Hindawi
Security and Communication Networks
Volume 2021, Article ID 6679288, 16 pages
https://doi.org/10.1155/2021/6679288

Research Article

A Novel Image Encryption Algorithm Based on the Delayed Maps and Permutation-Confusion-Diffusion Architecture

Pengcheng He,[1] **Kehui Sun**,[1] **and Congxu Zhu**[2]

[1]*School of Physics and Electronics, Central South University, Changsha 410083, China*
[2]*School of Computer Science and Engineering, Central South University, Changsha 410083, China*

Correspondence should be addressed to Kehui Sun; kehui@csu.edu.cn

Received 29 December 2020; Revised 20 February 2021; Accepted 23 February 2021; Published 8 March 2021

Academic Editor: Angel M. Del Rey

To improve the dynamical behaviors of 1D chaotic maps, a new linear-delay-modulation method (LDM) is proposed. Derived from the Sine map, a delayed Sine map (DSM) is proposed based on the LDM. Then, we substitute the Sine map in the SIMM system with DSM and obtained a delayed SIMM system (DSIMM). Its chaotic performance is analyzed through the phase diagram, Lyapunov exponent spectrum, and complexity. The results show that the delayed chaotic map can generate more complex dynamical behaviors and more random sequences. Hence, we apply the two delayed systems to a novel image encryption algorithm with the permutation-confusion-diffusion architecture. Firstly, to permutate the pixel of the image efficiently, the plain-image is scrambled by using a multilayer of the nonlinear index. Secondly, the image is confused by using the chaotic matrix generated with two chaotic sequences, and then, the ciphertext is transformed into a 1D sequence. Finally, to improve the plaintext sensitivity and facilitate key management, we enhance the sensitivity by applying a novel diffusion algorithm instead of using plaintext-related keystream. The diffusion equation contains the sum of undiffused pixels and the operation of cyclic bit-shift. Simulation results for the gray image illustrate the effectiveness of the proposed encryption algorithm.

1. Introduction

With the development of information technology, digital information security has become a major concern for individuals and organizations. When it comes to text encryption, there exist several traditional encryption schemes, such as DES, AES, and IDEA [1–3]. Besides text security, the protection of digital images is exceedingly important. The chaotic system has many unique features similar to the counterpart of the cryptosystem, such as bulky data capacity and high redundancy [4], which are difficult to handle by using text encryption techniques. In the last two decades, many new image encryption schemes were proposed, and researchers proposed many image encryption schemes by applying various technologies, such as compressive sensing [5–7], DNA coding [8–13], quantum theory [14–17], chaos systems [18–28], S-box [29], and networks [30, 31].

In [32], a general permutation-diffusion architecture for chaos-based image encryption was employed as illustrated in Figure 1. In the permutation stage, the position of the pixels is scrambled. In the diffusion stage, the pixel values are modified so that a tiny change in one-pixel spreads out to as many pixels as possible. In [33], Li et al. proposed a new image encryption algorithm that joins the pixel plane and bit-plane shuffle based on the nonlinearly modulated logistic map with delay. The nonlinear modulation with the delay operation effectively improves the dynamical behaviors of the chaotic system. With the introduction of the Sine function, the iteration time of the chaotic system is greatly increased. In [34], Liu et al. proposed a new two-dimensional Sine ICMIC modulation map (2D-SIMM) and designed a fast image encryption scheme using cyclic shift permutation and pixel-level diffusion. In [35], Chen and Sun applied the DPFMC method to resist the dynamical degradation of the 2D-SCL map and combined the confusion and diffusion processes in one stage to improve the running speed. In [36], Zhu and Sun proposed a new image security scheme using the modified skew tent map to generate

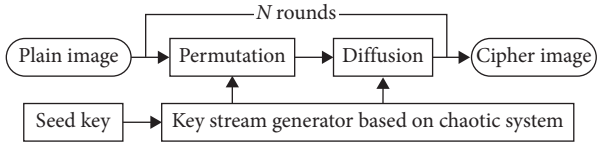


Figure 1: Permutation-diffusion architecture.

keystream correlated with plaintext. The initial-key generation stage uses plaintext to disturb the current chaotic number and then takes the disturbed chaotic number as the input of the system. In [5], Xu et al. proposed a fast image encryption algorithm, in which two circular measurement matrices measured a plain-image from two directions. Then, the image was reencrypted by employing row and column encryption to achieve permutation and diffusion simultaneously. Furthermore, the SHA-256 hash value of the plain-image is used to disturb the chaotic system's initial condition. Many existing image encryption algorithms correlate the plaintext with the initial condition of the chaotic system. In this way, with a different original image, the chaotic system's initial condition will be different. So, the differential attack resistance is strengthened. However, there are some defects of the traditional plaintext correlation method. Firstly, the secret key cannot be reused, and the key management is a challenging task because it contains the message of the plaintext. Secondly, the correlation method, such as SHA-256, is time-consuming and decreases the encryption algorithm's efficiency.

To overcome those weaknesses above, we enhance the Sine map and SIMM system's dynamical behaviors by introducing the LDM model and strengthening the resistance to differential attack using the novel diffusion algorithm. A high sensitive diffusion based on cyclic bit-shift is proposed. The rest of this paper is arranged as follows: In Section 2, we use the LDM model to improve the chaotic behaviors of the Sine map and the SIMM system. In Section 3, the image encryption algorithm of the permutation-confusion-diffusion architecture and its simulation results is presented. In Section 4, the security analysis is presented. Finally, we summarize the results.

2. DSM and 2D-DSIMM Models

2.1. LDM Model and Delayed Sine Map. The LDM model is displayed in Figure 2. The mathematical formula for this model is expressed as

$$x_{i+1} = Q\left(F\left(x_i \cdot L\left(x_{i-1}\right)\right)\right), \tag{1}$$

where F is a 1D chaotic map, L is a linear function, and Q denotes a qualified function. In this paper, we set F as Sine map, L as $(1 - x)$, and Q is the self-map, and the delayed Sine map (DSM) is obtained as follows:

$$x_{i+1} = \mu \sin\left(4\pi x_i\left(1 - x_{i-1}\right)\right), \tag{2}$$

where μ is the system parameter. When $\mu = 4$, the system has a Lyapunov exponent 3.9462. Therefore, the DSM is a chaotic map.

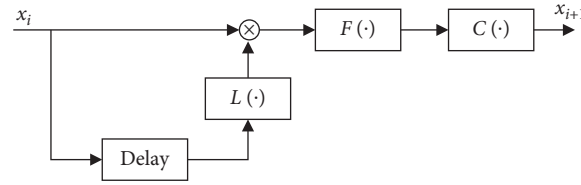


Figure 2: The LDM model.

2.2. 2D-DSIMM Model. The 2D-SIMM is obtained by the closed-loop coupling method (CMC) [34]. In the CMC model, when the Sine map is employed to modulate the output of ICMIC, the 2D-SIMM is defined as

$$\begin{cases} x_{i+1} = a \cdot \sin\left(\pi \cdot y_i\right) \cdot \sin\left(\dfrac{b}{x_i}\right), \\ \\ y_{i+1} = a \cdot \sin\left(\pi \cdot x_{i+1}\right) \cdot \sin\left(\dfrac{b}{y_i}\right). \end{cases} \tag{3}$$

We replaced the Sine map in 2D-SIMM with the DSM and obtained a two-dimensional delayed Sine ICMIC modulation map (2D-DSIMM):

$$\begin{cases} x_{i+1} = a \cdot \sin\left(4\pi \cdot y_i \cdot \left(1 - y_{i-1}\right)\right) \cdot \sin\left(\dfrac{b}{x_i}\right), \\ \\ y_{i+1} = a \cdot \sin\left(4\pi \cdot x_{i+1} \cdot \left(1 - x_i\right)\right) \cdot \sin\left(\dfrac{b}{y_i}\right), \end{cases} \tag{4}$$

where a and b are system parameters. When $a = 0.2$ and $b = 3$, the system has two positive Lyapunov exponents $(6.0788, 5.2529)$. Therefore, it is a hyperchaotic map.

2.3. Performance Evaluation of 2D-DSIMM. To evaluate the properties of the improved 2D-DSIMM, the chaos attractor, Lyapunov exponents, and approximate entropy were investigated.

2.3.1. Chaos Attractor. The chaos attractor shows the randomness of the chaotic sequence. A larger phase space means better ergodicity of the dynamical system.

Figure 3 shows the attractors of the 2D-DSIMM, 2D-SIMM, 2D-SLMM, and 2D-Logistic. The parameters of the 2D-DSIMM are set as $a = 5$ and $b = 5$. The parameters of the 2D-SIMM are set as $a = 1$ and $b = 5$. The parameters of the 2D-SLMM are set as 1 and 3, and the parameter of the 2D-logistic is set as 1.18. It is obvious that the 2D-DSIMM occupies a larger phase space and a more uniform distribution than that of the others.

