Research Article

An Image Encryption Algorithm Based on Multichaotic System and DNA Coding

Jiming Zheng,1 Zheng Luo,2 and Zhirui Tang3

1School of Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
2College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
3Department of Mathematics and Natural Sciences, University of Wuppertal, Wuppertal 42119, Germany

Correspondence should be addressed to Jiming Zheng; zhengjm0903@qq.com

Received 20 June 2020; Revised 19 August 2020; Accepted 11 September 2020; Published 27 September 2020

Academic Editor: Nikos I. Karachalios

Copyright © 2020 Jiming Zheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, an improved two-dimensional logistic-sine coupling map (N2D-LSCM) and an improved Henon map (NHenon) are proposed. Furthermore, by combining N2D-LSCM and NHenon map, an image encryption algorithm is proposed based on these two chaotic systems and DNA encoding. The chaotic sequences generated by N2D-LSCM are used as the parameters of NHenon. In the scrambling stage, DNA encoding is carried out for pixels after scrambling by two chaotic sequences generated by N2D-LSCM; in the stage of diffusion, DNA random coding acts on random matrix obtained by two chaotic sequences generated by NHenon, and DNA XOR operation is carried out with the image obtained in the scrambling stage to diffuse. Compared with other 2D map for image encryption algorithm, this algorithm exhibits good security and holds high efficiency.

1. Introduction

In recent years, with the rapid development of network communication technology, a large number of digital information is transmitted through the network at every moment. It includes not only text message and images but also audio, videos, and other digital information [1]. How to protect this private information is an important research issue. Image encryption is one of the useful methods to protect image information. Because the digital image has the characteristics of large data capacity, high information redundancy, and high correlation between adjacent pixels, and the traditional text encryption methods, such as DES and AES [2, 3], are no longer suitable for image encryption. The researchers introduced chaos theory into image encryption, and the properties of chaos like randomness, sensitive dependence on initial value, and unpredictability made chaos closely connected with cryptography [4]. In recent 10 years, with the increasing number of proposed chaotic systems, many image encryption algorithms have emerged.

Scrambling-diffusion is the classical structure of image encryption based on chaotic system [5], so image encryption mainly contains two stages: scrambling and diffusion. The scrambling phase changes the position of image pixels, and the diffusion phase conducts pixel values replacement and performs mutual diffusion among different pixels. For an encryption algorithm based on chaos, its encryption efficiency and security are mainly determined by the structure of encryption algorithm and the performance of chaotic maps.

In order to increase the efficiency of encryption algorithm, many encryption algorithms only use one-dimensional or two-dimensional chaotic system [6–9]. Because the chaotic system is deterministic, if the attacker obtains the corresponding chaotic orbit information, it is possible to use the existing analysis technology to analyze the structure type of the chaotic system and reduce the complexity of the key, so as to decode [10]. Therefore, the security of low-dimensional mapping is not enough. Some encryption algorithms use a high-dimensional chaotic system, which makes the encryption speed low and cannot meet the real-time requirements [11, 12].

The combination of multiple sets of low-dimensional maps can improve the security and efficiency of the
algorithm [13, 14]. In [13], Hua et al. proposed a composite structure of chaotic mapping based on cosine transform, which can compose many chaotic maps and has better chaotic performance. In [15], Abdelfatah obtained a new one-dimensional chaotic map DCPG by combining Chebyshev chaotic map with tent map; as a result, it largely increases the key space. Based on DCPG, a new encryption algorithm was proposed. Encryption was completed by two rounds of exclusive OR operations, with high encryption efficiency.

DNA molecules have the characteristics of large-scale parallel, huge storage space, and low power consumption. In recent years, many researchers have successfully combined DNA computing and chaotic systems into the field of image encryption [16, 17]. In [16], Wang et al. proposed an image encryption method using DNA sequence operation and introduced a high-dimensional chaotic map as a chaotic sequence generator and only uses a single fixed DNA coding rule. In [17], Niyat and Moattar proposed a color image encryption method based on hyperchaotic system and DNA sequence. Image encryption adopts the idea of block. Each block randomly selects a DNA coding rule. However, in the scrambling stage and the diffusion stage, two kinds of three-dimensional chaotic systems are used, resulting in large amount of calculation and low encryption efficiency. Dagdub et al. put forward a new image encryption method based on DNA sequence operation and chaos system [18]. Using random DNA coding rules and low-dimensional chaotic maps, the encryption effect and security were improved. Amani and Yaghboobi put forward a new color image adaptive encryption algorithm based on DNA sequence operation and hyperchaos system [19]. Due to the chaotic maps selected by the above algorithms, they have shortcomings in security or efficiency.

In order to overcome the problems of the above algorithms, we proposed an improved two-dimensional logistic-sine coupling map (N2D-LSCM) and an improved Henon map (NHenon) in this paper. Further, we propose an image encryption scheme based on multiple chaotic systems and DNA coding. Our scheme use scrambling-diffusion structure. Two chaotic sequences are generated by N2D-LSCM in the scrambling process. On the one hand, these two chaotic sequences are used to permute original image and DNA encoding, respectively. On the other hand, in the diffusion stage, firstly, the sequence value generated by N2D-LSCM is taken as the parameter values of NHenon, and then two chaotic sequences are obtained by iterating the NHenon mapping, one as the key matrix and the other as the corresponding DNA coding rule matrix used in diffusion stage. The key matrix is encoded by the DNA coding rule matrix. Finally, DNA XOR operation is performed on the DNA molecules between the permuted DNA matrix and the encoded key matrix to complete the image encryption. The simulation results show that this method has good security and can effectively resist many kinds of common attacks, such as statistical attack, brute-force, chosen/known plaintext attack, clipping attack, and noise attack.

