Article

On a Symmetric Image Encryption Algorithm Based on the Peculiarity of Plaintext DNA Coding

Yulan Kang 1,†, Linqing Huang 1,†, Yan He 2,*, Xiaoming Xiong 1 and Shuting Cai 1 and Haiyu Zhang 1

1 School of Automation, Guangdong University of Technology, Guangzhou 510006, China; kyl_2111804341@mail2.gdut.edu.cn (Y.K.); 2111604026@mail2.gdut.edu.cn (L.H.);
xmlxiong@gdut.edu.cn (X.X.); shutingcai@gdut.edu.cn (S.C.); hzhan46@uic.edu (H.Z.)

2 School of Biomedical and Pharmaceutical Sciences, Guangdong University of Technology, Guangzhou 510006, China

* Correspondence: heyan129@gdut.edu.cn

† Yulan Kang and Linqing Huang contributed equally to this work.

Received: 14 July 2020; Accepted: 19 August 2020; Published: 21 August 2020

Abstract: Many plaintext-related or non-plaintext-related image encryption algorithms based on chaotic systems have been found inefficient and insecure under chosen plaintext attacks. In this paper, a novel plaintext-related mechanism based on the peculiarity of plaintext DNA coding (PPDC) is presented and used to developed a symmetric image encryption algorithm. In our scheme, a hyper-chaotic Lorenz system is used to produce four chaotic sequences. Firstly, by using one chaotic sequence to control the DNA rules, the original image is encoded to obtain the image DNA coding and PPDC, and another chaotic sequence is encoded into a DNA sequence, which is used in the DNA XOR operation. Then, the processing of the remaining two chaotic sequences by using the PPDC is performed to obtain two key streams, which are used in the permutation stage. After performing the traditional permutation operation and DNA XOR operation, the cipher image is obtained. Because of the use of the PPDC, the key streams used in the permutation stage are related to the secret keys and plaintext image, which gives the encryption system higher plaintext sensitivity and security. The simulation experimental results and security analysis demonstrate that the proposed encryption system possesses high efficiency and security and can resist various typical attacks like exhaustive attack, statistical attack, and differential attack effectively.

Keywords: symmetric image encryption; hyper-chaotic system; DNA sequence; plaintext related

1. Introduction

The security of multimedia information communication has become the focus of attention due to the openness and sharing of network transmission, especially for digital images related to military, medical, and other fields that contain sensitive information. Therefore, developing a secure digital image encryption algorithm has become a vital research direction. However, for the inherent characteristics of digital images like the enormous data capacity, strong correlation among adjacent pixels, and high redundancy, the mainstream textual cryptography systems such as the Data Encryption Standard (DES), the Advanced Encryption Standard (AES), etc., are poorly suited to image encryption [1].

Chaotic systems possessing several salient features like extreme sensitivity to initial conditions and control parameters, ergodicity, aperiodicity, and randomicity have been used to generate pseudo-random sequences that are very complex and difficult to analyze and predict. In 1977, Fridrich firstly used the pseudo-random sequences generated by chaotic systems in image cryptosystems [2]. Subsequently, many chaos-based image encryption systems [1,3–18] have been constantly developed...
by researchers. For example, in [8], a new chaotic map was created based on a Beta function and used to develop a novel image encryption system with excellent resistance to various attacks. In 2019, Alawida et al. [14] proposed a hybrid digital chaotic system-based image cryptosystem, which has higher complexity, higher sensitivity, and a larger key space compared to other chaotic-based methods proposed previously. Some algorithms have been proven to be insecure due to the following drawbacks: (1) smaller key space; (2) lower plaintext sensitivity; (3) the chaotic sequence generated by the low-dimensional chaotic map is more likely to degenerate into a short-period sequence because of the finite precision of the computer; (4) the same key streams are used in the scrambling and diffusion operation when encrypting different original images. For example, A. Jain et al. presented a robust image encryption scheme [17] by using a one-dimensional logistic map and claimed that the cryptosystem could resist traditional attacks like statistical attack, exhaustive attack, and differential attack. In 2017, Dou et al. [19] analyzed and broke the encryption process by the chosen plaintext attack. More recently, Wang et al. [20] found an image encryption scheme using a 1D chaotic map [18] to be insecure and to be able to be completely cracked by the chosen plaintext.

