Chaotic image encryption algorithm based on bit-plane and DNA multiplex operation

Yun tian Zhang
Heilongjiang University

Bing Zhao (✉ zhaobing@hlju.edu.cn)
Heilongjiang University

Research Article

Keywords: Image encryption, bit layer, information security, DNA encoding, anti-attack

Posted Date: October 31st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2197914/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
Chaotic image encryption algorithm based on bit-plane and DNA multiplex operation

Yun Tian Zhang1 Department of Electronic Engineering, Heilongjiang University, Harbin, 150080, China, Email: Zzyty@163.com
Bing Zhao* Department of Electronic Engineering, Heilongjiang University, Harbin, 150080, China, ORCID: 0000-0001-6476-8201 Email: zhaobing@hlju.edu.cn

Abstract: A novel bit-plane chaotic image encryption algorithm is proposed further to improve image encryption systems' security and anti-attack ability. This paper combines bit layer scrambling and Deoxyribonucleic Acid (DNA) encoding. In this paper, we design a novel bit-plane scrambling method using parity alignment and plaintext correlation chaotic sequences, which can significantly reduce the correlation between image pixels and the correlation between RGB layers. Then the intermediate ciphertext is DNA encoded by hyper-chaotic sequences with plaintext correlation, after the designed multiple operation diffusion, the ultimate ciphertext is generated. Furthermore, devised a new way of generating keys, which are highly correlated with plaintexts. This increases the algorithm's resistance to selective plaintext attacks. According to the simulation results, this paper's proposed image encryption algorithm can effectively resist differential and statistical attacks, has ample key space and sensitivity, and has ideal information entropy.

Keywords: Image encryption; bit layer; information security; DNA encoding; anti-attack

1 Introduction

With the rapid progression of information civilization and the emergence of the epidemic, internet communication has become an essential way of life. However, communication security is facing the toughest test of all time; Digital image has intuition and hard expressiveness. Therefore image encryption algorithm design is increasing drastically. Previously, researchers have proposed various encryption algorithms for data security issues, for instance, Data Encryption Standards (DES)[1], and Advanced Encryption Standards (AES)[2]. Such group encryption algorithms and later emerged sequence ciphers such as ZUC [3] algorithm have blossomed in the field of data encryption. On the other hand, traditional data encryption algorithms are no longer applicable to image encryption due to characteristics such as color image redundancy. It is due to chaos having certainty, nonlinearity, and sensitivity to initial conditions. Simultaneously, it has numerous similarities with cryptology [4-6], thus widely used in encryption[7]. Ref. [8] proposed a coupling chaotic encryption algorithm. Its generation of keys is associated with plaintext and can also resist common attacks. However, it has an independent operation of scrambling and diffusion, and its key space is relatively small in comparison, so safety needs improvement. Ref. [9] proposed an algorithm that three chaotic hybrid systems are constructed by the difference of several one-dimensional chaotic systems, which do not have blank areas, have good performance, and are very appropriate for image encryption. The newly constructed chaotic system uses the encryption method to complete the steps of dislocation, diffusion, linear transformation and so on. This algorithm's security analysis reveals that it is effective and has specific anti-attack capabilities. Still, the security needs to be improved because it only uses the traditional encryption method of low-dimensional chaotic mapping and single diffusion.

In addition, the emerging DNA molecular encryption technique has high parallelism, low energy consumption, and high storage. Thus, it has been widely adopted by researchers. Ref. [10] proposed an approach based on DNA, Lorenz-Rossler chaotic system, and a 2D logistic map. This approach is similar to traditional DNA, combining chaotic systems with DNA sequences and bit-plane. A bitwise chaotic ponytail is used to transform bit streams from DNA sequences. This encryption algorithm's key is associated with the plaintext with high accuracy, and this method's encryption effect and security achieve good results. Ref. [11] proposed a late-model encryption algorithm. This method has ample secret key space because it is based on dynamic DNA coding and a chaotic system. Secondly, its plaintext sensitivity is robust. The new 3D projection scrambling method proposed in this algorithm significantly reduces the correlation between RGB layers. However, the
scrambling is not considered at the pixel level, so the indicators of the ciphertext still need to be improved. Ref. [12] proposed a Fisher-Yatess scrambling and DNA coding encryption algorithm. Although it has an excellent effect on encryption, the chaotic sequence used in this method of encryption is a one-dimensional chaotic sequence, which is highly efficient but of low complexity and therefore brings about less effective dislocation and diffusion. In addition, many scientists have combined DNA computing with chaotic systems. Moreover, have come up with many excellent algorithms [13-19].

The encryption algorithm, on account of bit-plane, has numerous advantages. It can operate the scrambling and diffusion simultaneously, which increases security performance. Ref. [20] proposed an algorithm. This algorithm decomposes the image into eight bit-plane matrices and converts them into 3D matrices of size $8 \times M \times N$. The 3D matrices are rotated in multiple directions according to the chaotic sequence generated from the plaintext information, thus disrupting the positions of the bit-plane pixels; in addition, the algorithm performs diffusion according to the chaotic system and achieves good encryption results. However, the algorithm encrypts all bit planes, which is slightly wasteful of resources given the vast differences in the information contained in different bit planes. Ref. [21] proposed a method that uses the hyper-chaotic and bit permutation, enhanced pseudo randomness of the key stream and cipher image. However, this algorithm also permutes all bit planes of the image. It converts all pixel points to one dimension, iterating the complex hyper-chaotic system many times, which is very computationally intensive if this method is used to encrypt color images. Furthermore, Ref. [22] proposed a cryptographic algorithm based on pixel permutation and bit-replace for a color image. Firstly, the encryption process is linked to plaintext, so the key can vary with the image. Secondly, this chaotic mapping is low dimensional, with an inadequate complexity; thus, its histogram and scatter diagram of ciphertext are not evenly distributed, and its key space is relatively small. Bit-plane [23-27] based research is becoming more and more widely noticed by researchers because of its excellent properties.