2.3.2. Bifurcation and Lyapunov Exponent Spectrum. To compare the chaotic characteristics of the 2D-DSIMM and the 2D-SIMM system, the chaotic dynamical behaviors of the two systems are described by using bifurcation and Lyapunov exponent diagrams. Lyapunov exponents (LEs)

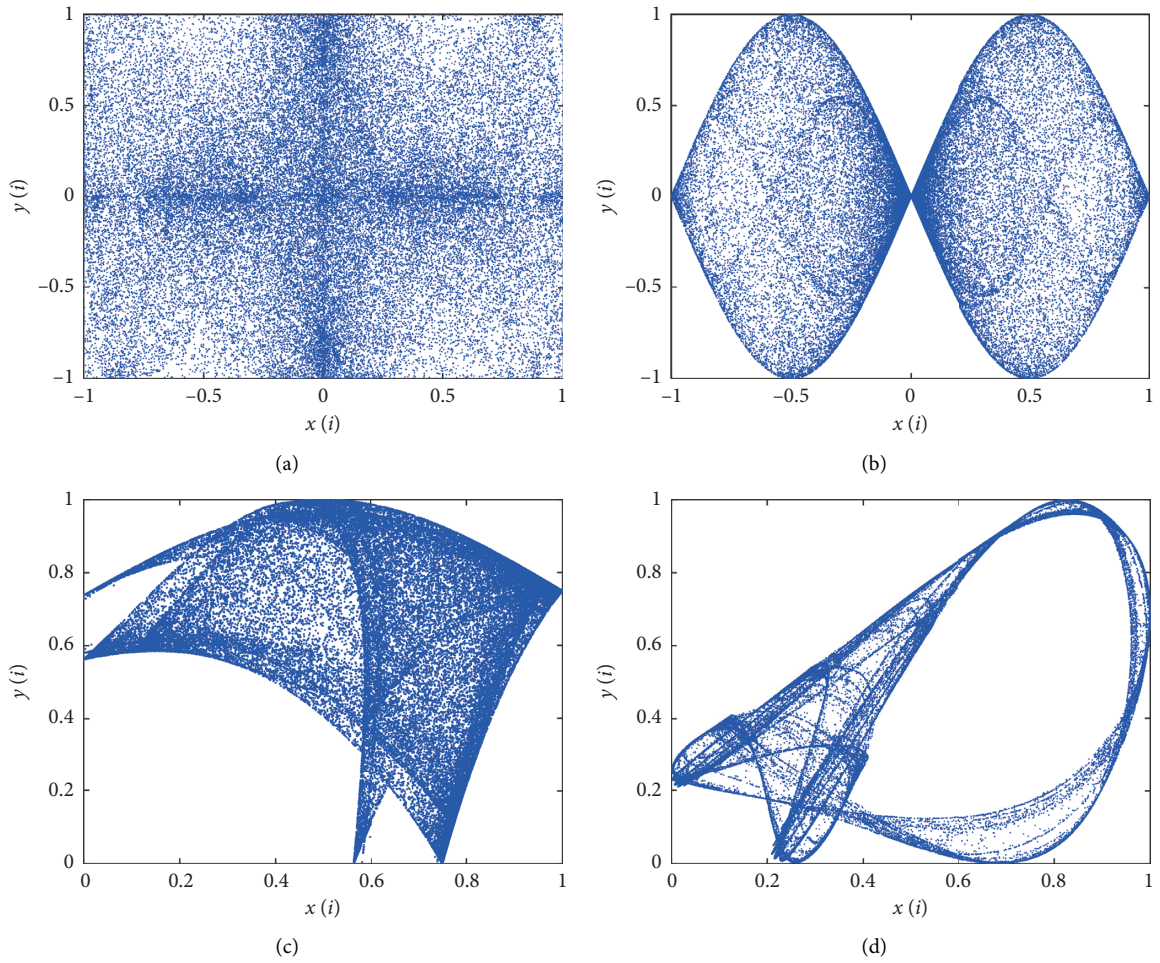


Figure 3: Attractors: (a) 2D-DSIMM with $a = 5, b = 5$; (b) 2D-SIMM with $a = 1, b = 5$; (c) 2D-SLMM with $\alpha = 1, \beta = 3$; (d) 2D-logistic map with $r = 1.18$.

measure the rate of divergence of orbits away from each other [37], and the maximum Lyapunov exponent (MLE) is concerned with the predictability of a chaotic system.

A 2D discrete chaotic map has two LEs. For the 2D-SIMM map and 2D-DSIMM map, Figure 4 presents their bifurcation and Lyapunov exponent diagrams with $b = 3$. The 2D-SIMM has obvious period windows. The 2D-DSIMM has almost none period windows and is hyperchaotic when $a \in [0.7, 2]$. The results indicate that the 2D-DSIMM has wider chaotic and hyperchaotic ranges than 2D-SIMM.

2.3.3. Approximate Entropy. Approximate entropy (ApEn) is a nonlinear index used to quantify the regularity and complexity of chaotic systems [38, 39]. Increasing ApEn value corresponds to the increasing unpredictability of the generated chaotic series. Figure 5 shows the ApEn of 2D-DSIMM, 2D-SIMM, 2D-SLMM, and logistic maps. As we can see, the ApEn complexity of 2D-DSIMM is higher than 2D-SIMM and is much larger than that of the other chaotic maps. It shows that 2D-DSIMM has a more complex dynamical performance than 2D-SIMM.

3. The Novel Image Encryption Algorithm

In this section, we present the permutation-confusion-diffusion architecture of the proposed algorithm, as shown in Figure 6. In this diagram, i denotes the row index, and j denotes the column index. k and l denote the 1D sequence index. As we can see, the proposed scheme is composed of key generation, permutation, confusion, and diffusion.

3.1. Key Generation. To resist brute-force attack, an image encryption algorithm's key length should be not less than 128 bits [40]. The key structure is presented in Figure 7. In particular, $K = a, b, x_{01}, y_{01}, x_{02}, y_{02}$. With the computer precision of 10^{-16}, each element has the size of 52 bits. Therefore, the length of this scheme's secret key is $6 \times 52 = 312$, and it is long enough to resist the attack from the most advanced computers nowadays. For the 52-bit binary string $b_1 b_2, \ldots, b_{312}$, we use the following equations to transform it to floating-point data:

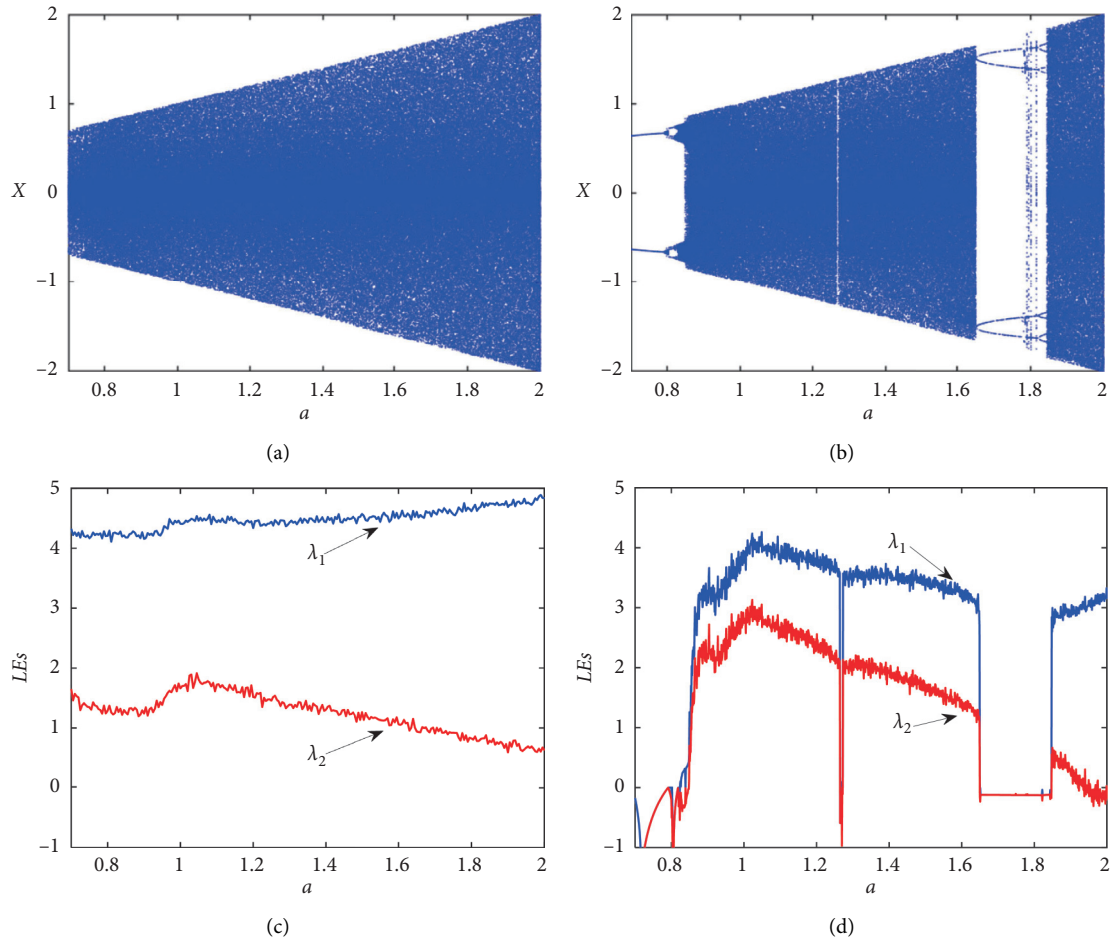


Figure 4: Bifurcation and Lyapunov exponents. (a) Bifurcation of 2D-DSIMM. (b) Bifurcation of 2D-SIMM. (c) Lyapunov exponent spectrum of 2D-DSIMM. (d) Lyapunov exponent spectrum of 2D-SIMM.

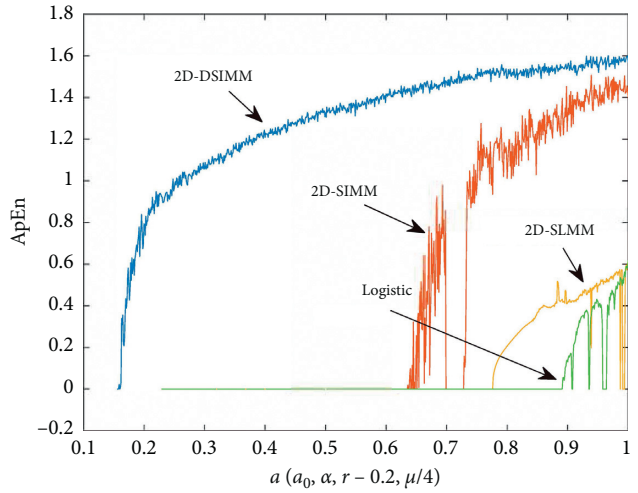


Figure 5: ApEn complexity of 2D-DSIMM, 2D-SIMM, 2D-SLMM, and logistic map.