The rest of this paper is arranged as follows. Section 2 introduces the chaotic system and DNA coding; Section 3 describes the encryption algorithm; Section 4 shows the simulation experiment results; Section 5 presents the security analysis of the encryption method we proposed and the comparison results with the other three methods; finally, Section 6 summarizes this paper.

2. Chaotic System and DNA Coding

In this part, we first introduce the new 2D logistic-sine coupling map and new Henon map of the two chaotic systems involved in our method. The chaotic behavior of each chaotic map is analyzed by bifurcation diagram, Lyapunov exponent (LE) diagram, and trajectory diagram. In the bifurcation diagram, the horizontal axis represents parameters of a system and the vertical axis represents variables. If the system is chaotic, each fixed variable will correspond to as many parameters as possible. For two trajectories of a chaotic system, from two closed initial states, LE describes their average separation rate. Positive maximum LE number means that the closed trajectories of the dynamic system diverge in each unit time and evolve into completely different trajectories with the increase of time. Therefore, if the maximum LE number of the dynamic system is positive, the system is chaotic and a larger maximum LE number means better performance [6]. If a dynamic system can get more than one positive solution, its trajectory will diverge in many directions, so it has hyperchaotic behavior. The hyperchaotic behavior is a much more complex motion than the chaotic behavior. Trajectories indicate the motion from a given initial state increases with time. The trajectory of periodic motion is a closed curve, and the trajectory of chaotic behavior will never be closed or repeated in theory. Therefore, the chaotic trajectory usually occupies part of the phase space, which can reflect the randomness of the output of the chaotic system. If the chaotic track of chaotic system can occupy more phase space, it has better random output. Finally, we will explain the DNA coding method used in this paper.

2.1. Chaotic System. Many chaos-based digital image encryption schemes considering image features have been proposed.

In [1], in order to design a new chaotic map for image encryption algorithm with higher security, a two-dimensional Logistic-Sine coupling map, namely, 2D-LSCM, was proposed. The 2D-LSCM is defined as

\[
\begin{align*}
  x_{i+1} &= \sin(\pi(4\theta x_i (1 - x_i) + (1 - \theta)\sin(\pi y_i))), \\
  y_{i+1} &= \sin(\pi(4\theta y_i (1 - y_i) + (1 - \theta)\sin(\pi x_i))),
\end{align*}
\]

where \( \theta \) is the control parameter. It is generated by first coupling the logistic and sine maps, then extending the dimension from one-dimensional (1D) to 2D.

The system is chaotic when \( \theta \in (0, 1) \), \( x_i \), and \( y_i \) are between \([0, 1]\) certainly. The bifurcation diagram of 2D-LSCM is shown in Figures 1(a) and 1(b) when \( x_0 = 0.4 \) and \( y_0 = 0.1 \).

As can be seen from Figure 1, the 2D-LSCM has the problem of uneven data distribution. Therefore, we have...
improved the 2D-LSCM. The new definition of 2D-LSCM (namely N2D-LSCM) is as follows:

\[
\begin{align*}
    x_{i+1} &= \left(\sin(\pi (4\theta x_i(1 - x_i) + (1 - \theta)\sin(\pi y_i)))\right) \times 2^h \mod 1, \\
    y_{i+1} &= \left(\sin(\pi (4\theta y_i(1 - y_i) + (1 - \theta)\sin(\pi x_{i+1})))\right) \times 2^h \mod 1,
\end{align*}
\]

(2)

where \(\theta\) is the parameter and \(\theta \in (0, 1)\). The system is chaotic when \(x_i \in [0, 1]\), \(y_i \in [0, 1]\). The bifurcation diagram of N2D-LSCM is shown in Figures 2(a) and 2(b) when \(x_0 = 0.4\), \(y_0 = 0.1\), and \(h = 10\). Compared with 2D-LSCM, N2D-LSCM not only increases the values of two maximum Lyapunov exponents but also further increases the proportion of trajectories in phase space.

2.2. Improved Henon Map. Henon map is a two-dimensional discrete chaotic map, which is defined by

\[
\begin{align*}
    x_{i+1} &= 1 - ax_i^2 + y_i, \\
    y_{i+1} &= bx_i.
\end{align*}
\]

(3)

When \(a \in [1.1, 1.4], b = 0.3\), the Henon map is chaotic. The bifurcation diagram of Henon map is shown in Figures 5(a) and 5(b), and its Lyapunov exponent diagram is shown in Figure 6(a).

According to Figure 5, it can be found that Henon mapping not only has the problem of uneven data distribution but also the parameter range is small. In order to solve the above problems, we combine the Henon map and the sine map to construct a new two-dimensional map. In addition, we also combine the linear part \(z_i\) into nonlinear part \(e^{zi}\), a new Henon (namely, NHenon) map is obtained, which is defined as follows:

\[
\begin{align*}
    z_{i+1} &= \left(1 - a(\sin z_i)^2 + \sin q_i\right) \mod 1, \\
    q_{i+1} &= be^{zi} \mod 1.
\end{align*}
\]

(4)

The range of parameters \(a\) and \(b\) is extended to \((-\infty, +\infty)\).

When \(z_0 = 0.5\) and \(q_0 = 0.5\), \(a \in (-10, 10), b = 1.3\), the bifurcation diagram of NHenon is shown in Figures 7(a) and 7(b), and its Lyapunov exponent diagram is shown in Figure 6(b). For the values of the above parameters, Figures 8(a) and 8(b) are the trajectories corresponding to formulas (5) and (6).