In the last few years, the properties of DNA computing like enormous parallelism, higher information density, and ultra-low power consumption have been discovered, and many image encryption algorithms [10,16,17,21–32] have been designed by combining DNA computing and chaos. For example, in [17], the DNA operation combined with chaotic systems was used to develop a robust image encryption system. Firstly, the pixels in the original image are encoded into the DNA sequence, and a mask is generated by a chaotic map. Secondly, the diffusion process is implemented through the DNA addition and the complementary operation between the mask and the DNA encoded image. Finally, the diffused image is permuted using chaotic sequences generated by 2D chaos and decoded to obtain the cipher image. However, for some image encryption algorithms [10,16,17,21], all pixels of the original image are encoded/decoded by fixed DNA encoding/decoding rules, which lead to less randomness of the encryption schemes. For example, in [21], the original image was divided into four equal sub-images, which were encoded into DNA matrices according to a specific DNA encoding/decoding rule. The index of the employed DNA encoding/decoding rules, ranging from one to eight, is stationary and used for secret keys. However, it is insecure against exhaustive attack because it can be broken by a maximum of eight tests. Furthermore, including the algorithm in [21], most DNA operation-based schemes suffer similar drawbacks in the pure chaos-based image encryption algorithms mentioned above, which may lead to the encryption system being vulnerable to chosen plaintext attacks. To overcome the security flaws, many plaintext-related mechanisms and random DNA encoding/decoding technology have been proposed and used in image encryption recently. In [28], 2D-HSMM was used to control the DNA encoding/decoding rules randomly. Specifically, the parameter of temp calculated from the pixel value of the original image was used to manage the encryption process. However, the image cryptosystems developed in [28] have been broken with the chosen plaintext attack because of the insufficient plaintext association in 2019 [33]. In [22], the initial values and parameters of the chaotic system were calculated by the SHA 256 hash of the original image and modified by the plain image’s Hamming distance, leading to the high plaintext sensitivity of the encryption algorithm to resist the chosen plaintext attacks effectively. However, for algorithms like those in [22,31], due to encrypting different images with distinct keys, one must transfer the keys through a secret channel every time, which increases the implementation complexity and limits the application of this algorithm to real-time encryption.

The discussions above show that existing algorithms have one or more of the following defects: (1) the potential insecurity of low-dimensional chaotic systems; (2) the shortcomings of fixed DNA coding rules; (3) the security flaws of low sensitivity to the plain images; (4) the implementation complexity of the one-time-pad mechanism. To overcome these flaws, we propose a novel plaintext-related mechanism based on the peculiarity of plaintext DNA coding (PPDC), which is used to control the encryption process. The value of PPDC is determined by all pixel values of the original image and the chaotic sequence generated by a four-wing hyper-chaotic system, which can
significantly increase plaintext sensitivity to strengthen the ability to resist powerful chosen plaintext attacks.

The paper is organized as follows: In Section 2, the related technologies are introduced. Section 3 provides the detailed encryption process. The simulation and security analysis results are described in Section 4. Section 5 gives a brief conclusion.

2. Theory of the Proposed Algorithm

2.1. Hyper-Chaotic System

In our scheme, a 4D hyper-chaotic Lorenz system [34] is used to implement the encryption and is described by Equation (1).

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

where parameters \( a = 10, b = 8/3, c = 28, -1.52 \leq r \leq -0.06 \), and the hyper-chaotic Lorenz system is chaotic. By using the method of calculating the Lyapunov exponential spectrum of the differential equation system proposed in [35], if the parameter is set as \( r = -1 \), we get four Lyapunov exponents, namely \( \lambda_1 = 0.3381, \lambda_2 = 0.1586, \lambda_3 = 0, \lambda_4 = -15.1752 \). Furthermore, the hyper-chaos attractors are shown in Figure 1.

![Figure 1. Attractors of Lorenz's hyper-chaotic system with \( r = -1 \).](image)

As shown in Figure 1, compared with low-dimensional systems, the hyper-chaotic system has more complex strange attractors.