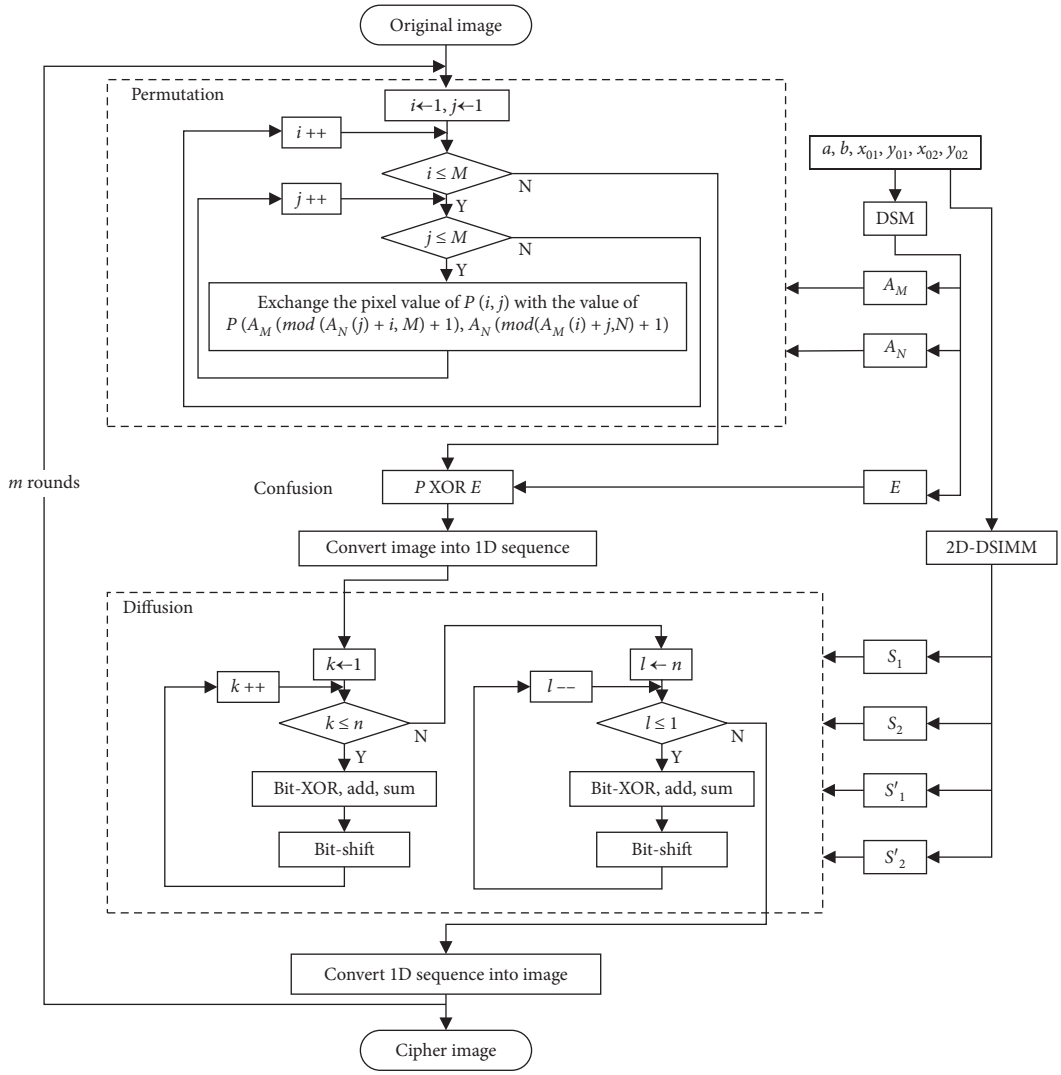


Figure 6: Block diagram of the image encryption scheme.

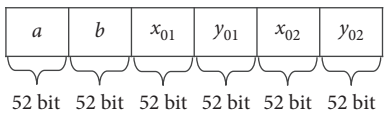


Figure 7: Secret key structure.

$$\begin{cases} a = \sum_{i=1}^{52} b_i \times 2^{-i}, \\ b = \sum_{i=52+1}^{2\times 52} b_i \times 2^{-(i-52)}, \\ x_{01} = \sum_{i=2\times 52+1}^{3\times 52} b_i \times 2^{-(i-2\times 52)}, \\ x_{02} = \sum_{i=3\times 52+1}^{4\times 52} b_i \times 2^{-(i-3\times 52)}, \\ y_{01} = \sum_{i=4\times 52+1}^{5\times 52} b_i \times 2^{-(i-4\times 52)}, \\ y_{02} = \sum_{i=5\times 52+1}^{6\times 52} b_i \times 2^{-(i-5\times 52)}. \end{cases} \tag{5}$$

$b_i (i = 1, 2, \ldots, 312)$ are integers which are obtained by directly transforming a 312-bit binary string to a decimal integer.

The initial value of the permutation and confusion $x_0^{(0)}$ is set as $((x_{01} + x_{02} + y_{01} + y_{02})/4)$. The initial value of the forward diffusion $(x_0^{(1)}, y_0^{(1)})$ is set as $(x_{01} y_{01})$. In backward diffusion, the initial value $(x_0^{(2)}, y_0^{(2)})$ is set as (x_{02}, y_{02}). The parameter in permutation and confusion is directly set as $\mu = 4$. The control parameters in forward and backward diffusion are generated by

$$\begin{cases} a^{(1)} = 0.7 + (a + x_{01}) \bmod 1, \\ b^{(1)} = 0.7 + (b + y_{01}) \bmod 1, \\ a^{(2)} = 0.7 + (a + x_{02}) \bmod 1, \\ b^{(2)} = 0.7 + (a + y_{02}) \bmod 1. \end{cases} \tag{6}$$

Using these parameters and initial states, the DSM and 2D-DSIMM can generate chaotic sequences for the following permutation, confusion, and diffusion.

3.2. Permutation and Confusion. To break the high correlations of the plain-image, we apply the multilayer of the nonlinear index to shuffle the pixels. The detailed procedures of permutation and confusion are described as follows:

Step 1: get the row and column of the plain-image P. M denotes the height of the matrix and N denotes the width of the matrix.

Step 2: using parameters $\mu = 4$, and initial state $((x_{01} + x_{02} + y_{01} + y_{02})/4)$, iterate the DSM for $(D + M + N)$ times, and abandon the previous D iterations to avoid the transient effect. The chaotic sequences X_M and X_N are obtained.

Step 3: quantify the sequences X_M and X_N, and obtain the sequence A_M and A_N. The quantification method is as follows:

$$\begin{cases} A_M = \mod\left(\text{floor}\left(X_M \times 2^{16}\right), M\right) + 1, \\ A_N = \mod\left(\text{floor}\left(X_N \times 2^{16}\right), N\right) + 1. \end{cases} \tag{7}$$

Step 4: from top to bottom and left to right, switch the pixel of position (i, j) with the pixel of the position $(A_M(\mod(A_N(j) + i, M) + 1), \quad A_N(\mod(A_M(i) + j, N) + 1))$, and we get the permutated image P_1.

Step 5: create a matrix E with the same size as P, and the matrix E is obtained by

$$E(i, j) = \mod\left(\text{floor}\left(\left(j \cdot X_N(i) + i \cdot X_M(j)\right) \times 2^{16}\right), 256\right). \tag{8}$$

Step 6: confuse the permutated image P_1 with the matrix E and get the image P_2. The confusion operation is $P_2 = E \oplus P_1$.

We demonstrate the permutation process by permutating a 5×5 pixel matrix, as shown in Figure 8. Figure 8(a) is the original 5×5 image, and Figure 8(b) shows the shuffling process using the proposed permutation algorithm. The permutation result of the proposed algorithm is shown in Figure 8(c). The row and column shuffling method achieved high time efficiency [41] as shown in Figure 8(d). The sorting method achieved a good shuffling effect [42] as shown in Figure 8(e). The proposed permutation method balanced the execution time and the shuffling effect. It effectively breaks the inner relationship of location between the pixels due to the multilayer of the nonlinear index. Figure 9 shows the result about one round permutation and confusion of the 256×256 gray Lena image. No fixture of the plain-image can be obtained after permutation. This permutation and confusion method is time-efficient since we only need to iterate DSM for $(M + N)$ times. The experimental environment is MATLAB 2016a with Intel (R) Core (TM) i5-6300HQ @ 2.6 Hz, and the random-access memory (RAM) adopted is 8 GB. The computation time of permutation and confusion is around 12 milliseconds.

3.3. Diffusion. To better resist differential attacks, we improve the plaintext sensitivity by enhancing the diffusion algorithm.

Figure 10 shows the elements we used in the diffusion function. In the equation of diffusion, we introduce the previously encrypted pixel and the sum of unencrypted pixels. The cyclic bit-shift is used to enhance the security of the algorithm further. Thus, the change of any pixel will affect every iteration of the diffusion.

There is a reasonable concern that the sensitivity will also exist in the decryption process. The avalanche effect does not happen in the decryption process. Decrypted image is not sensitive to the cipher image, and we can maintain robustness when it comes to noise attack or data loss. The simulation results of the robustness of the algorithm are given in Section 4.7.

The diffusion is divided into two steps: forward diffusion and backward diffusion. In the forward diffusion process, equation (10) is applied from the first pixel to the last pixel of the image. In the backward diffusion process, equation (11) is applied from the last pixel to the first pixel. We execute diffusion forward and backward during every round of encryption. The 2D-DSIMM diffusion is described as follows:

Step 1: converse the 2D image into 1D sequence.

Step 2: using parameter $(a^{(1)}, b^{(1)})$ and initial state $(x_0^{(1)}, y_0^{(1)})$, iterate the 2D-DSIMM for $D + M \cdot N$ times and dicard the first D points, so the chaotic sequences X_{DF} and Y_{DF} are obtained. Using parameter $(a^{(2)}, b^{(2)})$ and initial state $(x_0^{(2)}, y_0^{(2)})$, iterate the 2D-DSIMM for $D + M \cdot N$ times and dicard the first D points. The chaotic sequences X_{DB} and Y_{DB} are obtained.

Step 3: quantify the sequences X_{DF}, Y_{DF}, X_{DB} and Y_{DB}, and obtain the sequences S_1, S_2, S_1^b, S_1^b. The quantification method is defined by

$$\begin{cases} S_1 = \mod\left(\text{floor}\left(X_{\text{DF}} \times 2^{16}\right), F\right), \\ S_2 = \mod\left(\text{floor}\left(Y_{\text{DF}} \times 2^{16}\right), f\right), \\ S_1^b = \mod\left(\text{floor}\left(X_{\text{DB}} \times 2^{16}\right), F\right), \\ S_2^b = \mod\left(\text{floor}\left(Y_{\text{DB}} \times 2^{16}\right), f\right), \end{cases} \tag{9}$$

where F is the number of allowed pixel values in plain-image P and f is the gray level. $F = 256$ and $f = 8$ if P is $8 -$ bit grayscale image. Therefore, the elements of S_1 and S_1^b are the integer from 0 to 255, and the elements of S_2 and S_2^b are the integer from 0 to 7.