Hence, on account of the references’ issue, this paper proposed a color image encryption scheme that combined the bit-permutation and DNA coding. The present algorithm is optimized for the problems of low security and poor-quality cipher images. The proposed algorithm extracts the first three-bit planes of the RGB triplet of the plaintext image for parity arrangement and performs rank disorder using chaotic sequences because these three-bit planes occupy most of the information of the image pixels, and then performs pixel-level RGB disorder; then performs dynamic DNA encoding and adds all the three DNA matrices generated by the hyper-chaotic system into the DNA operation, because only one-third of the bit planes occupying most of the information and the parallelism of the DNA operation is chosen, the algorithm achieves good encryption effect with low operational complexity and also has better resistance to attack.

2 Basic theory
2.1 Chaotic system
2.1.1 Sine-cos system

Because the classic Sine system’s key space is small and the stability region being narrow with a blank area, [6] proposed a modified new chaotic system Sine-cos, which has no white space. Its Lyapunov exponent is 6.48. In addition, the parameter scale of this system is more expansive than Sine and traditional one-dimensional systems. Its system model can be expressed by Eq. (1).

$$x_{n+1} = \cos(\pi x_n \sin(\mu e^{0.5} \pi / 2) / 2) \tag{1}$$

Where $\mu$ represents the control parameters of this system, and $x$ is the state variable, while $\mu$ is on a scale of 1 to 4. In this paper, we take $\mu$ as 3.9999, $x$ is between 0 and 1. The system is in the entire mapping state and thus has benign chaos characteristics. The distribution of the ordinary sine mapping and the improved chaotic mapping is shown in Figure 1. From this, we can see that the improved chaotic mapping has good performance and high complexity and is suitable for image encryption.
2.1.2 Four-dimensional hyper-chaotic Chen system

Hyper-chaotic orbits have more complex stretching and folding properties than ordinary orbits, increasing encryption security. The use of higher-order chaotic systems, especially hyper-chaotic systems, can essentially enhance the pseudorandom and other properties of pseudorandom sequences, thus improving the overall security of the algorithm. The four-dimensional system can be expressed by Eq. (2).

\[
\begin{align*}
\dot{x} &= a(y-x) + w \\
\dot{y} &= -xz + dx + cy \\
\dot{z} &= xy - bz \\
\dot{h} &= yz + kw
\end{align*}
\]

(2)

Where \(a, b, c, d\) represent the control parameters of hyper-chaotic system; and \(x, y, z, h\) are state variable. When \(a = 35, b = 3, c = 12, d = 7\), and the \(k\) is on a scale of -0.7 to 0.7, the system is in a state of hyper-chaos. Take \(k = 0.6\), the differential equations of the chaotic system are solved using MATLAB function \texttt{ode45}, and some 2D and 3D attractor phase diagrams is shown in Figure 2.
2.2 DNA technique

2.2.1 Encoding and decoding of DNA

DNA cryptography has large-scale parallelism, massive storage capacity, and ultra-low power consumption. The nitrogen-containing bases of DNA molecules include four kinds, namely A(adenine), C(cytosine), G(guanine), and T(thymine). Because 0 and 1 in binary are complementary, T is complemented by A, and C is complemented by G. They are used in 24 coding schemes. Still, only eight comply with the complementary rules. Suppose A, T, C, G represent the binary numbers 00, 11, 01, 10, so the binary form of the pixel value 230 is (11100110)_2, and its base code is A, G, C, G. Table 1 shows the regulations for encoding and decoding.

<table>
<thead>
<tr>
<th>Regulations</th>
<th>00</th>
<th>01</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>C</td>
<td>G</td>
<td>T</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>A</td>
<td>T</td>
<td>G</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>G</td>
</tr>
<tr>
<td>5</td>
<td>G</td>
<td>A</td>
<td>T</td>
<td>C</td>
</tr>
<tr>
<td>6</td>
<td>G</td>
<td>T</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>7</td>
<td>T</td>
<td>C</td>
<td>G</td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>T</td>
<td>G</td>
<td>C</td>
<td>A</td>
</tr>
</tbody>
</table>

2.2.2 The DNA operation

DNA addition, subtraction and XOR are all examples of DNA operations. They all use the standard binary calculation. Table 2 shows the addition, subtraction and XOR regulations.

