrypting two original images whose pixels differ by only one bit. MN is the total number of image pixels. The literature [32] gives the test passing criteria in Table 7, where the ideal expectation values of NPCR and UACI are 99.6094% and 33.4635% respectively [32]. When the calculated values of both NPCR and UACI pass the test standard, the closer the calculation result is to the theoretical value, the better the encryption effect and the stronger the ability to resist differential attacks. In order to calculate without loss of generality, in this paper, we randomly perform 100 times NPCR and UACI analyses on the test images, and then we have the average value of the 100 times data. The final calculation results are shown in Table 8, which shows that the tested images can pass the test and are close to the theoretical value. In addition, the comparison of NPCR and UACI for different encryption algorithms using Lena images as an example is shown in Table 9. And the results show that the proposed encryption algorithm is closer to the theoretical value, so our algorithm is more resistant to differential attacks.

F. NOISE ATTACK ANALYSIS

Because all transmission channels exist noise, images are susceptible to noise contamination during transmission. A good encryption algorithm should be able to recover as much of the plain image information as possible in a noise-contaminated ciphertext image. In order to test the ability of our proposed algorithm to resist noise attack, we first add different degrees of noise to the ciphertext image, then use the correct key to decrypt the ciphertext image, finally observe the recovery degree of the image under different noise pollution. The salt and pepper noise with density of 1%, 5%, 10%, 20% is added to the cipher image respectively, and the decryption results are shown in Fig.12. The results show that even if the added noise density reaches 20%, the proposed algorithm can still recover a large amount of image information, indicating that our algorithm has a strong ability to resist noise attack.

G. DATA LOSS ATTACK ANALYSIS

In the process of information transmission, there is a high possibility of data loss. At the same time, there are also attackers who want to make the correct image information unavailable to the receiver by maliciously interfering with the normal transmission of the information. It is desirable for the receiver to recover the original image as much as possible in case of

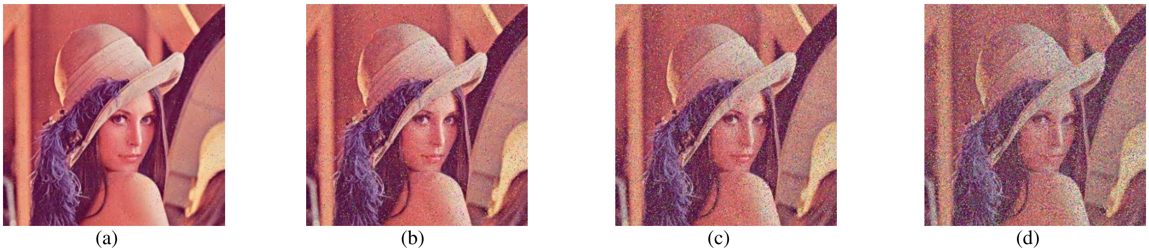


FIGURE 12. Salt and pepper noise test: (a) decrypted image with noise density of 1%; (b) decrypted image with noise density of 5%; (c) decrypted image with noise density of 10%; (d) decrypted image with noise density of 20%.