Step 4: execute forward diffusion:

$$\begin{cases} C_0(1) = T(1) \oplus S_1(1) \oplus \mod(\text{sum}(T(2 : \text{end})) + c0, F), \\ C(1) = C_0(1) \lll S_2(1), \\ C_0(i) = T(i) \oplus S_1(i) \oplus \mod(\text{sum}(T(i + 1 : \text{end})) + C(i - 1), F), \\ C(i) = C_0(i) \lll S_2(i), \\ C_0(n) = T(n) \oplus S_1(n) \oplus \mod(C(n - 1), F), \\ C(n) = C_0(n) \lll S_2(n), \end{cases} \tag{10}$$

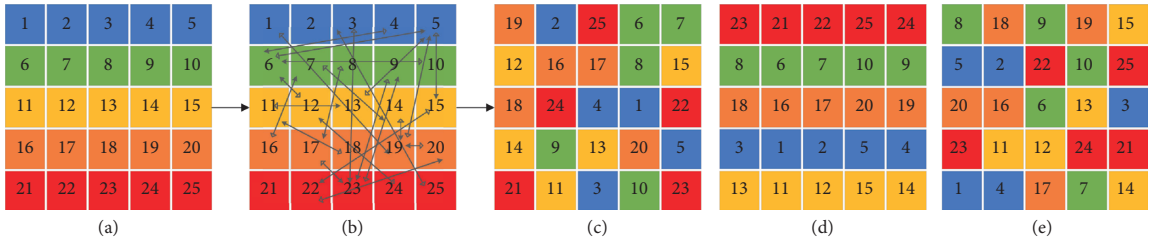


Figure 8: Demonstration of permutation, for $M = 5, N = 5, A_M = \{3, 2, 5, 4, 1\}, A_N = \{2, 3, 1, 5, 4\}$. (a) Original image P. (b) Process of pixel swapping. (c) Permutation result of the proposed algorithm. (d) Permutation result of row and column shuffling method. (e) Permutation result of sorting method.

Figure 9: Simulation results of one round permutation and confusion. (a) Plain-image P. (b) Permutation result of P, named P_{per}. (c) Confusion result of P_{per}, named P_1.

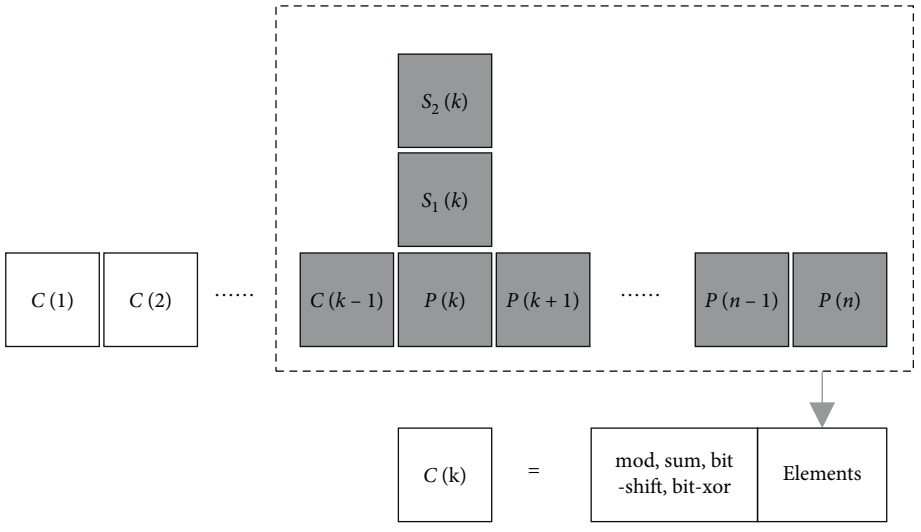


Figure 10: Operations and elements of the diffusion equation.

where T is the 1D sequence converted from the confused image P_2, n is the length of the permutated sequence T, and the $\lll$ is the operation of the cyclic left bit-shift. C is the forward diffused image.

Step 5: execute backward diffusion:

$$\begin{cases} C_0^b(n) = C(n) \oplus S_1^b(n) \oplus \mathrm{mod}\left(\mathrm{sum}\left(C(1:n-1)\right) + c^b 0, F\right), \\ C^b(n) = C_0^b(n) \lll S_2^b(n), \\ C_0^b(i) = C(i) \oplus S_1^b(i) \oplus \mathrm{mod}\left(\mathrm{sum}\left(C(1:i-1)\right) + C^b(i+1), F\right), \\ C^b(i) = C_0^b(i) \lll S_2^b(i), \\ C_0^b(1) = C(1) \oplus S_1^b(1) \oplus \mathrm{mod}\left(C^b(2), F\right), \\ C^b(1) = C_0^b(1) \lll S_2^b(1), \end{cases} \tag{11}$$

where C^b is the backward diffused image.

Step 6: convert the 1D sequence into a 2D digital image C.

The 2D-DSIMM diffusion in the decryption process is the inverse of the forward operation.

To show the diffusion algorithm's original image sensitivity, we apply the diffusion procedure on the 256×256 gray Lena image as shown in Figure 11. The proposed diffusion algorithm can turn the nature image into a meaningless one as shown in Figure 11(b). When using the same secret key to diffuse two plain-images with only a one-bit difference, we obtained Figure 11(c). After the forward and backward diffusion, a one-bit change of the original image can cause the avalanche effect, which can be seen from Figure 11(d). Thus, the diffusion algorithm has an excellent antidifferential attack property.

When we apply m round of the encryption, in the first $m-1$ rounds, we set the sum as constant 0. In the following section, we will analyze the encryption scheme of two rounds.

3.4. Decryption Algorithm. As shown in Figure 12, the steps of the decryption algorithm are similar to the steps of encryption. Firstly, we obtain the quantized pseudorandom sequence in the same way as encryption. Secondly, we transform the cipher image P into a 1D sequence. The decryption of 2D-DSIMM diffusion is represented as follows:

$$\begin{cases} C_0^b(1) = C^b(1) \ggg S_2^b(1), \\ T^b(1) = C_0^b(1) \oplus S_1^b(1) \oplus \mathrm{mod}\left(C^b(2), F\right), \\ C_0^b(i) = C^b(i) \ggg S_2^b(i), \\ T^b(i) = C_0^b(i) \oplus S_1^b(i) \oplus \mathrm{mod}\left(\mathrm{sum}\left(T^b(1:i-1) + C^b(i+1), F\right)\right), \\ C_0^b(n) = C^b(n) \ggg S_2^b(n), \\ T^b(n) = C_0^b(n) \oplus S_1^b(n) \oplus \mathrm{mod}\left(\mathrm{sum}\left(T^b(1:i-1) + c^b 0, F\right)\right), \end{cases}$$
$$\begin{cases} T_0^b(n) = T^b(n) \ggg S_2(n), \\ T(n) = T_0^b(n) \oplus S_1(1) \oplus \mathrm{mod}\left(T^b(n-1), F\right), \\ T_0^b(i) = T^b(i) \ggg S_2(i), \\ T(i) = T_0^b(i) \oplus S_1(i) \oplus \mathrm{mod}\left(\mathrm{sum}\left(T(i+1:\mathrm{end})\right) + T^b(i-1), F\right), \\ T_0^b(1) = T^b(1) \ggg S_2(1), \\ T(1) = T_0^b(1) \oplus S_1(1) \oplus \mathrm{mod}\left(\mathrm{sum}\left(T(1:\mathrm{end})\right) + c0, F\right), \end{cases} \tag{12}$$

where the $\ggg$ is the operation of right cyclic bit-shift. Thirdly, transform the obtained sequence into 2D matrix D_1 with the same size as the cipher image, and execute the anticonfusion operation $D_2 = E \oplus D_1$. Finally, we exchange the pixel value of $D_2(i,j)$ with $D_2 A_M(\mathrm{mod}(A_N(j) + i, M) + 1), A_N(\mathrm{mod}(A_M(i) + j, N) + 1)$, from right to left, bottom to top, and obtain the decrypted image. The decryption scheme is shown in Figure 12.

3.5. Simulation Results. Figure 13 shows the encryption result of three images with different sizes. One can observe that their cipher images are really noisy. The pixel distributions are meaningless to the attackers, and they can reveal nothing about the plain-image. With the right set of the secret key, the plain-image can be obtained through the decryption process.

4. Security Analysis

4.1. Histogram Analysis. The histogram can illustrate the statistical distribution of pixels visually. Figure 14 presents the histogram of plain-image Lena and the corresponding cipher image. Unlike the original image, the pixel distribution of the cipher image is relatively uniform. Thus, it is difficult for an attacker to extract useful information from the distribution pattern.

As a complement, we apply the histogram variance to quantitatively analyze the pixel distribution. A smaller histogram variance value means a more uniform distribution. The histogram variance is defined by

$$\mathrm{var}(Z) = \sum_{i=1}^{p} \sum_{j=1}^{p} \frac{\left(Z_i - Z_j\right)^2}{2p^2}, \tag{13}$$

where p denotes the grayscale value and Z_i denotes the number of the pixel with gray value i.

Table 1 shows the histogram variance of different 512×512 gray images using the proposed scheme and scheme from [43]. The results show that the histogram variance of the proposed scheme is smaller, which indicates a more uniform pixel distribution.

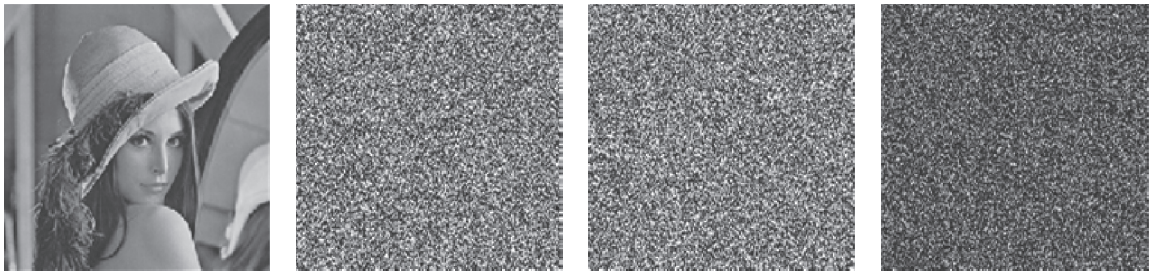
4.2. Chi-Square Test Analysis. The Chi-square test is used to further test the uniformity of the pixel values. A lower value of the Chi-square indicates a more uniform distribution. The Chi-square is defined as

$$\chi^2_{\mathrm{test}} = \sum_{i=0}^{255} \frac{\left(\mathrm{ob}_i - e_i\right)^2}{e_i}, \tag{14}$$

where ob_i denotes the observed frequencies of the pixel value i and e_i denotes the expected frequencies of the pixel value i. For a 256-level gray image, e_i can be expressed as

$$e_i = \frac{M \times N}{256}. \tag{15}$$

The Chi-square results of the encrypted images are listed in Table 2. In the table, the Chi-square test results are tested



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

Figure 11: Demonstration of 2D-DSIMM diffusion. (a) Plain-image I_1. (b) 2D-DSIMM diffusion result of I_1. (c) 2D-DSIMM diffusion result of I_2, where I_2 is another plain-image that has one-pixel difference with I_1 in position $(128, 128)$. (d) The difference of 2D-DSIMM diffusion results to I_1 and I_2.