<table>
<thead>
<tr>
<th>+</th>
<th>A</th>
<th>T</th>
<th>C</th>
<th>G</th>
<th>-</th>
<th>A</th>
<th>T</th>
<th>C</th>
<th>G</th>
<th>⊕</th>
<th>A</th>
<th>T</th>
<th>C</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>T</td>
<td>C</td>
<td>G</td>
<td>A</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>A</td>
<td>T</td>
<td>C</td>
<td>G</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>C</td>
<td>G</td>
<td>A</td>
<td>T</td>
<td>T</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>T</td>
<td>T</td>
<td>A</td>
<td>G</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>G</td>
<td>A</td>
<td>T</td>
<td>C</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>G</td>
<td>C</td>
<td>C</td>
<td>G</td>
<td>A</td>
<td>T</td>
</tr>
<tr>
<td>G</td>
<td>G</td>
<td>A</td>
<td>T</td>
<td>C</td>
<td>G</td>
<td>G</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>G</td>
<td>G</td>
<td>C</td>
<td>T</td>
<td>A</td>
</tr>
</tbody>
</table>

2.3 Image bit layering
Each pixel value in an image with a grey level of 256 is an integer value between 0 and 255, meaning an 8-bit binary number can represent each pixel value. The i-th bit of the eight-bit binary number of each pixel represents the i-th layer of the image for that pixel, and Figure 3 depicts the image's bit-plane decomposition. From the image, we can know that the amount of information represented by each layer of the image is different. The low-level images represent some detailed information about the image, and the eight layers of images are superimposed to form the original image. From [26], the first three bits of the digital image, i.e., bits 8 to 6, contain nearly 90 percent of the pixel information, while the other bit planes account for only about 10 percent of the image information; therefore, only the first three bit planes of the RGB channel are scrambled separately to save resources and ensure information security at the same time. After the decomposition of the three grey channels through the bit plane, three matrices of size $8 \times M \times N$ can be obtained.
3 Encryption algorithm description

The proposed encryption algorithm can be divided into three steps: 1, the plaintext image is taken as input and brought into the SHA256 function, and the key used for encryption is obtained by combining the external key and the plaintext pixel characteristics. 2, the bit-plane parity permutation and RGB permutation are performed. 3, the ciphertext is obtained by iterating the chaotic matrix from the hyper-chaotic sequence and the intermediate ciphertext matrix following the dynamic encoding method for DNA calculation. The overall block diagram of the encryption system is shown in Figure 4. The specific operation is as follows.

![Encryption schematic diagram](image)

### 3.1 Encryption key generation (initial value of chaotic sequence)

The algorithm key is divided into three parts, firstly, the two initial values of the one-dimensional sine-cos chaotic sequence used for the bit-plane permutation part. Secondly, the chaotic initial values required for the combined permutation of the RGB layers, and thirdly, the initial values of the hyper-chaotic sequence used for DNA encoding and computation. Firstly, the plaintext data is substituted into the hash function SHA256 to obtain a set of 256bit hash values, divide this set of values into 32 copies, each with 8 bits, which are denoted as \( H : (h_1, h_2, \ldots, h_{32}) \) in turn. Enter the external key \( E \), with a length of 100 bits. Then the external key is divided into five parts of 20 bits each, denoted as \( E : (e_1, e_2, \ldots, e_5) \), and then the two parts are combined to generate the key to increasing the security of the key. In this case, the key generated by RGB pixel layer scrambling is the pixel-average value of the plaintext image. All keys are highly correlated with plaintexts to increase key sensitivity and resist known plaintext attacks. The key generation formula is shown in Eq. (3).
\[
\begin{align*}
    k(1) &= \left\lfloor \left( [h_1 \oplus h_2 \oplus h_3 \oplus h_4 \oplus h_5] \times 10^7 \right) + [\left(e_1 \oplus e_2 \oplus e_3 \right) \times 10^7 \right\rfloor \\
    k(2) &= \left\lfloor \left( [h_1 \oplus h_2 \oplus h_3 \oplus h_4 \oplus h_5 \oplus h_6 \oplus h_7] \times 10^7 \right) + [\left(e_1 \oplus e_2 \oplus e_3 \right) \times 10^7 \right\rfloor \\
    k(3) &= \left\lfloor \left( \bar{I} \times 10^4 \right) + 10^7 \right\rfloor \\
    k(4) &= \left\lfloor \left( [h_{13} \oplus h_{15} \oplus h_{17} \oplus h_{19} \oplus h_{21}] \times 10^7 \right) + [\left(e_1 \oplus e_2 \oplus e_3 \right) \times 10^7 \right\rfloor \\
    k(5) &= \left\lfloor \left( [h_{13} \oplus h_{15} \oplus h_{17} \oplus h_{19} \oplus h_{21} \oplus h_{23}] \times 10^7 \right) + [\left(e_1 \oplus e_2 \oplus e_3 \right) \times 10^7 \right\rfloor \\
    k(6) &= \left\lfloor \left( [h_{13} \oplus h_{15} \oplus h_{17} \oplus h_{19} \oplus h_{21} \oplus h_{23} \oplus h_{25}] \times 10^7 \right) + [\left(e_1 \oplus e_2 \oplus e_3 \right) \times 10^7 \right\rfloor \\
    k(7) &= \left\lfloor \left( [h_{13} \oplus h_{15} \oplus h_{17} \oplus h_{19} \oplus h_{21} \oplus h_{23} \oplus h_{25} \oplus h_{27}] \times 10^7 \right) + [\left(e_1 \oplus e_2 \oplus e_3 \right) \times 10^7 \right\rfloor \\
\end{align*}
\]

Where \(\lfloor \cdot \rfloor\) is downward rounding operation, and \(\oplus\) is XOR operation. The \(\bar{I}\) is mean value of plaintext image pixels.

3.2 Scrambling operation

Step 1: Assuming that the plaintext image has a size of \(M \times N\), image \(I\) is divided into three channels, denoted as \(I_1, I_2, I_3\), and three bit-plane image matrices of size \(8 \times M \times N\) are obtained by bit-plane decomposition. Where the 8-bit plane matrix of the R channel is denoted as \(R_{ib} : (R_{ib1}, R_{ib2}, ..., R_{ib8})\), and G channel is \(G_{ib} : (G_{ib1}, G_{ib2}, ..., G_{ib8})\), B channel is \(B_{ib} : (B_{ib1}, B_{ib2}, ..., B_{ib8})\).