By comparing the bifurcation diagrams of Henon map and NHenon map, we can see that NHenon map has more uniform data distribution than Henon map. From Figure 6(b), it can be found that \(\lambda_1 \in (0.05, 0.2)\) and \(\lambda_2 \in (0.35, 0.5)\); therefore, there are always two positive maximum Lyapunov exponents in NHenon, which have a better chaotic property. At the same time, NHenon’s trajectories occupy more phase space than Henon mapping.

In our proposed encryption method, the values of two chaotic sequences generated by N2D-LSCM are taken as the parameters of the NHenon mapping, so as to further improve the chaotic performance of the mapping. The form of the composite chaotic mapping is as follows:

\[
\begin{align*}
    x_{i+1} &= \left(\sin(\pi (4\theta x_i(1 - x_i) + (1 - \theta)\sin(\pi y_i)))\right) \times 2^h \mod 1, \\
    y_{i+1} &= \left(\sin(\pi (4\theta y_i(1 - y_i) + (1 - \theta)\sin(\pi x_{i+1})))\right) \times 2^h \mod 1, \\
    z_{i+1} &= \left(1 - hx_{i+1} \times (\sin z_i)^2 + \sin q_i\right) \mod 1, \\
    q_{i+1} &= hy_{i+1} \times e^{zi} \mod 1,
\end{align*}
\]

(5)

where \(\theta \in (0, 1)\) is the parameter, \(x_n \in [0, 1]\), \(y_n \in [0, 1]\), \(z_n \in [0, 1]\), and \(q_n \in [0, 1]\).

2.3. DNA Coding. There are four bases in the DNA molecule in total, respectively, A, G, C, and T, where A is
Figure 2: The bifurcation diagram of N2D-LSCM. (a) $\theta - x$; (b) $\theta - y$.

Figure 3: The Lyapunov exponents diagram of (a) 2D-LSCM and (b) N2D-LSCM.

Figure 4: Trajectories of (a) 2D-LSCM and (b) N2D-LSCM.
complementary to T and G is complementary to C. Applying this property to the binary, that is 0 and 1 complement each other. By this, we can complete the DNA encoding of the binary number. In an image, the value of pixels is represented as binary, and each two bits represent one base, such that 00 and 11, 01 and 10 are complementary pairs in binary form. The eight DNA coding rules in Table 1 satisfy the DNA complementation rules proposed by Watson and Crick [20]. For example, the pixel value is 78, whose binary form is 01001110, if encoded according to rule 1 of Table 1, and the result is CATG.

Binary operations include addition, subtraction, XOR, and many other operations. As researchers delve deeper into DNA computing, they propose addition, subtraction, and exclusive OR operations on DNA computation. In this paper, to simplify the algorithm further, only the XOR operation of DNA is used in the diffusion phase, which is represented by $\oplus$.

According to the eight DNA coding rules in Table 1, there are eight corresponding DNA exclusive OR rules, and the DNA exclusive OR operation rules based on DNA encoding rule 1 are shown in Table 2.

3. Encryption Algorithm Description

SHA256 algorithm can generate a unique hash value of 256 bits for any length of data. In our algorithm, the SHA 256 hash value of plaintext image is used to get the initial values and parameters of N2D-LSCM and NHenon. In the scrambling process, two pseudorandom sequences are generated by N2D-LSCM. On the one hand, these two chaotic sequences are used as a scrambling sequence and a
scrambling DNA coding matrix, respectively. Firstly, the scrambling sequence is used to scramble plaintext image at pixel level, and then scrambling results are encoded according to scrambling DNA coding matrix. On the other hand, in the diffusion stage, the sequence generated by N2D-LSCM is taken as the parameter of NHenon. And then, two chaotic sequences are obtained by iterating the NHenon mapping, one as the key matrix and the other as the corresponding DNA coding rule matrix. The key matrix is encoded by the DNA coding matrix, and finally the permuted DNA matrix and random DNA matrix are performed DNA exclusive OR operation completes image encryption. The flow chart of our encryption algorithm is shown in Figure 9.

| Rule | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|---|---|---|---|---|---|---|
| A    | 00 | 00 | 01 | 01 | 10 | 10 | 11 | 11 |
| T    | 11 | 11 | 10 | 10 | 01 | 01 | 00 | 00 |
| C    | 01 | 10 | 00 | 11 | 00 | 11 | 01 | 10 |
| G    | 10 | 01 | 11 | 00 | 11 | 00 | 10 | 01 |

**Table 1: DNA encoding rules.**

| XOR ⊕ | A | G | C | T |
|--------|---|---|---|---|
| A      | A | G | C | T |
| G      | G | A | T | C |
| C      | C | T | A | G |
| T      | T | C | G | A |

**Table 2: DNA XOR operation rule based on DNA coding rule 1.**
3.1. Generating Initial Values and Parameters of Chaotic Maps. First, the SHA256 hash value of the original image is set every 8 bits as a group, and a total of 32 sets are used as the encryption key K:

\[ K_n = k_{n,1}k_{n,2}k_{n,3}k_{n,4}k_{n,5}k_{n,6}k_{n,7}k_{n,8}, \]
\[ K = K_1K_2\ldots K_{32}. \]  

(6)

Then, according to formula (7), the intermediate variables used to calculate the initial value and parameters of chaotic mapping are obtained:

\[ t_i = \left( l_i \times \sum_{j=1}^{16} K_j \times \sum_{j=1}^{16} (2^j \times K_{2j-1}) \right) \mod 1, \]
\[ i = 1, 2, 3, 4, 5. \]  

(7)

Among them, \( l_1, l_2, l_3, l_4, \) and \( l_5 \) are the intermediate variables for the parameters and initial values of N2D-LSCM and NHenon, respectively. The parameters and initial values of (5) are obtained by substituting the intermediate variables into the following equations:

\[ \theta = \frac{t_i \mod 1}{2} + 0.5, \quad i = 1, 2, 3, 4, 5, \]
\[ h = \text{round} \left( \frac{1}{5} \sum_{i=1}^{5} L_i \right) + 10, \]  

(8)

where \( \theta = L_1, \quad x_0 = L_2, \quad y_0 = L_3, \quad z_0 = L_4, \) and \( q_0 = L_5. \) Round (·) is rounding function. \( L_i (i = 1, 2, 3, 4, 5) \) and \( h \) are the initial values and parameters of N2D-LSCM and NHenon.