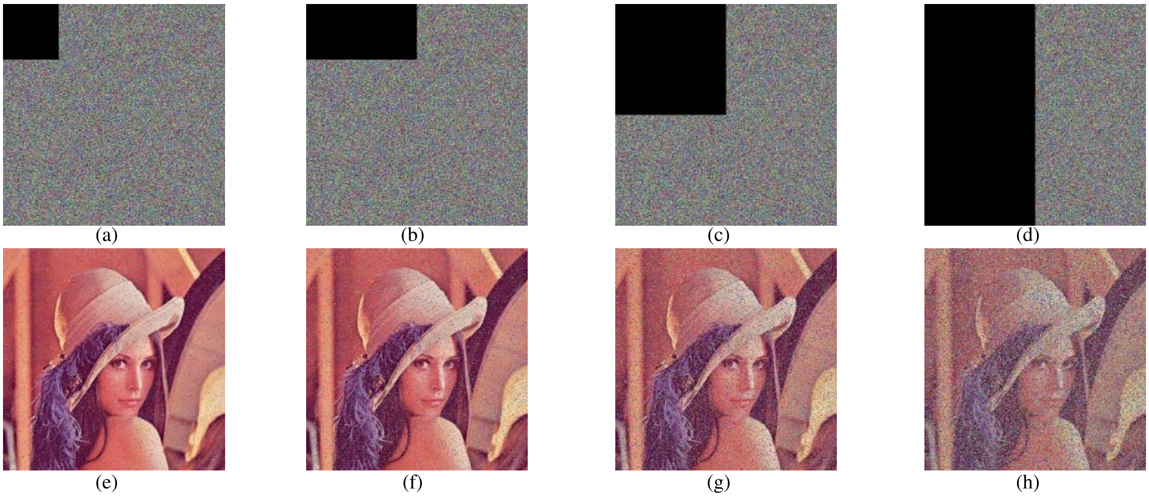


FIGURE 13. Analysis of information loss: (a) 1/16 loss of ciphertext information; (b) 1/8 loss of ciphertext information; (c) 1/4 loss of ciphertext information; (d) 1/2 loss of ciphertext information; (e) decrypted image of (a); (f) decrypted image of (b); (g) decrypted image of (c); (h) decrypted image of (d).

TABLE 10. Comparative analysis of encryption/decryption speed.

Methods in Literature	Image size	CPU(GHz)	Encryption time(s)	Decryption time(s)	Mean(s)
[17]	$512 \times 512 \times 3$	2.30	1.769703	0.837978	1.3038405
[31]	$512 \times 512 \times 3$	2.70	9.0016	9.1095	9.05555
[33]	$512 \times 512 \times 3$	3.45	1.3408	1.0158	1.1783
Our	$512 \times 512 \times 3$	2.50	0.40140	0.41895	0.41018

partial data loss. Therefore, in order to test the ability of our proposed algorithm to cope with information loss, we first perform the following four processes on the cipher image: (a) 1/16 of the data is lost in all components of the cipher image; (b) 1/8 of the data is lost in all components of the cipher image; (c) 1/4 of the data is lost in all components of the cipher image; (d) 1/2 of the data is lost in all components of the cipher image. Then, the cipher images with data loss are decrypted using the correct key. Finally the results are shown in Fig.13. The results show that most of the information of the image can still be recovered when 1/2 of the cipher image is lost. Thus, the proposed algorithm has the ability to resist information loss.

H. SPEED ANALYSIS

In order to better analyze the encryption and decryption speed of the proposed algorithm, the speed of different encryption algorithms is shown in Table 10. The results show that the encryption and decryption speed of the proposed algorithm is better than these algorithms. Therefore, the proposed algorithm has high encryption efficiency.

VI. CONCLUSION

In this paper, we propose a new two-dimensional discrete chaotic system, called 2D-SCCM, which is designed based on the study of the Logistic map and the cosine function. In order to better evaluate the chaotic behavior of the proposed system, we use various testing methods, including trajectory distribution map, Lyapunov exponent, sample entropy, sequence sensitivity. And the randomness of the generated time series was examined by using the NIST test tool. Experimental results and comparative analyses show that the 2D-SCCM has a wider hyperchaotic interval, more complex chaotic behavior, and better ergodicity compared to some existing chaotic systems. Based on the 2D-SCCM, we further propose a new color image encryption algorithm by combining the hash function. The algorithm consists of four main parts: key update, XOR-diffusion, pixel scrambling and substitution combined with S-Box, and Modular diffusion. Firstly, the updated key associates with the plain image, which makes the whole algorithm effective against the selective plaintext attack. Then, the XOR-diffusion and the Modular diffusion can effectively change the value of the pixels. Finally, in the pixel scrambling and substitution combined with S-Box stage, not only the value of the pixels can be changed, but also the position of the pixels, thereby breaking the correlation between the plain image pixels. Experimental simulations show that the proposed encryption algorithm can effectively convert natural images into unrecognizable noise-like images. The security analyses show that the algorithm is able to resist various common cryptanalysis attacks.

REFERENCES

[1] M. Fallahpour, "Reversible image data hiding based on gradient adjusted prediction," *IEICE Electron. Exp.*, vol. 5, no. 20, pp. 76–870, Oct. 2008.

[2] C. Wang, X. Wang, Z. Xia, and C. Zhang, "Ternary radial harmonic Fourier moments based robust stereo image zero-watermarking algorithm," *Inf. Sci.*, vol. 470, pp. 109–120, Jan. 2019.