Step 2: The bit-plane matrices of the RGB three channels are presented in layers 8, 7 and 6, respectively, to form a new matrix, which is denoted as \(T : (R_{ib1}, R_{ib2}, R_{ib3}, G_{ib1}, G_{ib2}, G_{ib3}, B_{ib1}, B_{ib2}, B_{ib3})\). The nine matrices in the matrix \(T\) of size \(9 \times M \times N\) are then arranged in the order 1, 3, 5, 7, 9, 2, 4, 6, 8 to obtain the scrambled image.

Step 3: The parameter \(\mu\) of the sine-cos chaotic sequence was set as 3.9999, \(K(1)\) and \(K(2)\) were substituted into chaos Eq. (1) as initial values, and two chaotic sequences were obtained by iterating \(M + p1\) and \(9N + p2\) times respectively. To eliminate the transient effect, the former \(p1\) and \(p2\) terms are removed to obtain chaotic sequence \(P_{xi} (i = 1, 2, ..., M)\) with length \(M\) and chaotic sequence \(P_{yi} (j = 1, 2, ..., 9N)\) with length \(9N\). And then the sequences are quantized, and the value ranges of the two sequences are respectively converted to \((1, M)\) and \((1, 9N)\). The position index sequence is generated by ascending sort: \(Ux_i (i = 1, 2, ..., M)\) and \(Uy_j (j = 1, 2, ..., 9N)\). The normalization and sorting are shown in Eq. (4). As the scrambling sequence of rows and columns respectively, the element with subscript \((i, j)\) in the bit plane matrix is placed at the position with subscript \((Ux_i, Uy_j)\) to complete bit scrambling. After the bit-level disruption, we get \(T_2 : (T_{21}, T_{22}, ..., T_{29})\), after that, the matrices in \(T_2\) are restored to the RGB layer in order, obtain the intermediate ciphertext image \(D_{x1, y1, z1}\), which completes the dislocation operation of the bit layer.

\[
\begin{align*}
    P_{xi} &= (\left\lfloor \left( P_{xi} + 100 \right) \times 10^{10} \right\rfloor \mod M + 1 \\
    P_{yi} &= (\left\lfloor \left( P_{yi} + 100 \right) \times 10^{10} \right\rfloor \mod 9N + 1 \\
    Ux_i &= \text{sort}(P_{xi}, \text{`ascent'}) \\
    Uy_j &= \text{sort}(P_{yi}, \text{`ascent'})
\end{align*}
\]

Where the mod is remainder operation, \text{`ascent'} stands for sorted in ascending order.

Step 4: To further reduce the correlation between the RGB layers of image pixels and to improve the encryption algorithm's resistance to clipping attacks, the matrix \(T3\) of size \(3 \times M \times N\) is generated by arranging \(D_{x1, y1, z1}\) in order and substituting \(K(3)\) into equation (1), iterating \(3 \times M \times N + p3\) times to obtain the chaotic sequence \(P_{zi} (i = 1, 2, ..., 3MN)\),...
and also removing the first $p$ points and normalizing them to $(1,3 \times M \times N)$ and arranging them in descending order to get the position sequence $U_z$, the quantization method is the same as for bit-layer dislocation, then using the position sequence $U_z$ to complete the dislocation of $T_3$. The pixel-level chaotic matrix $T_4$ is obtained.

3.3 DNA coding, operation and encoding

Step 1: $K(4), K(5), K(6), K(7)$ was substituted into Eq. (2) as the initial value. First, the hyper-chaotic system is iterated $M \times N + p$ times, removing the first $p$ items to generate four chaotic sequences $X_1(i = 1, 2, ..., M \times N)$, $Y_1(i = 1, 2, ..., M \times N)$, $Z_1(i = 1, 2, ..., M \times N)$, $H_1(i = 1, 2, ..., M \times N)$, which are combined and quantized into three matrices involved in DNA operations: $W_1(i = 1, 2, ..., M \times N)$, $W_2(i = 1, 2, ..., M \times N)$, $W_3(i = 1, 2, ..., M \times N)$. The three operation sequences are then transformed into two-dimensional matrices of the same size as the image matrix to participate in the operation. The normalized formula is shown in Eq. (5), and the DNA calculation formula is generated as in Eq. (6).

\[
\begin{align*}
X_1 &= \left[X_1 \times 10^4\right] \mod 256 + 1 \\
Y_1 &= \left[Y_1 \times 10^5\right] \mod 256 + 1 \\
Z_1 &= \left[Z_1 \times 10^6\right] \mod 256 + 1 \\
H_1 &= \left[H_1 \times 10^6\right] \mod 256 + 1 
\end{align*}
\]  

(5)

Where "[ ]" is rounding operation, "mod" is remainder operation.

\[
\begin{align*}
W_1 &= X_1 \oplus Y_1 \oplus Z_1 \\
W_1 &= \text{reshape}(W_1, M, N) \\
W_2 &= X_1 \oplus Y_1 \oplus H_1 \\
W_2 &= \text{reshape}(W_2, M, N) \\
W_3 &= Y_1 \oplus Z_1 \oplus H_1 \\
W_3 &= \text{reshape}(W_3, M, N) 
\end{align*}
\]  

(6)

Where "reshape" representation matrix dimensional transformation.