2.2. DNA Sequence Operations

A DNA sequence is made up of four nucleic acids, namely purine A (adenine), T (thymine), C (cytosine), and G (guanine). According to the Watson–Crick base pairing rules [28], A and G
are complementary to T and C, respectively. In the binary system, the binary data 00 and 01 are complementary to 11 and 10, respectively [31]. Therefore, if the four nucleic acids A, T, C, and G are used to encode binary numbers 00, 01, 10, and 11, respectively, we get $4! = 24$ kinds of coding rules. However, only eight kinds of coding rules match the principle of complementary base pairing, which are listed in Table 1.

In image encryption, the pixel values are converted into binary numbers firstly and then encoded into DNA sequence. Inversely, the DNA sequence can be decoded with the same coding rule to get the corresponding pixel value. For example, one pixel with a value of 188, the binary value of which is “10111100”, can be encoded as DNA sequence “TGGC” by Rule 3 and “GAAT” by Rule 7. On the other hand, the same DNA sequence can be decoded into different binary sequences using different DNA rules. For example, DNA sequence “GATC” can be decoded as “10001101” and “11100100” using Rules 1 and 5, respectively. The XOR operation of nucleic acids, which is used in the diffusion stage, is shown in Table 2.

<table>
<thead>
<tr>
<th>Table 1. Coding rules of the DNA sequence.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
</tr>
<tr>
<td>Rule 1</td>
</tr>
<tr>
<td>Rule 2</td>
</tr>
<tr>
<td>Rule 3</td>
</tr>
<tr>
<td>Rule 4</td>
</tr>
<tr>
<td>Rule 5</td>
</tr>
<tr>
<td>Rule 6</td>
</tr>
<tr>
<td>Rule 7</td>
</tr>
<tr>
<td>Rule 8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. XOR operation of the DNA sequence.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XOR</strong></td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>G</td>
</tr>
</tbody>
</table>

3. The Proposed Image Cryptosystem

3.1. PPDC-Based Image Encryption Algorithm

The flowchart of the proposed encryption system is shown in Figure 2.

Figure 2. Flowchart of the proposed image encryption algorithm. PPDC, peculiarity of plaintext DNA coding.

Without loss of generality, we assume that the original image $I$ has a size of $M \times N$. Firstly, reshape $I$ into one-dimensional array $P$ of length $M \times N$. In our scheme, four secret keys \{x0, y0, z0, w0\} are chosen as the initial values of the 4D Lorenz hyper-chaotic system, which is solved using the fourth-order Runge–Kutta algorithm with step size $h = 0.002$. The former $N_0$ iteration values are discard to obtain four random sequences $x_n$, $y_n$, $z_n$, and $w_n$ with the same size of $r = M \times N$, given by Equation (2).
\[
\begin{align*}
    x_n &= \{x_1, x_2, x_3, \cdots, x_r\} \\
y_n &= \{y_1, y_2, y_3, \cdots, y_r\} \\
z_n &= \{z_1, z_2, z_3, \cdots, z_r\} \\
w_n &= \{w_1, w_2, w_3, \cdots, w_r\}
\end{align*}
\]

Then, perform Equation (3) on the two sequences \(x_n\) and \(y_n\) to obtain two other new sequences \(X\) and \(Y\).

\[
\begin{align*}
    X &= \text{floor}(\text{mod}(x_n(i) \times 10^{15}, 8)) + 1 \\
    Y &= \text{floor}(\text{mod}(y_n(j) \times 10^{15}, 256))
\end{align*}
\]

where \(i, j = 1, 2, \cdots, r\). The sequence \(X\) is used as the DNA coding rules.

Next, convert the array \(P\) and sequence \(Y\) into binary arrays \(P_b\) and \(Y_b\), respectively. Then, by using sequence \(X\) as the coding rules, binary arrays \(P_b\) and \(Y_b\) are encoded to get \(\text{DNA}_P\) and \(\text{DNA}_Y\), respectively. Note that each \(\text{DNA}_P^i\) and \(\text{DNA}_Y^j\), \(i = 1, 2, \cdots, M \times N\) consists of four elements, which are represented by “A”, “T”, “C”, “G”.