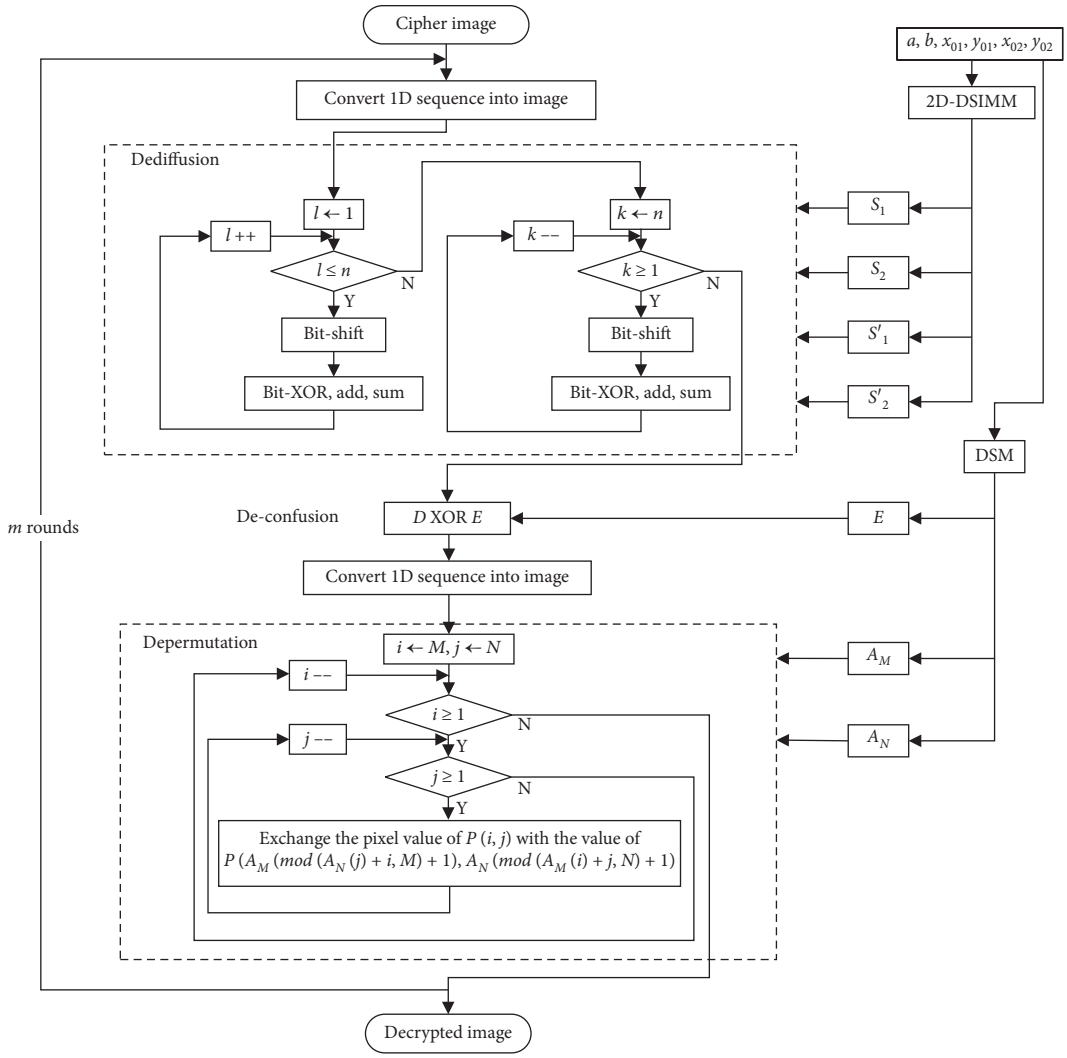


Figure 12: Block diagram of the image decryption scheme.

under the significance level of 0.01 and 0.05. At 0.01 and 0.05 significance level, the Chi-square values are $\chi^2_{0.01} = 310.4$ and $\chi^2_{0.05} = 293.2$, respectively. Evidently, the Chi-square value is acceptable at both 0.01 and 0.05 significance level.

4.3. Correlation Analysis. After randomly picking 10000 points inside the plain-image and its corresponding cipher image, their correlation coefficients ρ_{xy} of two adjacent pixels can be obtained by

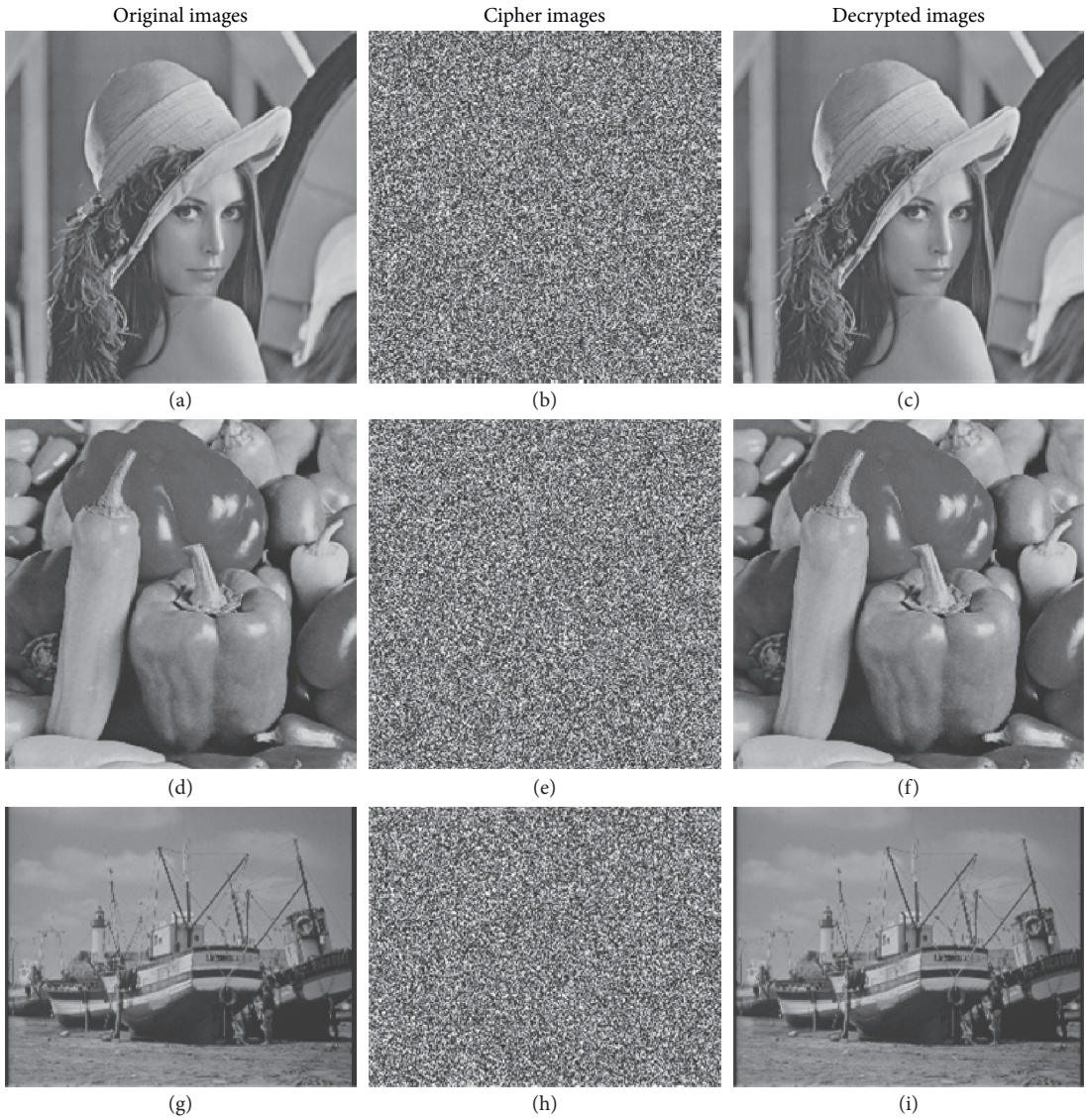


Figure 13: Original image and its cipher image, decrypted image.

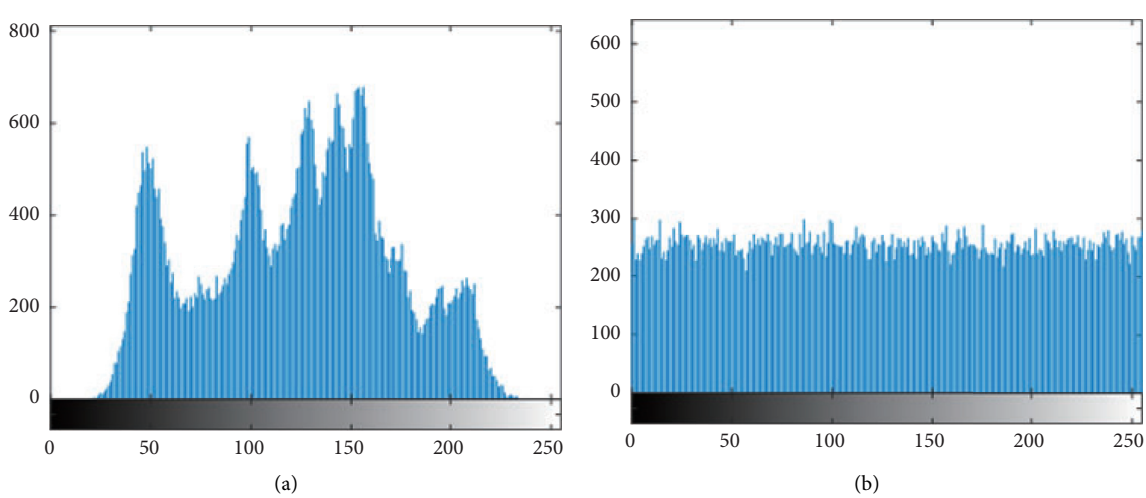


Figure 14: Histograms of the original image and cipher image. (a) Original Lena image. (b) Cipher Lena image after one round encryption.