Step2: Then, the chaotic system is iterated again $(M + s) \times (N + s) + p$ times and remove the first $p$ items to obtain four chaotic sequences $X_2(i = 1, 2, ..., M \times N + s^2)$, $Y_2(i = 1, 2, ..., M \times N + s^2)$, $Z_2(i = 1, 2, ..., M \times N + s^2)$, $H_2(i = 1, 2, ..., M \times N + s^2)$, where $s$ is the edge length when the matrix is encoded in blocks, i.e., the image matrix and the key matrix are divided into small blocks with edge length $s$ for encoding, which has the advantage of reducing the number of iterations to the hyper-chaotic system, so that each block has a corresponding encoding rule instead of each pixel point, as a way to reduce the consumption of resources for dynamic encoding. The four pseudo-random sequences are quantized and combined to finally generate five sequences used to determine the dynamic DNA encoding rules: $X_2$, $Y_2$, $Z_2$, $H_2$, $L_2$. Where $X_2$, $Y_2$, $Z_2$, and $H_2$ correspond to the encoding rules of the image matrix and the DNA operation matrix generated in the previous section, respectively, and $L_2$ corresponds to the decoding rules. It is quantized to the range $(1,8)$, thus representing 8 different ways of encoding (decoding). The randomness of DNA coding rules is greatly improved. Its calculation formula is as in Eq. (7).
\[
\begin{align*}
X_2 &= \left[ X_2 \times 10^4 \right] \mod 8 + 1 \\
Y_2 &= \left[ Y_2 \times 10^4 \right] \mod 8 + 1 \\
Z_2 &= \left[ Z_2 \times 10^4 \right] \mod 8 + 1 \\
H_2 &= \left[ H_2 \times 10^4 \right] \mod 8 + 1 \\
L_2 &= \left[ X_2 + Y_2 + Z_2 + H_2 \right] \mod 8 + 1
\end{align*}
\] (7)

Step 3: The RGB layers of the intermediate cipher image \( T_4 \) are separated and noted as \( T_{4_R}, T_{4_G}, T_{4_B} \). Each layer is encoded separately according to the dynamic encoding matrix generated in the previous step, while the operation matrices \( W_1, W_2, W_3 \) are encoded, and then DNA operations are performed. In the operation stage, this paper adopts a combination of addition, subtraction and dissimilarity to perform the operation. Taking the R channel as an example. The operation formula is as in Eq. (8).

\[
\begin{align*}
Q_1 &= \text{DNAbian}(T_{4_R}, X_2) \\
Q_2 &= \text{DNAbian}(W_1, Y_2) \\
Q_3 &= \text{DNAbian}(W_2, Z_2) \\
Q_4 &= \text{DNAbian}(W_3, H_2) \\
Q_R &= Q_2 + Q_1 \odot Q_3 - Q_4
\end{align*}
\] (8)

Where "\( \text{DNAbian} \)" indicates that the first term in the parentheses is encoded in DNA according to the rules indicated in the second term, and the encoding and calculation operations are completed to obtain the three-channel DNA matrix: \( Q_R, Q_G, Q_B \).

Step 4: After performing the DNA operation, the image's pixel value is changed for diffusion, and the DNA dynamic decoding operation is performed. Again, decode the R, G, and B channels separately. The calculation equation is as in Eq. (9).

\[
\begin{align*}
Q_R &= \text{DNAjie}(Q_R, L_2) \\
Q_G &= \text{DNAjie}(Q_G, L_2) \\
Q_B &= \text{DNAjie}(Q_B, L_2)
\end{align*}
\] (9)

Where "\( \text{DNAjie} \)" stands for DNA decoding operation, the first term in parentheses is decoded according to the decoding rules represented by the second sequence. After the decoding is completed, \( Q_R, Q_G, Q_B \) are combined into the final ciphertext \( C \). The algorithm's decryption process is the inverse of the encryption process, which is not described here, and the block diagram of the decryption principle is shown in Figure 5.
4 Experimental simulation results and performance analysis

This experiment's simulation test environment is MATLAB R2016a, and the standard 512×512 color RGB images are adopted. The simulation results of encryption and decryption are as follows in Figure 6.

The experimental figure clearly shows that the ciphertext image obtained after encryption of the plaintext image is a noise-like image with uniform pixels, the ciphertext image contains no helpful information, and the decryption algorithm can successfully recover the original image data.
4.1 Anti-statistics attack analysis

To effectively against statistical analysis attacks, an exemplary encryption algorithm must be capable of conceal plaintext statistics. The grey histograms of the RGB channels of the original image and the secret image are shown in Figure 7 and Figure 8.
Figure 7. Grey histogram of original image. (a) R-channel raw histogram; (b) G-channel raw histogram; (c) B-channel raw histogram.

Figure 8. Grey histogram of encryption image. (a) R-channel encryption histogram; (b) G-channel encryption histogram; (c) B-channel encryption histogram.