To improve the plaintext sensitivity, a novel plaintext-related mechanism based on the peculiarity of plaintext DNA coding (PPDC) is presented. Firstly, calculate the numbers of A, T, C, and G in \(\text{DNA}_P\), which are named as \(\text{num}_A\), \(\text{num}_T\), \(\text{num}_C\), and \(\text{num}_G\), respectively. Then, the values of \(\text{num}_A\), \(\text{num}_T\), \(\text{num}_C\), and \(\text{num}_G\) are used to calculate PPDC by Equations (4)–(8). Obviously, because the attackers cannot know the elements’ value of \(X\), it is entirely impossible to get the value of PPDC and to select or construct special images to crack the proposed algorithm.

\[
\begin{align*}
    \text{num}1 &= \text{mod}(\text{num}_A \times 10^{15} / (M \times N \times 4), 32768); \\
    \text{num}2 &= \text{mod}(\text{num}_T \times 10^{15} / (M \times N \times 4), 32768); \\
    \text{num}3 &= \text{mod}(\text{num}_C \times 10^{15} / (M \times N \times 4), 32768); \\
    \text{num}4 &= \text{mod}(\text{num}_G \times 10^{15} / (M \times N \times 4), 32768); \\
    \text{PPDC} &= \text{num}1 \times \text{num}2 \times \text{num}3 \times \text{num}4;
\end{align*}
\]

Finally, the PPDC is used to process the two chaotic sequences \(z_n\) and \(w_n\) to generate two sequences \(Z\) and \(W\) by the following equation, which are used in the DNA level permutation stage. The detailed permutation process is presented in Algorithm 1.

\[
\begin{align*}
    Z &= \text{floor}(\text{mod}((z_n(k) \times \text{PPDC})/M) + 1 \\
    W &= \text{floor}(\text{mod}((w_n(l) \times \text{PPDC})/4 \times N) + 1
\end{align*}
\]

After the permutation stage, the DNA XOR operation is performed to implement the diffusion stage, and the diffusion process is presented in Algorithm 2. After reshaping the encrypted binary array \(C\) into a matrix of size \(M \times N\), the image encryption process is complete.

Furthermore, Figure 3 gives the flow diagram of using the encryption-decryption system to encrypt an image of size \(3 \times 3\).
Algorithm 1 Permutation algorithm.

**Input:** The plaintext DNA coding DNA\textsubscript{P} and two sequences Z and W.

**Output:** Permutated plaintext DNA coding DNA\textsubscript{C}.

1. A matrix DNA\textsubscript{C} ← Reshape DNA\textsubscript{P}
2. for \( i = 1 : M \) do \{ temp = DNA\textsubscript{C}(i,:) \}
3. DNA\textsubscript{C}(i,:) = DNA\textsubscript{C}(Z(i,:),:)
4. DNA\textsubscript{C}(Z(i,:),:) = temp 
5. end for
6. for \( j = 1 : 4N \) do \{ temp = DNA\textsubscript{C}(:,j) \}
7. DNA\textsubscript{C}(:,j) = DNA\textsubscript{C}(:,W(j))
8. DNA\textsubscript{C}(,:,W(j)) = temp 
9. end for

Algorithm 2 Diffusion algorithm.

**Input:** Permutated plaintext DNA coding DNA\textsubscript{C} and DNA\textsubscript{Y}.

**Output:** Encrypted binary array C.

1. A vector New\textsubscript{P} ← Reshape DNA\textsubscript{C}.
2. for \( i = 1 : 4 \times M \times N \) do
3. DNA\textsubscript{T}(i) = New\textsubscript{P}(i) \oplus DNA\textsubscript{Y}(i)
4. end for
5. Encrypted binary array C ← Decode DNA\textsubscript{T} with rule X.

Figure 3. A simple example of the encryption scheme.
3.2. Decryption Process of the Proposed System

The symmetric encryption system uses a single-key encryption method, that is the same key can be used for both the encryption and decryption process, and decryption is the inverse process of encryption. In the decryption process, the receiver can decrypt cipher image $C$ using the secret key $K = \{x_0, y_0, z_0, w_0, N_0\}$. Firstly, reshape the image $C$ into a one-dimensional vector $Q$ of size $M \times N$. Then, using the same methods shown in the encryption process, four new random sequences $x_n, y_n, z_n,$ and $w_n$ are obtained. Finally, Algorithm 3 shows the detailed decryption process.