Table 1: Comparison of histogram variance on different 512 × 512 gray images.

Image name	Histogram variance		
	Original images	Encrypted images	
		Proposed	Ref. [43]
Lena	633400	841.2	982.6
Peppers	780660	952.8	1087.6
Boats	1535900	952.1	1044.9
Elaine	562670	830.9	993.7

Table 2: Chi-square test results of different 512 × 512 gray images.

Image name	Chi-square	Critical value	
		$\chi^2_{0.05} = 293.2$	$\chi^2_{0.01} = 310.4$
Lena	242.8	Pass	Pass
Peppers	238.2	Pass	Pass
Boats	207.8	Pass	Pass
Elaine	239.9	Pass	Pass

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

$$\operatorname{cov}(x, y) = E\{[x - E(x)][y - E(y)]\},$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{16}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} [x_i - E(x)]^2.$$

Figure 15 depicts the correlation between adjacent pixels in the plain-image Lena and its corresponding cipher image in the horizontal, vertical, and diagonal direction. The plain-image exhibits a high correlation in all three directions. The points in Figure 15(b) appear to be uniformly distributed. Further, Table 3 shows their correlation coefficients quantitatively. The correlation coefficients of plain-image are nearly 1, while their cipher images are nearly 0. Table 4 compares the correlation coefficients of cipher image Lena under different encryption algorithms. Obviously, the proposed algorithm leads to a smaller correlation coefficient.

4.4. Information Entropy Analysis. The information entropy can be used to describe the uncertainties of grayscale in a digital image. The information entropy of the source m is defined as

$$H(m) = -\sum_{i=1}^{L} P(m_i)\log p(m_i), \tag{17}$$

where m_i is the ith gray value for an L level gray image and $p(m_i)$ is the probability of the gray value m_i. When $L = 256$, for an image with uniformly distributed pixel values, it has an information entropy of 8. To measure the randomness of an image at a local lever, Wu et al. proposed the local Shannon entropy [47]. The information entropy of encrypted images is listed in Table 5. The critical values of local entropy for 256×256 the gray image are given. The local entropy test results should be at the range of $h^*_{\alpha^-}$ and $h^*_{\alpha^+}$. Evidently, the global entropy of all cipher images is greater than 7.99, and the local entropy is acceptable in different significance levels.

4.5. Secret Key Analysis. To endure the brute-force attack, a good image encryption algorithm should have a large keyspace. The proposed algorithm has a secret key with a length of 312 bits under the computational precision of 10^{-15}. Even the most advanced computer system cannot check so many options at present.

To resist differential attack, an encrypted image should be sensitive to minor changes in the secret key. The results of the secret key sensitivity test are displayed in Figure 16. To test the key sensitivity, the 256×256 Lena image is encrypted with the K_1 key, which is the correct key. When we change one bit of each part successively, six different and incorrect secret keys $K_2 - K_7$ are obtained to decode the encrypted image. As we can see, even with the flipping of one bit in the key, the difference is quite large. Therefore, the key sensitivity test is positive.

4.6. Resisting Differential Attack Analysis. A number of pixels change rate (NPCR) and unified average changing intensity (UACI) are used to quantify the ability to resist differential attack. The NPCR and UACI between two encrypted images C_1 and C_2 are defined by

$$\mathrm{NPCR} = \frac{1}{M \times N}\sum_{i,j} D(i, j),$$

$$\mathrm{UACI} = \frac{1}{M \times N}\sum_{i,j} \frac{|C_1(i, j) - C_2(i, j)|}{255}, \tag{18}$$

$$D(i, j) = \begin{cases} 1, & \text{if } C_1(i, j) \neq C_2(i, j), \\ 0, & \text{if } C_1(i, j) = C_2(i, j). \end{cases}$$

Here, M and N denote the height and the width of the image, respectively. Firstly, for the test image Lena, we select a pixel of random position and flip its least significant bit. Then, we encrypt these two images using the same set of random key and calculate the values of NPCR and UACI. These processes are executed 1000 times. The results are presented in Table 6. The data show that a one-bit flip of the original image can lead to a pretty high resistance value even through one round encryption. Apparently, with two rounds of encryption, the value of NPCR is close to the ideal value 99.61%, and the value of UACI is close to the ideal value 33.46%.

4.7. Robustness Analysis. When the cipher image is transmitted in a real communication channel, it will be contaminated with noise. While we strengthen the sensitivity

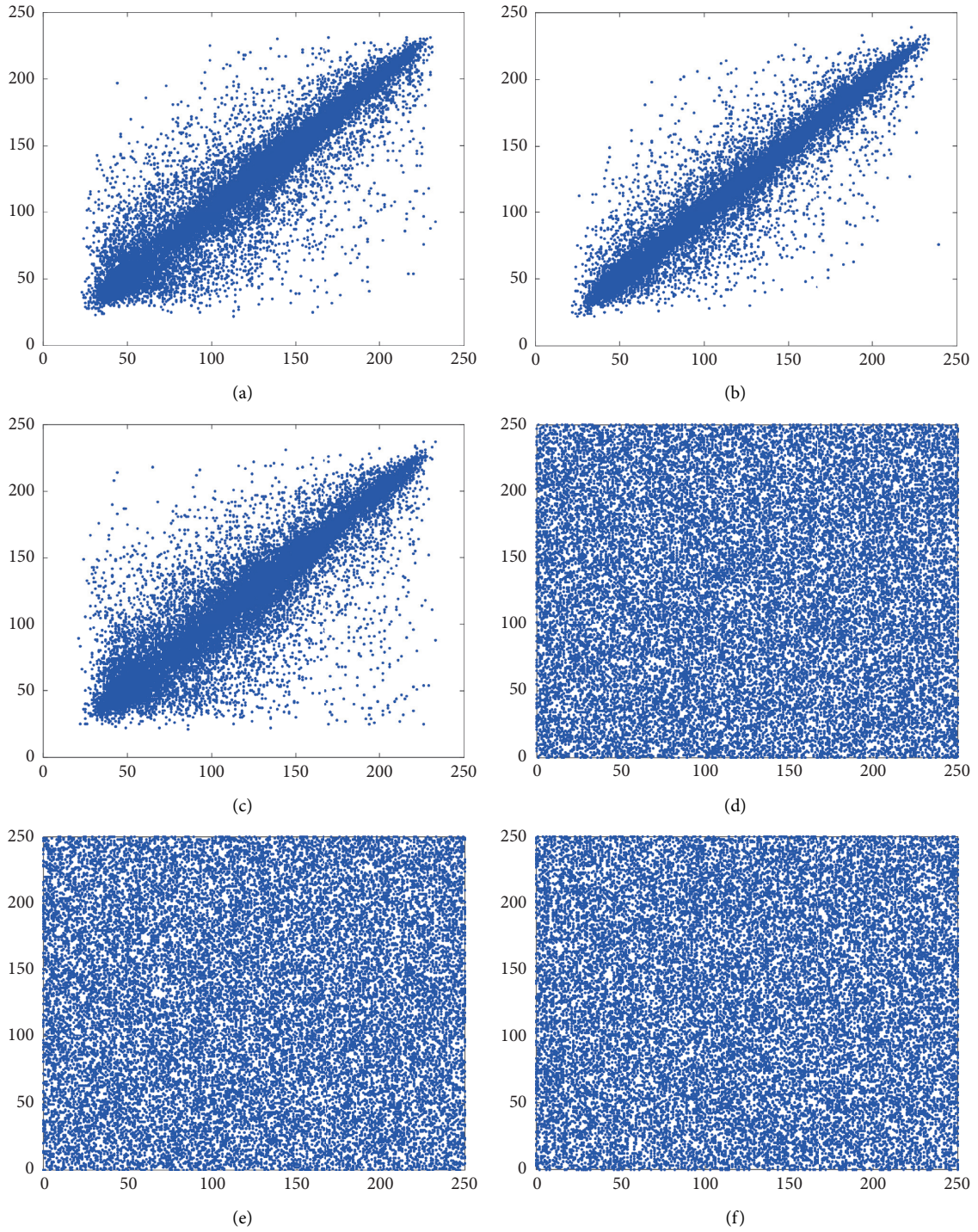


Figure 15: Correlation of adjacent pixels in the plain-image of 256×256 Lena. (a) Horizontal direction. (b) Vertical direction. (c) Diagonal direction. Correlation of two adjacent pixels in the cipher image of 256×256 Lena. (d) Horizontal direction. (e) Vertical direction. (f) Diagonal direction.

between the plaintext and the encrypted image during the encryption process, it is essential to ensure that the sensitivity does not exist between the cipher image and decrypted image during the decryption process, which will propagate error and weaken the stability of the encryption system.

Figure 17 demonstrates the robustness of the algorithm. In Figures 17(a) and 17(b), some part of the cipher image Lena is cropped, and their decryption images are shown in Figures 17(e) and 17(f). Figures 17(c) and 17(d) are the encrypted Lena images contaminated with pepper and salt

Table 3: Correlation coefficients between two adjacent pixels in different images.

Image name	Original image			Encrypted image		
	Horizontal	Diagonal	Vertical	Horizontal	Diagonal	Vertical
Lena	0.9460	0.9321	0.9725	0.0013	0.0002	0.0033
Peppers	0.9639	0.9414	0.9709	0.0021	−0.0024	0.0009
Boats	0.9370	0.9040	0.9571	0.0043	0.0017	0.0031
Man	0.9405	0.9133	0.9540	−0.0022	−0.0011	−0.0029

Table 4: Correlation coefficients of the encrypted Lena image with different algorithms.

Direction	Original image	Our scheme	Ref. [44]	Ref. [45]	Ref. [46]
Horizontal	0.9582	0.0013	0.0090	0.0015	−0.0029
Diagonal	0.9266	0.0002	0.0041	−0.0005	−0.0001
Vertical	0.9566	0.0033	0.0024	0.0021	0.0012

Table 5: Information entropy of 256×256 gray images.