3.2. Pixel Level Scrambling

Step 1: four chaotic sequences \( X, Y, Z, \) and \( Q \) are obtained by iterative system (5) for \( 1000 + M \times N \) times, and their first 1000 items are deleted. These four sequences is calculated as follows:

\[ X = \{ x_i \}, \quad i = 1, 2, \ldots, M \times N \],
\[ Y = \{ y_i \times 8 \}, \quad i = 1, 2, \ldots, M \times N \],
\[ Z = \{ z_i \times 10^{14} \} \mod 256 \}, \quad i = 1, 2, \ldots, M \times N \],
\[ Q = \{ q_i \times 8 \}, \quad i = 1, 2, \ldots, M \times N \],  

(9)

where the symbol \( i \) indicates rounding up; that is, the smallest integer greater than \( i \). mod indicates modulo, and \( M \) and \( N \) are the height of the image and the width of the image, respectively.

Step 2: the plaintext image of \( M \times N \) is represented by \( P1 \), which is converted into a one dimension array \( P2 = [P_{21}, P_{22}, \ldots, P_{2M\times N}] \).

Step 3 (scrambling stage): the specific process is as follows:

1. Order \( X \) increased to obtain the sequence \( X1 = \{ x_i \}, \quad i = 1, 2, \ldots, M \times N \}, \) and then the elements in \( X1 \) corresponds to the subscript in \( X \) to form a sequence \( X2 \), so that it satisfies the requirements of \( x_\theta = x_{\theta}, \quad i = 1, 2, \ldots, M \times N \).
2. Rearrange \( P2 \) as \( P3 \), so that it satisfies \( P_{3i} = P_{2\theta_i}, \quad i = 1, 2, \ldots, M \times N \).
3. Reshape \( P3 \) into \( M \times N \) matrix and define it as \( P4 \). \( P5 \) is a \( M \times 8N \) matrix that expands the value of pixels in \( P4 \) into 8-bit binary.
4. Carry out DNA encoding for \( P5 \). The 8-bit binary of each pixel adopts the same coding rule. That is, each pixel corresponds to a DNA coding rule. Determine the DNA coding rule corresponding to each pixel according to the value of sequence \( Y \) and carry out DNA encoding for it. Obtain the DNA matrix \( P6 \) of \( M \times 4N \) after scrambling and complete the scrambling.

Figure 9: Encryption algorithm flow chart.
3.3. Diffusion Stage

Step 1: the matrix $P_6$ is obtained by rearranging the sequence $Z$ into the matrix of $M \times N$.
Step 2: each pixel of matrix $Z$ is expanded into 8-bit binary to get matrix $Z_2$.
Step 3: each pixel of $Z_1$ adopts the same coding rule; that is, each pixel corresponds to a DNA coding rule. According to the elements’ value of sequence $Q$, the corresponding DNA coding rule of each pixel is determined, and the DNA coding of $Z_2$ is carried out to obtain the DNA random matrix of $M \times 4N$.
Step 4: assume that the DNA coding matrix after diffusion is $C_{DNA}$. Then, we use the diffusion algorithm proposed in the reference, and the specific process is described as follows:

(1) First, calculate the center point of matrix $P_6$ according to equation (11) and then diffuse the center point of $P_6$ according to equation (12)id2:

$$
\begin{align*}
   cr &= \frac{M}{2} \\
   cc &= \frac{4N}{2} \\
   C_{DNA_{cr,cc}} &= P_6^{cr,cc} \oplus \text{DNA}_{\text{Matrix}_{cr,cc}}
\end{align*}
$$

where $\oplus$ is the exclusive OR operator.

(2) Starting from the center point of $P_6$, diffuse the $cr$-th row of the matrix $P_6$ toward the left and right at the same time. Diffusion in the left direction is as follows:

$$
\begin{align*}
   C_{DNA_{cr,cc-1}} &= P_6^{cr,cc-1} \oplus \text{DNA}_{\text{Matrix}_{cr,cc-1}} \oplus C_{DNA_{cr,cc-1}}, \\
   i &= 1, 2, \ldots, cc - 1.
\end{align*}
$$

(11)

Diffusion in the right direction is as follows:

$$
\begin{align*}
   C_{DNA_{cr,cc+1}} &= P_6^{cr,cc+1} \oplus \text{DNA}_{\text{Matrix}_{cr,cc+1}} \oplus C_{DNA_{cr,cc+1}}, \\
   i &= 1, 2, \ldots, cc - 1.
\end{align*}
$$

(12)

(3) Start from the middle row of $P_6$, and simultaneously spread in the upper and lower directions to accelerate diffusion.