Algorithm 3 Decryption algorithm.

**Input:** The one-dimensional vector $Q$ and four random sequences $x_n, y_n, z_n,$ and $w_n.$

**Output:** Decrypted image $P.$

1. Obtain two new sequences $X$ and $Y$ using Equation (3).
2. $DNA_Q \leftarrow$ Encode $Q$ with rule $X.$
3. $DNA_Y \leftarrow$ Encode $Y$ with rule $X.$
4. for $i = 1 : 4 \times M \times N$ do
5. $New_p(i) = DNA_Q(i) \oplus DNA_Y(i)$
6. end for
8. Calculate PPDC by Equations (4)--(8).
9. Generate two sequences $Z$ and $W$ using Equation (9).
10. A matrix $DNA_C \leftarrow$ Reshape $New_p$.
11. for $i = 1 : M$ do
12. $DNA_C(i,:)$ = $DNA_C(Z(i,:),:)$
13. $DNA_C(Z(i,:),:) = temp$
14. end for
15. for $j = 1 : 4N$ do
16. $DNA_C(:,j)$ = $DNA_C(:, W(j))$
17. $DNA_C(:, W(j)) = temp$
18. end for
19. A array $DNA_C \leftarrow$ Reshape matrix $DNA_C$
20. Decrypted binary array $P_b \leftarrow$ Decode $DNA_C$ with rule $X.$
21. A matrix $P \leftarrow$ Reshape array $P_b.$

4. Experimental Simulation Results and Security Analysis

In order to verify the effectiveness and security of the proposed encryption system, some standard grayscale images are used to perform the various simulated experiments with the platform of MATLAB 8.6.0.267246(R2015b).

4.1. Simulation Results

This section, we implemented the simulation for standard images in the condition of $key_{x0} = 1.1,$ $key_{y0} = 2.2,$ $key_{z0} = 3.3,$ $key_{w0} = 4.4,$ and $N_0 = 2000.$ Figure 4a,b shows the original image and the encrypted image, respectively. According to the simulation results, one cannot identify the plaintext image information from the cipher image. Figure 4c is the decrypted image, which has completely restored the plaintext image information. The decrypted image with wrong key $key_{x0} = 1.1000000000000001,$ $key_{y0} = 2.2,$ $key_{z0} = 3.3,$ $key_{w0} = 4.4,$ and $N_0 = 2000$ is shown in Figure 4d, and we can see that the wrong key cannot recover the original image.
4.2. Differential Attack Analysis

Attackers often use a set of original images that are slightly different to access the same cryptosystem and obtain corresponding encrypted images. By analyzing the differences between the encrypted images, the correlation between the plain image and cipher image, cryptanalysts may deduce the vulnerabilities in the encryption system. The unified average changing intensity (UACI) and the number of pixels change rate (NPCR) are usually used as indicators of resistance to differential attack [36], which is denoted as Equation (10).
Thus, the key space volume can ensure that the proposed scheme resists the exhaustive attack. The four initial values of the hyper-chaotic Lorenz system are denoted as 

\[ (x_0, y_0, z_0) \in (-40, 40), \quad (x_0, y_0, z_0) \in (1, 81), \quad (x_0, y_0) \in (-250, 250), \quad \text{and} \quad N_0 \in (1000, 2500). \]

If all the initial values have the precision of 10^{-12}, we can calculate the key space as 

\[ \text{key}_{\text{space}} = 80 \times 10^{15} \times 80 \times 10^{15} \times 80 \times 10^{15} \times 500 \times 10^{15} \times 1500 \approx 2^{237} > 2^{128}. \]

Thus, the key space volume can ensure that the proposed scheme resists the exhaustive attack.