Round	Image name	Global entropy	Local entropy	Critical value		
			No. of blocks = 30 Block size = $44 * 44$	$h^{*-}_{0.05} = 7.9019$ $h^{*+}_{0.05} = 7.9030$	$h^{*-}_{0.01} = 7.9017$ $h^{*+}_{0.01} = 7.9032$	$h^{*-}_{0.001} = 7.9015$ $h^{*+}_{0.001} = 7.9034$
1 round	Lena	7.9972	7.9024	Pass	Pass	Pass
	Peppers	7.9972	7.9028	Pass	Pass	Pass
	Boats	7.9972	7.9025	Pass	Pass	Pass
	Man	7.9972	7.9027	Pass	Pass	Pass
	Camera	7.9972	7.9025	Pass	Pass	Pass
2 rounds	Lena	7.9972	7.9027	Pass	Pass	Pass
	Peppers	7.9972	7.9025	Pass	Pass	Pass
	Baboon	7.9972	7.9026	Pass	Pass	Pass
	Man	7.9973	7.9027	Pass	Pass	Pass
	Camera	7.9972	7.9028	Pass	Pass	Pass

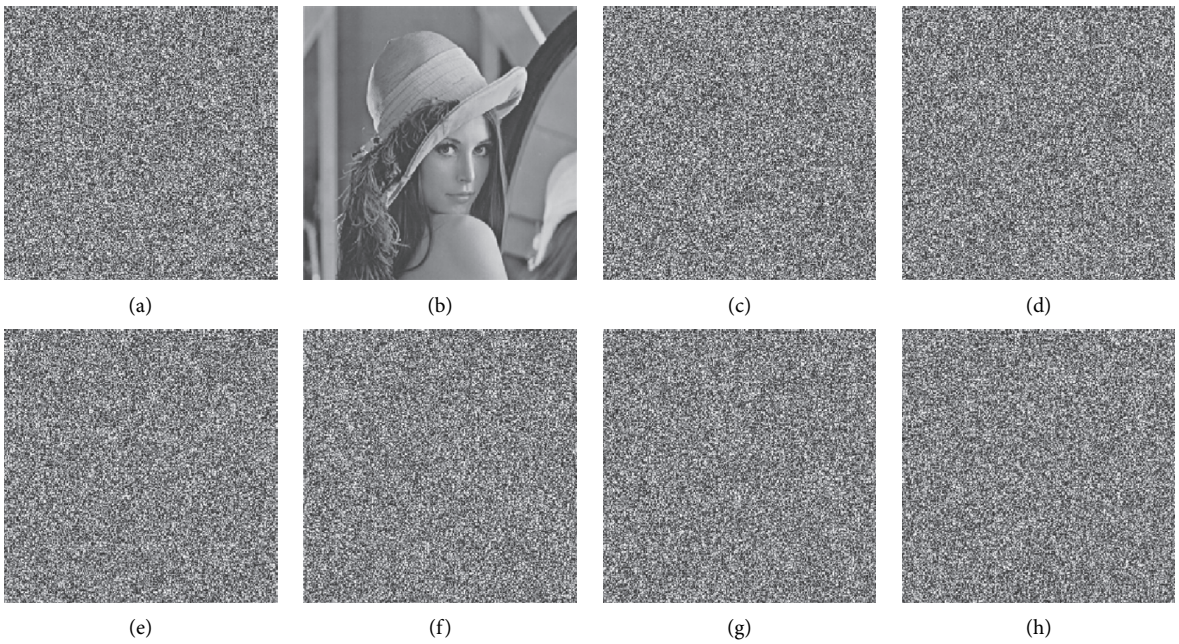


Figure 16: Secret key sensitivity test results under one round of the proposed algorithm. (a) Encrypted image. (b) Decrypted image with K_1. (c) Decrypted image with K_2. (d) Decrypted image with K_3. (e) Decrypted image with K_4. (f) Decrypted image with K_5. (g) Decrypted image with K_6. (h) Decrypted image with K_7.

Table 6: Values of NPCR and UACI.

Image name	Index	1 round			2 rounds		
		Minimum	Maximum	Mean	Minimum	Maximum	Mean
Lena	NPCR	0.9921	0.9983	0.9952	0.9955	0.9968	0.9961
	UACI	0.3316	0.3362	0.3342	0.3322	0.3364	0.3346
Peppers	NPCR	0.9923	0.9981	0.9951	0.9957	0.9964	0.9961
	UACI	0.3334	0.3358	0.3346	0.3334	0.3358	0.3346
Boats	NPCR	0.9920	0.9979	0.9950	0.9959	0.9963	0.9961
	UACI	0.3339	0.3354	0.3347	0.3340	0.3358	0.3346
Man	NPCR	0.9923	0.9980	0.9952	0.9960	0.9962	0.9961
	UACI	0.3336	0.3350	0.3345	0.3342	0.3349	0.3346

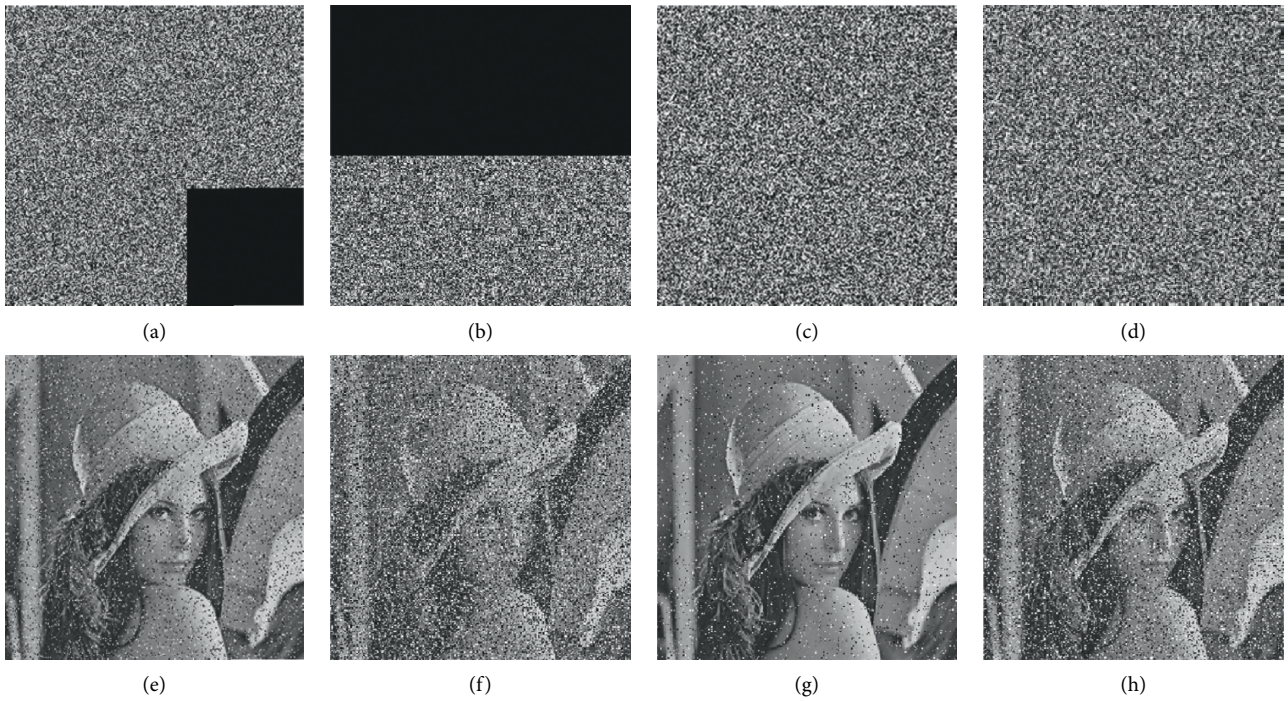


Figure 17: Robustness analysis. (a) The encrypted image with 100×100 data loss. (b) The encrypted image with 128×128 data loss. (c) Cipher Lena image polluted by pepper and salt noise with noise densities of 0.01. (d) Cipher Lena image polluted by pepper and salt noise with noise densities of 0.04. (e, f) The decrypted images of (a–d).

noise, and their decryption images are shown in Figures 17(g) and 17(h). The result demonstrates that the proposed algorithm is robust.

4.8. Time Complexity Analysis. To evaluate the proposed algorithm's computational complexity, the time complexity of generating sequences, permutation, confusion, and diffusion is given as follows [43, 50]. In the stage of sequence generation, we iterate the DSM $M + N$ times for permutation and confusion and iterate 2D-SIMM $2M \times N$ times for diffusion. In the permutation, confusion, and diffusion stage, the operation is executed pixel by pixel, so their computational complexity is $O(M \times N)$.

Let $M \times N$ be the size of all the k grayscale images, the computational complexity for once DSM iteration is denoted as D, and the computational complexity for one 2D-DSIMM is denoted as I. The computational complexity for XOR operation of one bit is denoted as X. Then, the computational complexity for the proposed algorithm is $D(M + N) + (2S + 9rX + 2I)MN$. Since decryption is the inverse operation of encryption, it has the same time complexity.

We use gray images with different sizes as the algorithm's input and perform the encryption 1000 times. In the proposed algorithm, The average value is the encryption time of the algorithm. The experimental environment is MATLAB 2016a with Intel (R) Core (TM) i5-6300HQ @ 2.6 Hz, and the random-access memory (RAM) adopted is 8 GB. Three gray images with different sizes are encrypted 100 times, and the average execution times are listed in Table 7. In comparison to the two earlier proposed algorithms [48, 49], our algorithm is faster and can meet the daily needs of secure image transmission.

Table 7: Encryption and decryption time for different size images under 2 rounds of the proposed algorithm.

Image name	Size	Encryption time (ms)			Decryption time (ms)
		Our scheme	Ref. [48]	Ref. [49]	Our scheme
Lena	256 × 256	76	127	172	82
Lena	512 × 512	282	516	701	297
Man	1024 × 1024	1021	2132	2593	1046

5. Conclusions

In this paper, we introduced LDM to enhance the chaotic behaviors of the 1D chaotic system, and a delayed Sine map and delayed SIMM map are proposed. The 2D-DSIMM has a better dynamical performance than the original map and some other recently proposed chaotic maps. The evidence is provided by analyzing its attractor, Lyapunov exponent spectrum, and approximate entropy. Based on these two maps, a new image encryption algorithm is proposed. The algorithm is composed of three procedures: permutation based on multilayer of the nonlinear index, confusion, and diffusion based on cyclic bit-shift. The permutation procedure efficiently scrambles the position of the pixels. With cyclic bit-shift operation and massive pixels' introduction into its diffusion equation, the diffusion procedure owns high sensitivity to the minor change of the plaintext image. It can quickly diffuse pixel values into the full image. Thus, we do not need to relate the secret key with the plaintext to maintain high resistance to differential attack. The results show that the proposed algorithm has high security and fast speed, and it can resist common attacks, including statistical attacks, differential attacks, and brute-force attacks. Simulation results show that our scheme has good application prospects in image encryption communication.