   Diffusion in the upper direction is as follows:

$$
\begin{align*}
   \text{up} &= cr - i.
\end{align*}
$$

(13)

If $\text{up} > 0$:

$$
\begin{align*}
   C_{DNA_{up,j}} &= P_6^{up,j} \oplus \text{DNA}_{\text{Matrix}_{up,j}} \oplus C_{DNA_{up-1,j}},
\end{align*}
$$

where $j = 1, 2, \ldots, 4N$.

If $\text{down} \leq M$:

$$
\begin{align*}
   C_{DNA_{\text{down},j}} &= P_6^{\text{down},j} \oplus \text{DNA}_{\text{Matrix}_{\text{down},j}} \oplus C_{DNA_{\text{down}-1,j}},
\end{align*}
$$

(16)

where $i = 1, 2, \ldots, (M - 1)/2, j = 1, 2, \ldots, 4N$.

3.4. Getting the Encrypted Image. After the diffusion process at the DNA level, the $C_{DNA}$ needs to be decoded to get the encrypted image. The decoding steps are as follows:

Step 1: the matrix $C_{DNA}$ is randomly decoded into binary form. Let the array of DNA decoding rules be $\text{Rules} = \{\text{rule}_i = (Y_i + Q_i) \mod 8 + 1 | i = 1, 2, \ldots, M \times N\}$, decode the four DNA codes corresponding to each pixel according to Rules, and obtain the binary matrix $C_{bin}$ of $M \times 8N$ after DNA decoding.
Step 2: every 8 bits of binary matrix $C_{bin}$ are restored to decimal numeral system, and the encrypted image $C_{\text{img}}$ is obtained.

4. Experimental Results

Run PyCharm software in Windows 10 64-bit system environment and use the algorithm proposed in this paper to encrypt and decrypt Baboon.pgm (256 x 256), House.pgm (256 x 256), Barche.pgm (512 x 512), and Peppers.pgm (512 x 512) grayscale image. The selected key is shown in Table 3. The experimental results are shown in Figure 10.

By comparing the decrypted images in the experiment with the corresponding original images, the result shows that all the pixels are the same.

5. Security Analysis

5.1. Key Space. In order to resist brute attacks effectively, the key space should be large enough. The initial key values of our proposed method are $l_1, l_2, l_3$ and $l_4$, and the calculation accuracy is about $10^{-14}$, so our key space is $10^{70}$, which is far larger than $2^{100}$. That means we can effectively resist brute attacks [21].

5.2. Key Sensitivity. A secure encryption scheme has a very high sensitivity to the key. Even if the encryption key has a tiny change, the encryption/decryption result should be completely different. In our test, we add $0.1234 \times 10^{-14}$ to the key $s_{l_1}$ in Table 3, and keep the rest of the key $l_1$ unchanged to get key $2$. Encrypt the Peppers.pgm with key 1 and key 2, respectively. The encryption results are shown in Figures 11(a) and 11(b). Here, Figure 11(c) shows the difference between Figures 11(a) and 11(b). Key 2 is used as the key to decrypt the encryption graph corresponding to key 1, and the decryption result is shown in Figure 11(d). The experimental results show that our method has good key sensitivity in the encryption and decryption processes.
5.3. Differential Attack Analysis. Differential attack is a very common and effective analysis method in image analysis. It modifies the plaintext image slightly and then encrypts the original plaintext image and the modified plaintext image, respectively. By analyzing the difference between the plaintext image and the encrypted image, attackers try to get the relationship between the plaintext image and the encrypted image. In order to resist differential attack effectively, encryption algorithm should have excellent sensitivity to plaintext image. Generally, we measure it according to NPCR and UACI. NPCR represents the pixel change rate when modifying one pixel, and UACI represents the average intensity of the difference between plaintext image and ciphertext image. For 8-bit grayscale images, the expected values of NPCR and UACI are 99.6094% and 33.4635%, respectively. The formulas are as follows:

\[
\text{NPCR} = \frac{1}{M \times N} \left( \sum_{i,j} D(i, j) \right) \times 100%,
\]

\[
D(i, j) = \begin{cases} 
0, & C(i, j) = C'(i, j), \\
1, & C(i, j) \neq C'(i, j), 
\end{cases}
\]

\[
\text{UACI} = \frac{1}{M \times N} \left( \sum_{i,j} \frac{|C(i, j) - C'(i, j)|}{255} \right) \times 100%,
\]

where \(C\) is the encrypted image of the unmodified plaintext image, \(C'\) is the encrypted image of the modified plaintext image, \(M\) is the height of the plaintext image, \(N\) is the width of the plaintext image, and \(C(i, j)\) is the pixel value at the coordinate \((i, j)\) of the encrypted image \(C\). The keys used by

\begin{table}[h]
\centering
\caption{Encryption key we used.}
\begin{tabular}{|c|c|c|c|c|}
\hline
Key & \(l_1\) & \(l_2\) & \(l_3\) & \(l_4\) & \(l_5\) \\
\hline
key_1 & 0.5721 & 0.3138 & 0.7564 & 0.5492 & 0.8589 \\
\hline
\end{tabular}
\end{table}

Figure 10: Encryption and decryption experiment result (from top to bottom: original image, encrypted image, and decrypted image, respectively).
our encryption method are obtained by the user key and the SHA256 hash of the plaintext image. The encryption result is highly related to the plaintext, which can resist the differential attack effectively. The experimental results of differential analysis with other methods in the comparative literature are shown in Table 4.