### 4.3.2. Secret Key Sensitivity

Key sensitivity can be evaluated by the difference degree between two cipher images, which is obtained by encrypting the same original image with two secret keys that have an extremely tiny alteration. There are five secret keys \((key_{x0}, key_{y0}, key_{z0}, key_{w0}, N_0)\) in our proposed scheme. To test the secret keys’ sensitivity, we select randomly one set of keys from the secret key space. The selected keys are used to encrypt the plaintext image to get the corresponding cipher image \(C_1\). If we want to evaluate the sensitivity of \(key_{x0}\), keep the other four keys \((key_{y0}, key_{z0}, key_{w0}, \text{and} \ N_0)\) unchanged,

\[
\begin{align*}
\text{NPCR} & = \frac{1}{W \times H} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} D(i,j) \\
\text{UACI} & = \frac{1}{W \times H} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} \frac{|c_1(i,j) - c_2(i,j)|}{255}
\end{align*}
\]

where \(c_1, c_2\) are the encrypted images corresponding to two plaintext images, which have one different pixel \(P_x, P_y = \text{mod}(P_x + 1, 256)\), and 

\[
D(i,j) = \begin{cases} 
0, & \text{if } c_1(i,j) = c_2(i,j) \\
1, & \text{if } c_1(i,j) \neq c_2(i,j)
\end{cases}
\]

In our test, four standard images were selected as original images, i.e., “Peppers”, “Camera”, “Lena”, and “Baboon”. Similar algorithms published in recent years were used to encrypt the same plain images, and the corresponding analysis results of NPCR and UACI are shown in Table 3. One can clearly observe that the proposed cryptosystem has more excellent performance compared to other schemes.

<table>
<thead>
<tr>
<th>Image</th>
<th>Size</th>
<th>Peppers</th>
<th>Camera</th>
<th>Lena</th>
<th>Baboon</th>
<th>Average Value</th>
<th>Theoretical Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>[28]</td>
<td>(256 × 256)</td>
<td>NPCR 0.996078, 0.996139, 0.996200, 0.995925, 0.996085, 0.996094</td>
<td>UACI 0.334953, 0.333316, 0.334169, 0.333822, 0.334856, 0.334635</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zhang2014improved</td>
<td>(512 × 512)</td>
<td>NPCR 0.996112, 0.996082, 0.996002, 0.995903, 0.996023, 0.996094</td>
<td>UACI 0.335265, 0.335574, 0.335079, 0.335281, 0.335299, 0.334635</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[37]</td>
<td>(256 × 256)</td>
<td>NPCR 0.995014, 0.994322, 0.994938, 0.994863, 0.994789, 0.994698</td>
<td>UACI 0.335265, 0.335574, 0.335079, 0.335281, 0.335299, 0.334635</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>(256 × 256)</td>
<td>NPCR 0.996201, 0.995987, 0.996063, 0.995865, 0.996029, 0.996094</td>
<td>UACI 0.335891, 0.334316, 0.332830, 0.333885, 0.334231, 0.334635</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[512 × 512]</td>
<td>NPCR 0.996197, 0.996013, 0.996178, 0.996004, 0.996098, 0.996094</td>
<td>UACI 0.334957, 0.334513, 0.334412, 0.334422, 0.334401, 0.334635</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3. Exhaustive Attack Analysis

In the following subsections, we calculate the key space and analyze the key sensitivity, which are important evaluation criteria for an image encryption system resisting the exhaustive attack.

#### 4.3.1. Security Key Space

A secure image cryptosystem should have a large enough secret key space to effectively resist exhaustive attacks. The four initial values of the hyper-chaotic Lorenz system are denoted as 

\((key_{x0}, key_{y0}, key_{z0}, key_{w0})\), and the iteration times \(N_0\) are used as the secret key in the proposed scheme. The ranges of these keys are 

\(key_{x0} \in (-40, 40), \quad key_{y0} \in (-40, 40), \quad key_{z0} \in (1, 81), \quad key_{w0} \in (-250, 250), \quad \text{and} \quad N_0 \in (1000, 2500). \)

If all the initial values have the precision of 10^{-12}, we can calculate the key space as 

\[ \text{key}_{\text{space}} = 80 \times 10^{15} \times 80 \times 10^{15} \times 80 \times 10^{15} \times 500 \times 10^{15} \times 1500 \approx 2^{237} > 2^{128}. \]

Thus, the key space volume can ensure that the proposed scheme resists the exhaustive attack.