Data Availability

The data used to support the funding of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61901530 and 62071496).

References

[1] M. Ledda, B. Gerardo, and A. Hernandez, "Enhancing IDEA algorithm using circular shift and middle square method," in *Proceedings of the 2019 17th International Conference on ICT and Knowledge Engineering (ICT and KE)*, Bangkok, Thailand, November 2019.

[2] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Encryption Standard*, Springer Science and Business Media, Berlin, Germany, 2013.

[3] A. Hamdan and B. Bilal, "New comparative study between DES, 3DES and AES within nine factors," *Journal of Computing*, vol. 2, pp. 2151–9637, 2013.

[4] G. Bhatnagar and Q. Wu, "Biometric inspired multimedia encryption based on dual parameter fractional fourier transform," *IEEE Transactions on Systems Man and Cybernetics Systems*, vol. 44, pp. 1234–1247, 2017.

[5] Q. Xu, K. Sun, C. Cao et al., "A fast image encryption algorithm based on compressive sensing and hyperchaotic map," *Optics and Lasers in Engineering*, vol. 121, pp. 203–214, 2019.

[6] Q. Xu, K. Sun, C. Cao et al., "A visually secure asymmetric image encryption scheme based on RSA algorithm and hyperchaotic map," *Physica Scripta*, vol. 95, p. 35223, 2019.

[7] S. Zhu, C. Zhu, Y. Fu, W. Zhang, and X. Wu, "A secure image encryption scheme with compression-confusion-diffusion structure," *Multimedia Tools and Applications*, vol. 79, no. 43-44, pp. 31957–31980, 2020.

[8] W. Liu, K. Sun, Y. He, and M. Yu, "Color image encryption using three-dimensional Sine ICMIC modulation map and DNA sequence operations," *International Journal of Bifurcation and Chaos*, vol. 27, no. 11, Article ID 1750171, 2017.

[9] S. K. Pujari, G. Bhattacharjee, and S. Bhoi, "A hybridized model for image encryption through genetic algorithm and DNA sequence," *Procedia Computer Science*, vol. 125, pp. 165–171, 2018.

[10] J. Chen, Z.-l. Zhu, L.-b. Zhang, Y. Zhang, and B.-q. Yang, "Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption," *Signal Processing*, vol. 142, pp. 340–353, 2018.

[11] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Optics and Lasers in Engineering*, vol. 88, pp. 197–213, 2017.

[12] X. Ouyang, Y. Luo, J. Liu, L. Cao, and Y. Liu, "A color image encryption method based on memristive hyperchaotic system and dna encryption," *International Journal of Modern Physics B*, vol. 34, no. 4, Article ID 2050014, 2020.

[13] S. Zhu and C. Zhu, "Secure image encryption algorithm based on hyperchaos and dynamic DNA coding, entropy," *Physical Review E*, vol. 22, pp. 1–20, 2020.

[14] W. Hu, R. Zhou, J. Luo et al., "Quantum image encryption algorithm based on Arnold scrambling and wavelet transforms," *Quantum Information Processing*, vol. 19, pp. 1–29, 2020.

[15] N. R. Zhou, T. X. Hua, L. H. Gong, D. J. Pei, and Q. H. Liao, "Quantum image encryption based on generalized Arnold transform and double random-phase encoding," *Quantum Information Processing*, vol. 14, no. 4, pp. 1193–1213, 2015.

[16] G. Cao, J. Zhou, and Y. Zhang, "Quantum chaotic image encryption with one time running key," *International Journal of Security and Its Applications*, vol. 8, no. 4, pp. 77–88, 2014.

[17] H. Liu and C. Jin, "Color image encryption scheme based on Arnold scrambling and quantum chaotic," *International Journal of Network Security*, vol. 19, pp. 347–357, 2017.

[18] Y. Zhang and Y. Tang, "A plaintext-related image encryption algorithm based on chaos," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 6647–6669, 2018.

[19] M. Ahmad, E. Al Solami, X.-Y. Wang, M. Doja, M. Beg, and A. Alzaidi, "Cryptanalysis of an image encryption algorithm based on combined chaos for a BAN system, and improved scheme using SHA-512 and hyperchaos," *Symmetry*, vol. 10, no. 7, pp. 266–280, 2018.

[20] Q. Li, X. Wang, X. Wang, B. Ma, C. Wang, and Y. Shi, "An encrypted coverless information hiding method based on generative models," *Information Sciences*, vol. 553, pp. 19–30, 2021.

[21] K. A. K. Patro and B. Acharya, "A simple, secure, and time-efficient bit-plane operated bit-level image encryption scheme using 1-d chaotic maps," *Innovations in Soft Computing and Information Technology*, vol. 3, pp. 261–278, 2019.

[22] G. Silva, C. Flore, M. Renter et al., "Substitution box generation using chaos: an image encryption application," *Applied Mathematics and Computation*, vol. 332, pp. 123–135, 2018.

[23] Y. Xian and X. Wang, "Fractal sorting matrix and its application on chaotic image encryption," *Information Sciences*, vol. 547, pp. 1154–1169, 2021.

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

[25] C. Yu, J. Li, X. Li, X. Ren, and B. B. Gupta, "Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram," *Multimedia Tools and Applications*, vol. 77, no. 4, pp. 4585–4608, 2018.

[26] X. Wang and S. Gao, "Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network," *Information Sciences*, vol. 539, pp. 195–214, 2020.

[27] X. Wang, L. Feng, and H. Zhao, "Fast image encryption algorithm based on parallel computing system," *Information Sciences*, vol. 486, pp. 340–358, 2019.

[28] C. Xu, J. Sun, and C. Wang, "A novel image encryption algorithm based on bit-plane matrix rotation and hyper chaotic systems," *Multimedia Tools and Applications*, vol. 79, no. 9-10, pp. 5573–5593, 2020.

[29] Q. Lu, C. Zhu, and X. Deng, "An efficient image encryption scheme based on the LSS chaotic map and single S-box," *IEEE Access*, vol. 8, pp. 25664–25678, 2020.

[30] L. Zhang, R. Xiong, J. Chen et al., "Optical image compression and encryption transmission-based on deep learning and ghost imaging," *Applied Physics B*, vol. 126, pp. 1–10, 2020.

[31] F. Hu, J. Wang, X. Xu et al., "Batch image encryption using generated deep features based on stacked autoencoder network," *Mathematical Problems in Engineering: Theory, Methods and Applications*, vol. 1, pp. 1–12, 2017.

[32] R. Ye, "A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism," *Optics Communications*, vol. 284, no. 22, pp. 5290–5298, 2011.

[33] S. Li, B. Yin, W. Ding, T. Zhang, and Y. Ma, "A nonlinearly modulated logistic map with delay for image encryption," *Electronics*, vol. 7, no. 11, pp. 326–337, 2018.

[34] W. Liu, K. Sun, and C. Zhu, "A fast image encryption algorithm based on chaotic map," *Optics and Lasers in Engineering*, vol. 84, pp. 26–36, 2016.

[35] C. Chen and K. Sun, "An improved image encryption algorithm with finite computing precision," *Signal Processing*, vol. 168, Article ID 107340, 2019.

[36] C. Zhu and K. Sun, "Chaotic image encryption algorithm by correlating keys with plaintext," *China Communications*, vol. 9, pp. 73–79, 2012.

[37] B. Keith, "An improved method for estimating Liapunov exponents of chaotic time series," *Physics Lettres A*, vol. 151, pp. 27–32, 1990.

[38] K. H. Chon, C. G. Scully, and S. Lu, "Approximate entropy for all signals," *IEEE Engineering in Medicine and Biology Magazine*, vol. 28, no. 6, pp. 18–23, 2009.

[39] B. Delgado and A. Marshak, "Approximate entropy and sample entropy: a comprehensive tutorial," *Entropy*, vol. 21, p. 541, 2019.

[40] C. Li, D. Lin, J. Lu, and F. Hao, "Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography," *IEEE Multimedia*, vol. 25, no. 4, pp. 46–56, 2018.

[41] K. A. K. Patro and B. Acharya, "An efficient colour image encryption scheme based on 1-D chaotic maps," *Journal of Information Security and Applications*, vol. 46, pp. 23–41, 2019.

[42] K. A. K. Patro, M. P. J. Babu, K. P. Kumar et al., "Dual-layer DNA-encoding-decoding operation based image encryption using one-dimensional chaotic map," in *Proceedings of the 2019 International Conference on Data Intelligence and Security (ICDIS)*, South Padres Island, TX, USA, June 2019.

[43] K. A. K. Patro, A. Soni, P. K. Netam, and B. Acharya, "Multiple grayscale image encryption using cross-coupled chaotic maps," *Journal of Information Security and Applications*, vol. 52, Article ID 102470, 2020.

[44] Y. Zhang, "The unified image encryption algorithm based on chaos and cubic S-Box," *Information Sciences*, vol. 450, pp. 361–377, 2018.

[45] X. Wang and L. Liu, "Image encryption based on hash table scrambling and DNA substitution," *IEEE Access*, vol. 99, no. 1, 2020.

[46] H. Zhang, X. Wang, H. Xie et al., "A color image encryption method based on memristive hyperchaotic system and DNA encryption," *International Journal of Modern Physics B*, vol. 34, Article ID 2050014, 2020.

[47] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Information Sciences*, vol. 222, pp. 323–342, 2013.

[48] M. Alawida, J. S. Teh, A. Samsudin, and W. H. Alshoura, "An image encryption scheme based on hybridizing digital chaos and finite state machine," *Signal Processing*, vol. 164, pp. 249–266, 2019.

[49] P. Naskar, S. Bhattacharyya, D. Nandy et al., "A robust image encryption scheme using chaotic tent map and cellular automata," *Nonlinear Dynamics*, vol. 100, p. 3, 15.

[50] K. A. K. Patro and B. Acharya, "A novel multi-dimensional multiple image encryption technique," *Multimedia Tools and Applications*, vol. 79, no. 5, 2020.