5.4. Information Entropy. Shannon entropy analysis has been widely used to evaluate the uncertainty or unpredictability. The formula is as follows:

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i) \log \frac{1}{p(m_i)}.$$  \hspace{1cm} (18)

Here, $N$ is the total number of samples (when the image is with a bit-depth of 8, $N$ is 8), $m$ is the information source, and $p(m_i)$ represents the probability of sample $m_i$. For the image with gray level 8, according to the principle of maximum entropy, the information entropy is closer to 8, and it means that the probability of each pixel is more average. Using the methods in this paper and references [9, 15, 18] to analyze the information entropy of Baboon, House, Lena, Peppers, and their corresponding encrypted images, the comparative experimental results are shown in Table 5.

5.5. Histogram Analysis. A secure image encryption scheme should be able to ensure that the pixel values of the ciphertext image are evenly distributed and enhance the

| Images  | Size      | Measure | Ref. [18] | Ref. [9] | Ref. [15] | Our study |
|---------|-----------|---------|-----------|---------|-----------|-----------|
| Baboon  | 256 * 256 | NPCR    | 0.99603   | 0.99574 | 1.00000   | 0.99607   |
|         |           | UACI    | 0.33479   | 0.33492 | 0.33143   | 0.33362   |
|         | 512 * 512 | NPCR    | 0.99612   | 0.99613 | 1.00000   | 0.99596   |
|         |           | UACI    | 0.33477   | 0.33408 | 0.35162   | 0.33459   |
| House   | 256 * 256 | NPCR    | 0.99613   | 0.99568 | 1.00000   | 0.99609   |
|         |           | UACI    | 0.33534   | 0.33596 | 0.33100   | 0.33492   |
|         | 512 * 512 | NPCR    | 0.99582   | 0.99615 | 1.00000   | 0.99668   |
|         |           | UACI    | 0.33460   | 0.33279 | 0.24614   | 0.33486   |
| Barche  | 256 * 256 | NPCR    | 0.99633   | 0.99618 | 1.00000   | 0.99611   |
|         |           | UACI    | 0.33402   | 0.33445 | 0.24038   | 0.33514   |

Figure 11: Key sensitivity analysis results. (a) Encryption result of key1; (b) encryption result of key2; (c) difference of (a) and (b); (d) decryption result of using key2 to decrypt (a).
resistance to statistical attacks, so as to prevent attackers from obtaining any meaningful data from the ciphertext image. We tested Baboon.pgm, House.pgm, Barche.pgm, Peppers.pgm, and their corresponding ciphertext images with different keys. The histogram of the original images is shown in Figure 12.

In the mathematical quantity analysis of each key, we judge the uniformity by the variance of the histogram of the encrypted image. Generally, if the variance of the encrypted image is lower, the uniformity is better. We encrypt the same image using key1 and different passwords obtained by changing only one parameter in key1 at a time and then calculate the variance of each encrypted image. In the case of different keys, the closer the two variance values the higher the uniformity. And, the results are shown in Table 6. As can be seen from Table 6, the average variance is about 950 when using key1, which is smaller than 1017 in reference [15]. The variance of the histogram is defined as follows:

$$\text{var}(C) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (C_i - C_j)^2,$$

(19)

where C is the vector of histogram values and $C = \{C_1, C_2, \ldots, C_{256}\}$ and $C_i$ and $C_j$ represent the number of pixel value equal to i and j, respectively.

By calculating the differences of variance between key1 and the secret key obtained by changing only one parameter in key1, we can measure the uniformity of each parameter in secret keys. The result is shown in Table 7. As can be seen from Table 7, the differences of all variances are around 100, so each parameter in the secret key has good uniformity.

Through the above analysis, it can be seen that the pixel values of the encrypted image obtained by this encryption method are evenly distributed and can effectively resist statistical attacks.

5.6. Correlation Analysis. Because of the high correlation of adjacent pixels in the image, in order to resist the statistical attack, the correlation between adjacent pixels should be reduced as much as possible. The correlation coefficient is used to describe the degree of correlation between pixels. The smaller the correlation coefficient is, the lower the correlation between adjacent pixels is. The calculation method is as follows:

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

(20)

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

where x and y are grey-level values of the two adjacent pixels in the image, and the expectation and variance of x are $E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i$ and $D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2$, respectively.

Five-thousand pairs of adjacent pixels are randomly selected from the horizontal direction, vertical direction, and diagonal direction of the original image and the encrypted image, and the correlation coefficient is calculated. The experimental results are shown in Table 8 and Figure 13. Compared with other recently proposed methods, the results are shown in Table 9. It can be seen that the correlation between adjacent pixels of plaintext image is very strong, but the correlation coefficient of encrypted image is close to 0, which proves that the correlation between adjacent pixels is very low.

5.7. Resistance to Cropping Attacks. Because some information loss may occur when data are transmitted in the channel, a good encryption algorithm should recover data as much as possible in the case of information loss. It can be measured by MSE and NSPR, which are defined as follows:

$$\text{MSE} = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} [P(x,y) - D(x,y)]^2,$$

(21)

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \text{dB},$$

where M and N are the height and width of image P, respectively, $P(x,y)$ represents the pixel value of the original image at coordinates (x, y), and D is the decrypted image of P after losing some data.