[3] Z. Hua and Y. Zhou, "Design of image cipher using block-based scrambling and image filtering," *Inf. Sci.*, vol. 396, pp. 97–113, Aug. 2017.

[4] Z. Hua, Z. Zhu, Y. Chen, and Y. Li, "Color image encryption using orthogonal Latin squares and a new 2D chaotic system," *Nonlinear Dyn.*, vol. 104, pp. 4505–4522, May 2021.

[5] J. Wu, X. Liao, and B. Yang, "Image encryption using 2D henon-sine map and DNA approach," *Signal Process.*, vol. 153, pp. 11–23, Dec. 2018.

[6] Y. Luo, M. Du, and J. Liu, "A symmetrical image encryption scheme in wavelet and time domain," *Commun. Nonlinear Sci.*, vol. 20, no. 2, pp. 447–460, Feb. 2015.

[7] M. Ramzan, T. Shah, M. M. Hazzazi, A. Aljaedi, and A. R. Alharbi, "Construction of S-boxes using different maps over elliptic curves for image encryption," *IEEE Access*, vol. 9, pp. 157106–157123, 2021.

[8] Z. Hua, K. Zhang, Y. Li, and Y. Zhou, "Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing," *Signal Process.*, vol. 183, Jun. 2021, Art. no. 107998.

[9] J. Arif, M. A. Khan, B. Ghaleb, J. Ahmad, A. Munir, U. Rashid, and A. Y. Al-Dubai, "A novel chaotic permutation-substitution image encryption scheme based on logistic map and random substitution," *IEEE Access*, vol. 10, pp. 12966–12982, 2022.

[10] M. A. B. Farah, R. Guesmi, A. Kachouri, and M. Samet, "A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation," *Opt. Laser Technol.*, vol. 121, Jan. 2020, Art. no. 105777.

[11] P. Mani, R. Rajan, L. Shanmugam, and Y. H. Joo, "Adaptive control for fractional order induced chaotic fuzzy cellular neural networks and its application to image encryption," *Inf. Sci.*, vol. 491, pp. 74–89, Jul. 2019.

[12] Z. Hua, F. Jin, B. Xu, and H. Huang, "2D logistic-sine-coupling map for image encryption," *Signal Process.*, vol. 149, pp. 148–161, Aug. 2018.

[13] Y. Zhou, L. Bao, and C. L. P. Chen, "Image encryption using a new parametric switching chaotic system," *Signal Process.*, vol. 93, no. 11, pp. 3039–3052, 2013.

[14] J. Fridrich, "Image encryption based on chaotic maps," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Simulation*, vol. 2, Oct. 1997, pp. 1105–1110.

[15] A. Mansouri and X. Wang, "A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme," *Inf. Sci.*, vol. 520, pp. 46–62, May 2020.

[16] Z. Hua, J. Li, Y. Chen, and S. Yi, "Design and application of an S-box using complete Latin square," *Nonlinear Dyn.*, vol. 104, no. 1, pp. 807–825, Mar. 2021.

[17] L. Teng, X. Wang, F. Yang, and Y. Xian, "Color image encryption based on cross 2D hyperchaotic map using combined cycle shift scrambling and selecting diffusion," *Nonlinear Dyn.*, vol. 105, no. 2, pp. 1859–1876, Jul. 2021.

[18] Z. Hua, Z. Zhu, S. Yi, Z. Zhang, and H. Huang, "Cross-plane colour image encryption using a two-dimensional logistic tent modular map," *Inf. Sci.*, vol. 546, pp. 1063–1083, Feb. 2021.

[19] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Inf. Sci.*, vol. 297, pp. 80–94, Mar. 2015.

[20] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, pp. 459–467, Jun. 1976.

[21] Z. Hua and Y. Zhou, "Image encryption using 2D logistic-adjusted-sine map," *Inf. Sci.*, vol. 339, pp. 237–253, Apr. 2016.

[22] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *Amer. J. Physiol.-Heart Circulatory Physiol.*, vol. 278, no. 6, pp. H2039–H2049, Jun. 2000.