From Figure 7-8, the original image's histogram distribution is irregular and has solid statistical characteristics. In contrast, the ciphertext is distributed uniformly, and the algorithm significantly conceals the image's statistical characteristics. As a result, the algorithm can withstand statistical attacks. Figure 9, Figure 10, and Figure 11 display the scatter diagram of 5000 adjacent pixels chosen randomly from each direction, and classify the R, G, and B channels separately.
As can be seen from Figure 9-11, the majority of the pixel points in the original plaintext image are concentrated around the diagonal direction. There is a robust inter-pixel correlation, while the ciphertext image's pixel points after this algorithm's encryption are evenly distributed with almost no correlation, which confirms the algorithm's ability to withstand statistical feature attacks.
The adjacent pixels of grey-scale image data have a strong correlation, making the attacker obtain some rules to crack the image. Therefore, the encryption algorithm minimizes the correlation between adjacent pixels by scrambling close to nil to resist statistical property analysis attacks. The correlation between pixels is calculated using Eq. (10).

\[
\begin{align*}
E(x) &= \frac{1}{N} \sum_{i=1}^{N} x_i \\
D(x) &= \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2 \\
\text{cov}(x, y) &= \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \\
R_{xy} &= \frac{\text{cov}(x, y)}{\sqrt{D(x) \times D(y)}}
\end{align*}
\]

(10)

Where \(x\) and \(y\) are the grey values of neighboring pixels in each directions on the three channels of the image. \(D(x)\) is the variance; \(E(x)\) is mathematical expectation; \(\text{cov}(x, y)\) is covariance; Table 3 displays the correlation of neighboring pixels in the R, G, and B channels in each direction in this experiment, as well as the comparison results with the competitive algorithm of the same category.

### Table 3. Pixel coefficient of correlation

<table>
<thead>
<tr>
<th>Image Type</th>
<th>Direction</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena original image</td>
<td>H</td>
<td>0.9743</td>
<td>0.9873</td>
<td>0.9632</td>
</tr>
<tr>
<td>Lena this paper</td>
<td>V</td>
<td>0.9750</td>
<td>0.9881</td>
<td>0.9648</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.9533</td>
<td>0.9743</td>
<td>0.9344</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>0.0090</td>
<td>0.0107</td>
<td>0.0008</td>
</tr>
<tr>
<td>Lena [28]</td>
<td>V</td>
<td>0.0149</td>
<td>0.0126</td>
<td>0.0074</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>-0.0045</td>
<td>0.0060</td>
<td>0.0221</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>-0.0033</td>
<td>-0.0013</td>
<td>0.0021</td>
</tr>
<tr>
<td>Mandrill original image</td>
<td>V</td>
<td>0.8928</td>
<td>0.8181</td>
<td>0.7678</td>
</tr>
<tr>
<td>Mandrill this paper</td>
<td>V</td>
<td>0.0161</td>
<td>0.0013</td>
<td>-0.0043</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>-0.0158</td>
<td>0.0012</td>
<td>-0.0072</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>0.9464</td>
<td>0.8999</td>
<td>0.8722</td>
</tr>
<tr>
<td>Mandrill [28]</td>
<td>V</td>
<td>0.0046</td>
<td>-0.0078</td>
<td>-0.0010</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>-0.0013</td>
<td>0.0007</td>
<td>0.0032</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>0.9452</td>
<td>0.9642</td>
<td>0.9125</td>
</tr>
<tr>
<td>House original image</td>
<td>V</td>
<td>0.9225</td>
<td>0.9189</td>
<td>0.8456</td>
</tr>
<tr>
<td>House this paper</td>
<td>V</td>
<td>-0.0099</td>
<td>0.0094</td>
<td>0.0107</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.0075</td>
<td>-0.0017</td>
<td>0.0073</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>0.9678</td>
<td>0.9789</td>
<td>0.9503</td>
</tr>
<tr>
<td>House [28]</td>
<td>V</td>
<td>0.0045</td>
<td>0.0168</td>
<td>0.0040</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.0067</td>
<td>0.0060</td>
<td>0.0221</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>-0.0045</td>
<td>0.0026</td>
<td>-0.0008</td>
</tr>
<tr>
<td>Peppers original image</td>
<td>V</td>
<td>0.9755</td>
<td>0.9798</td>
<td>0.9553</td>
</tr>
<tr>
<td>Peppers this paper</td>
<td>V</td>
<td>0.0045</td>
<td>0.0168</td>
<td>0.0040</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>0.0067</td>
<td>0.0060</td>
<td>0.0221</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>-0.0045</td>
<td>0.0026</td>
<td>-0.0008</td>
</tr>
</tbody>
</table>
The pixel point correlation values of the current encryption algorithm and similar encryption algorithms are listed in Table 3; the correlation of adjacent pixels in the unencrypted image is extremely high, close to one, with solid correlation distribution characteristics, whereas the correlation in the encrypted image is nearly zero, implying that the current algorithm can withstand statistical characteristic attacks effectively.

4.2 Key space analysis

To ensure that the algorithm is secure enough, an algorithm with reasonable performance must be able to resist exhaustive attacks, which means the key space must be larger than $2^{128}$. The key in this paper is divided into four parts: three initial values of a sine-cos chaotic sequence: $(K(1), K(2), K(3))$, four hyper-chaotic Chen system initial values: $(K(4), K(5), K(6), K(7))$, chaos control parameters: $\mu$, external key: $E: ('36FC49AD2E4C27F796CA9C48B')$. Assuming the computer's precision is $10^{15}$, the algorithm at least has a key space of $10^{150}$, which is much larger than $2^{128}$ and sufficient to withstand brute force attack.

4.3 Entropy analysis

The information entropy of image data reflects the randomness, or chaos, of image data. Cypher text's ideal information entropy should be 8. Table 4 compares the proposed algorithm's information entropy to that of similar algorithms, and it is known that the information entropy of the proposed algorithm is closer to the ideal value. The information entropy formula is given by Eq. (11).