By testing, we can see that the cipher images can be decrypted with some data loss, which means that proposed algorithm has a good robustness as for data loss. The experimental results are shown in Figure 14 and Table 10.

| Images  | Size       | Original image | Information entropy of cryptograph | Information entropy of cryptograph |
|---------|------------|----------------|-----------------------------------|-----------------------------------|
| Baboon  | 256 * 256  | 7.33848        | 7.99720                           | 7.99719                           | 7.99710                           |
|         | 512 * 512  | 7.35794        | 7.99745                           | 7.99933                           | 7.99923                           |
| House   | 256 * 256  | 6.49627        | 7.99751                           | 7.99744                           | 7.99740                           |
| Barche  | 256 * 256  | 7.56829        | 7.99726                           | 7.99719                           | 7.99754                           |
|         | 512 * 512  | 7.44557        | 7.99923                           | 7.99940                           | 7.99935                           |
| Peppers | 256 * 256  | 7.52513        | 7.99744                           | 7.99700                           | 7.99707                           |
|         | 512 * 512  | 7.57148        | 7.99930                           | 7.99924                           | 7.99929                           |
Table 6: Variances of histograms compared among all secret keys.

| Ciphered image | key1 | l₁ | l₂ | l₃ | l₄ | l₅ |
|----------------|------|----|----|----|----|----|
| Baboon         | 1114.0390 | 1033.3203 | 1158.7187 | 1110.1250 | 1030.6718 | 1064.8125 |
| Barche         | 980.2109 | 928.9921 | 1114.4921 | 1024.2187 | 989.1640 | 918.5390 |
| Peppers        | 756.8437 | 842.0703 | 961.9609 | 947.9531 | 863.9609 | 926.7109 |
| Average        | 950.3645 | 934.7942 | 1078.3905 | 1027.4320 | 961.2656 | 970.0208 |

Table 7: Variances difference of histograms compared among all secret keys.

| Ciphered image | l₁ | l₂ | l₃ | l₄ | l₅ |
|----------------|----|----|----|----|----|
| Baboon         | 80.7187 | 44.6797 | 3.914 | 83.3672 | 49.2265 |
| Barche         | 51.2188 | 134.2812 | 44.0078 | 8.9531 | 61.68 |
| Peppers        | 85.2266 | 205.1172 | 191.1094 | 107.1172 | 169.8672 |
| Average        | 72.3880 | 128.0260 | 79.6770 | 66.4791 | 93.5912 |

Table 8: Correlation coefficient of images.

| Image          | Horizontal | Vertical | Diagonal |
|----------------|------------|----------|----------|
| Baboon         | 0.6578     | 0.7301   | 0.6387   |
| Cipher baboon  | -0.0022    | -0.0064  | -0.0023  |
| House          | 0.9528     | 0.9769   | 0.9331   |
| Cipher house   | -0.0008    | 0.0033   | 0.0007   |
| Barche         | 0.9759     | 0.9605   | 0.9447   |
| Cipher barche  | 0.0060     | 0.0021   | -0.0061  |
| Peppers        | 0.7471     | 0.8655   | 0.7172   |
| Cipher peppers | -0.0020    | -0.0046  | -0.0070  |
**Figure 13:** The adjacent-pixel distribution maps of the input images and the cipher images. (a) Baboon. (b) House. (c) Barche. (d) Peppers.

**Table 9:** Correlation coefficient of house with different algorithms.

| Direction | Plain image | Ref. [18] | Ref. [9] | Ref. [15] | Ours   |
|-----------|-------------|-----------|----------|-----------|--------|
| Horizontal| 0.9528      | 0.0060    | 0.0043   | -0.0012   | -0.0008|
| Vertical  | 0.9769      | 0.0038    | 0.0034   | -0.0008   | 0.0033 |
| Diagonal  | 0.9331      | -0.0013   | -0.0070  | -0.0060   | 0.0007 |

**Figure 14:** Cropping attack analysis. (a) Missing 12.5% of data; (b) missing 25% of data; (c) missing 50% of data.
| Attacks               | Degree            | Ref. [18]  | Ref. [9]  | Ref. [15]  | Our study |
|----------------------|-------------------|------------|------------|------------|-----------|
| **Gaussian noise**   | Variance is 0.0001 | 35.1889    | 27.9122    | 32.1278    | 35.6820   |
|                      | Variance is 0.0003 | 33.2182    | 27.8909    | 32.1362    | 35.6493   |
|                      | Variance is 0.0005 | 32.3709    | 27.8935    | 32.1352    | 35.6915   |
| **Salt and pepper noise** | Density is 0.01   | 47.7232    | 27.9196    | 39.0378    | 42.3384   |
|                      | Density is 0.05   | 40.6816    | 27.9184    | 32.5863    | 34.9672   |
|                      | Density is 0.25   | 33.7355    | 27.8949    | 28.2967    | 29.1269   |
| **Cropping**         | 12.5%             | 36.7361    | 27.8948    | 36.5865    | 37.0017   |
|                      | 25%               | 33.7410    | 27.9166    | 33.6903    | 33.9202   |
|                      | 50%               | 30.7403    | 27.8844    | 30.7129    | 30.8924   |

Figure 15: Gaussian noise (a)–(c) and salt and pepper noise (d)–(f) results. (a) Variance is 0.0001; (b) variance is 0.0003; (c) variance is 0.0005; (d) density is 0.01; (e) density is 0.05; (f) density is 0.25.

Figure 16: Continued.
5.8. Resistance to Noise Attacks. When image data are transmitted in the channel, it may be affected by noise, and the most common ones are Gaussian noise and salt and pepper noise. A good encryption system should be able to recover the encrypted image affected by noise to the maximum extent. By simulating different intensities of Gaussian noise and salt and pepper noise, the corresponding decrypted image is obtained. The experimental results are shown in Figure 15. From testing results, we can see that the cipher images can be decrypted with some noise.

5.9. Resistance to Plaintext Attacks. The classical attack types include ciphertext only, known plaintext, chosen plaintext, and chosen ciphertext, and chosen plaintext is the most powerful attack. The proposed algorithm is sensitive to the initial parameters and values. Moreover, the encryption result of every byte is influenced by the contents of the previous bytes. In general, if the white and black plaintext images can be encrypted, and its histogram with random distribution can effectively resist plaintext attack and selective plaintext attack. The experimental results are shown in Figure 16, and the NPCR and UACI results of the encrypted image are shown in Table 11. It can be seen that the pixels of the encrypted image obey the random distribution.