[23] Z. Hua, Y. Zhou, and H. Huang, "Cosine-transform-based chaotic system for image encryption," *Inf. Sci.*, vol. 480, no. 1, pp. 403–419, Apr. 2019.

[24] L. E. Bassham. (Sep. 2010). *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Accessed: Mar. 16, 2022. [Online]. Available: https://www.nist.gov/publications/statistical-test-suite-random-and-pseudorandom-number-generators-cryptographics

[25] G. Alvarez and L. I. Shujun, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. Bifurcation Chaos*, vol. 16, no. 8, pp. 2129–2151, Aug. 2006.

[26] G. Hu and B. Li, "Coupling chaotic system based on unit transform and its applications in image encryption," *Signal Process.*, vol. 178, Jan. 2021, Art. no. 107790.

[27] K. Xuejing and G. Zihui, "A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system," *Signal Process., Image Commun.*, vol. 80, Feb. 2020, Art. no. 115670.

[28] X. Wu, K. Wang, X. Wang, H. Kan, and J. Kurths, "Color image DNA encryption using NCA map-based CML and one-time keys," *Signal Process.*, vol. 148, pp. 272–287, Jul. 2018.

[29] J. Chen, J. Tang, F. Zhang, H. Ni, and Y. Tang, "A novel digital color image encryption algorithm based on a new 4-D hyper-chaotic system and an improved S-box," *Int. J. Innov. Comput., Inf. Control*, vol. 18, no. 1, pp. 73–92, Feb. 2022.

[30] X. Wu, J. Kurths, and H. Kan, "A robust and lossless DNA encryption scheme for color images," *Multimed Tools Appl.*, vol. 77, no. 10, pp. 12349–12376, 2018.

[31] A. U. Rehman, X. Liao, R. Ashraf, S. Ullah, and H. Wang, "A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2," *Optik*, vol. 159, pp. 348–367, Apr. 2018.

[32] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber J., Multidisciplinary J. Sci. Technol., J. Select. Areas Telecommu.*, vol. 1, pp. 31–38, Apr. 2011. [Online]. Available: http://www.cyberjournals.com/Papers/Apr2011/05.pdf

[33] G. Kaur, R. Agarwal, and V. Patidar, "Color image encryption scheme based on fractional Hartley transform and chaotic substitution–permutation," *Vis. Comput.*, vol. 38, no. 3, pp. 1027–1050, Mar. 2022.

[34] F. Zhang, X. Zhang, M. Cao, F. Ma, and Z. Li, "Characteristic analysis of 2D lag-complex logistic map and its application in image encryption," *IEEE MultimediaMag.*, vol. 28, no. 4, pp. 96–106, Oct. 2021.

ZEZONG ZHANG received the B.S. degree in electronic information science and technology from the College of Electronic Information Engineering, China West Normal University, China, in 2021. He is currently pursuing the degree with the College of Information Science and Engineering, Huaqiao University, China. His research interests include nonlinear dynamics and control, information security, cryptanalysis, and the application of chaotic cryptography in image processing.

JIANENG TANG received the B.Sc. degree in electronic information science and technology from Xijiang Normal University, China, in 2006, the M.Sc. degree in circuits and systems from Ningxia University, China, in 2009, and the Ph.D. degree in information and communication engineering from Southeast University, China, in 2012. He is currently an Associate Professor at the College of Engineering, Huaqiao University, China. He has published over 30 papers in journals and conferences. His research interests include image encryption, RF circuit design, complex network synchronization, and chaos synchronization and control.

FENG ZHANG received the B.Sc. degree in applied physics from the University of Electronic Science and Technology of China, China, in 2007. He is currently a Deputy General Manager of Fujian MM Electronics Company Ltd. His research interests include image encryption and RF circuit design.

HUI NI received the B.Sc. degree in project management from Fuzhou University, China, in 2014. He is currently a Deputy General Manager of Fujian MM Electronics Company Ltd. His research interests include image encryption and RF circuit design.

JINYUAN CHEN is currently pursuing the M.S. degree with the College of Engineering, Huaqiao University, Quanzhou, China. His research interests include information security, chaotic synchronization and control, and the application of chaotic cryptography in image processing.

ZHONGMING HUANG is currently pursuing the M.S. degree with the College of Engineering, Huaqiao University, Quanzhou, China. His research interests include information security and image privacy protection.