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) \log_2 \left( \frac{1}{p(m_i)} \right)$$ (11)

Where $n$ is the pixel value's number of digits. $p(m_i)$ is the probability of the pixel's grey level. By calculation, Table 4 lists the information entropy values of plaintext and ciphertext for each channel.

<table>
<thead>
<tr>
<th>Image Type</th>
<th>Channel</th>
<th>Plaintext Entropy</th>
<th>Ciphertext Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena this paper</td>
<td>R</td>
<td>7.2682</td>
<td>7.9993</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>7.5901</td>
<td>7.9992</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>6.9951</td>
<td>7.9994</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>7.6736</td>
<td>7.9993</td>
</tr>
<tr>
<td>Mandrill this paper</td>
<td>G</td>
<td>7.4469</td>
<td>7.9994</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>7.7316</td>
<td>7.9993</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>7.4270</td>
<td>7.9991</td>
</tr>
<tr>
<td>House this paper</td>
<td>G</td>
<td>7.2493</td>
<td>7.9994</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>7.4508</td>
<td>7.9992</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>7.3279</td>
<td>7.9993</td>
</tr>
<tr>
<td>Peppers this paper</td>
<td>G</td>
<td>7.5466</td>
<td>7.9993</td>
</tr>
</tbody>
</table>
Table 4 shows that the entropy of the experimental graph after encryption by the encryption algorithm designed in this paper is very near the ideal value. We can see that the algorithm's security in this paper is slightly higher when compared to other similar algorithms.

4.4 Key sensitivity and differential attack analysis

The key sensitivity is determined by the fact that no matter how minor the key changes, the secret image cannot be correctly decrypted, or no useful information can be obtained. Figure 12 shows that, the decryption graph of $\mu, K(3), K(6)$ changing $10^{-15}$ respectively. Where the correct key for $\mu$ should be 3.9999, the correct key for $K(3)$ is 0.1282, and then the correct key for $K(6)$ should be 0.3047. Because this minor key change has a magnitude of $10^{-15}$, the three keys after the change are: 3.999900000000001, 0.128200000000001, and 0.304700000000001. Under extreme conditions of $10^{-15}$ orders of magnitude, the key’s tiny changes in the graph cannot obtain useful information at all, so the algorithm’s key sensitivity is very high.

Differential attack is a plaintext attack that can change an image pixel by encrypting two images, determining their corresponding relationship, and obtaining valuable information. In general, as shown in Eq. (12), pixel change rate (\textit{NPCR})
and normalized average change intensity (UACI) are used to determine the encryption scheme's plaintext sensitivity and resistance to differential attack.

In this paper, five pairs of pixel points are chosen randomly from the plaintext, NPCR and UACI values are calculated when only one-pixel value changes. To test the algorithm's tolerance to differential attacks. As shown in Table 5, the NPCR and UACI values of the image are extremely close to ideal values when only one-pixel changes, indicating that the image is resistant to differential attack.

\[
\begin{align*}
\text{NPCR} &= \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} C(i, j)}{M \times N} \times 100% \\
\text{UACI} &= \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} |D(i, j) - D2(i, j)|}{M \times N} \times 100% \\
C(i, j) &= 0, P1(i, j) = P2(i, j) \\
C(i, j) &= 1, P1(i, j) \neq P2(i, j)
\end{align*}
\]

(12)

Where \(M\) and \(N\) are the image's length and width, respectively; The ciphertext obtained by encrypting the original Lena image is denoted by \(P1\), and \(P2\) is Wrong decrypted image after a tiny key change. The ideal NPCR and UACI values are 99.6094\% and 33.4635\%. As shown in Table 6, the error decrypted image after a tiny change in the key and the original ciphertext image are taken and the result is calculated by Eq. (12), respectively. Furthermore, the table also lists the NPCR and UACI values of similar algorithms in [31] and [32] for small changes in the key. Table 6 shows that the experimental data of this algorithm are extremely close to the theoretical value, demonstrating the algorithm's extremely high key sensitivity.

<table>
<thead>
<tr>
<th>Index</th>
<th>Location</th>
<th>NPCR</th>
<th>UACI</th>
</tr>
</thead>
<tbody>
<tr>
<td>(03,04)</td>
<td>(70,83)</td>
<td>99.6134</td>
<td>33.4659</td>
</tr>
<tr>
<td>(88,102)</td>
<td>(210,337)</td>
<td>99.6003</td>
<td>33.4704</td>
</tr>
<tr>
<td>(356,489)</td>
<td></td>
<td>99.6014</td>
<td>33.4588</td>
</tr>
</tbody>
</table>

Table 5. NPCR and UACI values with tiny pixel changes

<table>
<thead>
<tr>
<th>Lena</th>
<th>Channel</th>
<th>NPCR (%)</th>
<th>UACI (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K(3) change10^{-15}</td>
<td>R</td>
<td>99.6140</td>
<td>33.4652</td>
</tr>
<tr>
<td></td>
<td>G</td>
<td>99.5934</td>
<td>33.3814</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>99.6296</td>
<td>33.4506</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>99.6044</td>
<td>33.5015</td>
</tr>
<tr>
<td>K(5) change10^{-15}</td>
<td>G</td>
<td>99.6059</td>
<td>33.4878</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>99.6082</td>
<td>33.4190</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>99.6269</td>
<td>33.4750</td>
</tr>
<tr>
<td>K(7) change10^{-15}</td>
<td>G</td>
<td>99.6132</td>
<td>33.4481</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>99.5979</td>
<td>33.4395</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>99.6428</td>
<td>34.3935</td>
</tr>
<tr>
<td>Ref.[31]</td>
<td>G</td>
<td>99.6328</td>
<td>34.6547</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>99.6773</td>
<td>34.4841</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>99.6124</td>
<td>33.5715</td>
</tr>
<tr>
<td>Ref.[32]</td>
<td>G</td>
<td>99.6277</td>
<td>33.3356</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>99.6399</td>
<td>33.4044</td>
</tr>
</tbody>
</table>