6. Conclusions

In this paper, we improve the 2D-LSCM and Henon mapping to make the data more uniform and have larger Lyapunov exponents and then propose our image encryption algorithm. In this algorithm, the SHA256 hash value of plaintext image is used to get the initial values and parameters of N2D-LSCM and NHenon. In the scrambling process, two pseudorandom sequences are generated by N2D-LSCM.

These two chaotic sequences are used as scrambling sequences and scrambling DNA coding matrix respectively. Firstly, scrambling sequence is used to scrambling plaintext image at pixel level, and then scrambling results are encoded according to scrambling DNA coding matrix. In addition, in the diffusion stage, firstly, the sequence values generated by N2D-LSCM are taken as the parameters of NHenon, and then two chaotic sequences are obtained by iterating the NHenon mapping, one is as the key matrix, and the other is as the corresponding DNA coding rule matrix. The chaotic key matrix is encoded by the DNA coding matrix. Finally, the permuted DNA matrix and random DNA matrix performed DNA exclusive OR operation to complete image encryption. The simulation shows that this encryption method has good security and can effectively resist many common attacks, such as statistical attack, violent attack, select/known plaintext attack, cropping attack, and noise attack. The scrambling method proposed in this paper is not efficient enough in operation. In the next step, we will improve the scrambling efficiency in this algorithm and develop an encryption algorithm suitable for color image.

Data Availability

The data used to support the findings of this study are included within the article.
Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Science Foundation of China under the Grant no. 61876201.

References

[1] Z. Hua, F. Jin, B. Xu, and H. Huang, “2D Logistic-Sine-coupling map for image encryption,” *Signal Processing*, vol. 149, pp. 148–161, 2018.

[2] National Technical Information Service, *FIPS PUB 46: Data Encryption Standard (DES)*, National Technical Information Service, Springfield, VA, USA, 1999.

[3] National Technical Information Service, *FIPS PUB 197: Advanced Encryption Standard (AES)*, National Technical Information Service, Springfield, VA, USA, 2001.

[4] A. Kanso and M. Ghebleh, “A novel image encryption algorithm based on a 3D chaotic map,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 7, pp. 2943–2959, 2012.

[5] T. Gao and Z. Chen, “A new image encryption algorithm based on hyper-chaos,” *Physics Letters A*, vol. 372, no. 4, pp. 394–400, 2008.

[6] C. Pak, K. An, P. Jang, J. Kim, and S. Kim, “A novel bit-level color image encryption using improved 1D chaotic map,” *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 12027–12042, 2018.

[7] L. Zhu, H. Song, X. Zhang, M. Yan, L. Zhang, and T. Yan, “A novel image encryption scheme based on nonuniform sampling in block compressive sensing,” *IEEE Access*, vol. 7, pp. 22161–22174, 2019.

[8] G. Zhang and Q. Liu, “A novel image encryption method based on total shuffling scheme,” *Optics Communications*, vol. 284, no. 12, pp. 2775–2780, 2011.

[9] C. Cao, K. Sun, and W. Liu, “A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map,” *Signal Processing*, vol. 143, pp. 122–133, 2018.

[10] Y. P. Zhang, F. Zuo, and Z. J. Qu, “A survey of digital image encryption based on Chaos,” *Computer Engineering & Design*, vol. 32, no. 2, pp. 463–466, 2011, (In Chinese).

[11] L. F. Liu and S. X. Miao, “A new simple one-dimensional chaotic map and its application for image encryption,” *Multimedia Tools & Applications*, vol. 77, no. 1, pp. 1–18, 2018.

[12] K. A. Kumar 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.

[13] Z. Hua, Y. Zhou, and H. Huang, “Cosine-transform-based chaotic system for image encryption,” *Information Sciences*, vol. 480, pp. 403–419, 2019.

[14] J. Wu, X. Liao, and B. Yang, “Image encryption using 2D Hénon-Sine map and DNA approach,” *Signal Processing*, vol. 153, pp. 11–23, 2018.

[15] R. L. Abdelfatah, “A new fast double-chaotic based Image encryption scheme,” *Multimedia Tools & Applications*, vol. 79, no. 1-2, pp. 1241–1259, 2019.

[16] X.-Y. Wang, Y.-Q. Zhang, and X.-M. Bao, “A novel chaotic image encryption scheme using DNA sequence operations,” *Optics and Lasers in Engineering*, vol. 73, pp. 53–61, 2015.

[17] A. Y. Niyat and M. H. Moattar, “Color image encryption based on hybrid chaotic system and DNA sequences,” *Multimedia Tools and Applications*, vol. 79, no. 1, pp. 1497–1518, 2020.

[18] J. C. Dagadu, J.-P. Li, and P. C. Addo, “An image cryptosystem based on pseudorandomly enhanced chaotic DNA and random permutation,” *Multimedia Tools & Applications*, vol. 78, no. 17, pp. 24979–25000, 2019.

[19] H. R. Amani and M. Yaghoobi, “A new approach in adaptive encryption algorithm for color images based on DNA sequence operation and hyper-chaotic system,” *Multimedia Tools and Applications*, vol. 78, no. 15, pp. 21537–21556, 2019.

[20] J. D. Watson and F. H. C. Crick, “Molecular structure of nucleic acids,” *American Journal of Psychiatry*, vol. 160, no. 4, pp. 623–624, 2003.

[21] G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.