Table 6. NPCR and UACI values under tiny variation of the key.
From the data in Table 6, in the case of small changes in the key, the $\text{NPCR}$ and $\text{UACI}$ values of the proposed algorithm in this paper are extremely close to ideal values. Compared to similar algorithms, the data are closer to the ideal values, indicating this algorithm's high key sensitivity.

4.5 Anti-noise analysis

This section examines the algorithm's anti-noise performance and compares the decryption results of the cipher text with varying degrees of salt and pepper noise. An algorithm with good noise immunity should be able to restore the image more clearly on the decryption side even after the image is scrambled by noise during transmission. Figure 13 shows the original and decrypted maps for the three noise scrambling cases. The noise intensities are 0.1, 0.2, and 0.3. It can be seen that the decryption algorithm can successfully recover image information that has been interfered with by noise.

![Image](image_url)

Figure 13. The decryption image in various noise densities. (a) The encryption image with 0.1 noise intensity; (b) The encryption image with 0.2 noise intensity; (c) The encryption image with 0.3 noise intensity; (d) The decryption image with 0.1 noise intensity; (e) The decryption image with 0.2 noise intensity; (f) The decryption image with 0.3 noise intensity;

Here, the concept of $\text{PSNR}$ is used to analyze image quality quantitatively. As shown in Eq. (13), $\text{PSNR}$ is an essential indicator for determining image quality. The higher the value, the higher the quality of image encryption.

$$\text{PSNR} = 10 \times \log(\frac{L^2}{\text{MSE}}) = 10 \times \log(\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (f(i, j) - f'(i, j))^2)$$  \hspace{1cm} (13)

In the formula, $M$ and $N$ are image size information; In $\text{MSE}$, $f(i, j)$ is the original pixel value; $f'(i, j)$ is pixel value with reduced quality; It determines the degree of distortion by calculating the mean square value. Table 7 shows the noise of different densities in each channel. Furthermore, compared with the competitive algorithm, the algorithm in this paper has a higher value, indicating that the image encryption quality is better.
Table 7. PSNR compared with Ref.[33]

<table>
<thead>
<tr>
<th>Density</th>
<th>Channel</th>
<th>Ref.[33]</th>
<th>Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>0.0001</td>
<td>66.85</td>
<td>67.84</td>
<td>66.79</td>
</tr>
<tr>
<td>0.001</td>
<td>57.21</td>
<td>57.89</td>
<td>57.38</td>
</tr>
<tr>
<td>0.01</td>
<td>47.24</td>
<td>47.29</td>
<td>47.33</td>
</tr>
</tbody>
</table>

As seen in Table 7, the results of the x-values calculated under different noise attacks, compared with similar algorithms, show that the security and resistance to noise attacks of this algorithm are higher.

4.6 Anti-trimming analysis

Images may suffer damage during transmission. This paper's encryption algorithm was tested for anti-clipping capability, Figure 14 shows the encrypted and decrypted maps when cropping different sizes, respectively.

![Fig 14. Decryption image.](image)

It can be seen from the image that when trimming the middle as well as the upper left quarter is more straightforward than when trimming one-half. However, most of the image information can still be recovered when trimming one-half, which indicates that this algorithm is robust against trimming attacks.

4.7 Time Complexity Analysis

Time complexity is an essential metric for assessing an algorithm's performance. A good algorithm should not only have good results, but its time complexity also determines its practicality. Due to the differences in the implementation platform, hardware configuration and so on. It is not appropriate to use the encryption and decryption time to measure the merit of an
algorithm, and the concept of time complexity is needed. First, the scrambling process is divided into two parts: bit-plane scrambling and RGB layer pixel-level scrambling. The time complexity of the bit-plane disorder is $\Theta(9M + N)$; the time complexity of the prevalent disorder is $\Theta(3M \times N)$. The hyperchaotic Chen system used in this paper is a four-dimensional, so the time complexity of generating four sequences of operations in the DNA encoding stage is $\Theta(4M \times N)$, and the time complexity of generating four dynamically encoded sequences is $\Theta(4M \times N \div t^2)$. The time complexity of each channel for DNA encoding or decoding is $\Theta(4M \times N)$. Four matrices of size $M \times N$ are involved in the operation for DNA operations, so the time complexity is $\Theta(4M \times N)$. Therefore, the time complexity of the algorithm in this paper is $\Theta(4M \times N)$.

5 Conclusion

This paper proposes a new chaotic image encryption algorithm based on bit-plane and DNA multiplex operation. We also design a new bit-plane disruption, key generation, and DNA operation method. Moreover, the key generation is highly correlated with the plaintext, which can be well protected against selective plaintext attacks. The correlation between plaintext bit layers and RGB is well hidden with low operation complexity. The quantitative analysis of the algorithm performance and comparison with other algorithms in the same category show that the algorithm in this paper can effectively resist differential and statistical analysis attacks, almost ideal information entropy, ample key space, key sensitivity, and nearly ideal information entropy.