#### 4.3.2. Secret Key Sensitivity

Key sensitivity can be evaluated by the difference degree between two cipher images, which is obtained by encrypting the same original image with two secret keys that have an extremely tiny alteration. There are five secret keys \((key_{x0}, key_{y0}, key_{z0}, key_{w0}, N_0)\) in our proposed scheme. To test the secret keys’ sensitivity, we select randomly one set of keys from the secret key space. The selected keys are used to encrypt the plaintext image to get the corresponding cipher image \(C_1\). If we want to evaluate the sensitivity of \(key_{x0}\), keep the other four keys \((key_{y0}, key_{z0}, key_{w0}, \text{and} \ N_0)\) unchanged,
and replace $key_{x0}$ with $key_{x0} + 10^{-15}$ to obtain a new set of secret keys. Then, encrypt the same original image with the new keys to obtain another encrypted image $C_2$. Finally, we use NPCR and UACI to analyze the differences between $C_1$ and $C_2$ according to Equation (10). After 200 trials, the average values of NPCR and UACI are calculated. As Table 4 shows, the NPCR and UACI values are approximate to the theoretical values, which proves that our scheme possesses high key sensitivity. Another four keys ($key_{y0}$, $key_{z0}$, $key_{w0}$, and $N_0$) are analyzed and calculated in the same way (note that the value of $N_0$ is changed by one).

Table 4. The secret key sensitivity analysis test.

<table>
<thead>
<tr>
<th></th>
<th>Peppers</th>
<th>Lena</th>
<th>Baboon</th>
<th>Fruits</th>
<th>Flower</th>
<th>Camera</th>
<th>Theoretical Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x0$</td>
<td>NPCR</td>
<td>0.995846</td>
<td>0.996181</td>
<td>0.996002</td>
<td>0.996061</td>
<td>0.996069</td>
<td>0.995972</td>
</tr>
<tr>
<td></td>
<td>UACI</td>
<td>0.334877</td>
<td>0.334551</td>
<td>0.334877</td>
<td>0.334134</td>
<td>0.333400</td>
<td>0.333695</td>
</tr>
<tr>
<td>$y0$</td>
<td>NPCR</td>
<td>0.995903</td>
<td>0.996277</td>
<td>0.996166</td>
<td>0.996114</td>
<td>0.996322</td>
<td>0.996246</td>
</tr>
<tr>
<td></td>
<td>UACI</td>
<td>0.333776</td>
<td>0.334713</td>
<td>0.334679</td>
<td>0.334484</td>
<td>0.334204</td>
<td>0.334865</td>
</tr>
<tr>
<td>$z0$</td>
<td>NPCR</td>
<td>0.996056</td>
<td>0.996159</td>
<td>0.995926</td>
<td>0.996029</td>
<td>0.996122</td>
<td>0.996201</td>
</tr>
<tr>
<td></td>
<td>UACI</td>
<td>0.334550</td>
<td>0.335127</td>
<td>0.334124</td>
<td>0.335011</td>
<td>0.334676</td>
<td>0.333056</td>
</tr>
<tr>
<td>$w0$</td>
<td>NPCR</td>
<td>0.996086</td>
<td>0.996151</td>
<td>0.996136</td>
<td>0.996094</td>
<td>0.996021</td>
<td>0.996185</td>
</tr>
<tr>
<td></td>
<td>UACI</td>
<td>0.334191</td>
<td>0.334661</td>
<td>0.334985</td>
<td>0.334401</td>
<td>0.334319</td>
<td>0.334408</td>
</tr>
<tr>
<td>$N_0$</td>
<td>NPCR</td>
<td>0.996155</td>
<td>0.996273</td>
<td>0.996104</td>
<td>0.995984</td>
<td>0.996094</td>
<td>0.995682</td>
</tr>
<tr>
<td></td>
<td>UACI</td>
<td>0.334661</td>
<td>0.334080</td>
<td>0.334646</td>
<td>0.334209</td>
<td>0.333572</td>
<td>0.334520</td>
</tr>
</tbody>
</table>

A visual key sensitivity test of $key_{x0}$ is given. Firstly, one key set $key(1) = \{1.1, 2.2, 3.3, 4.4, 2000\}$ is selected at random from the key space. Then, the value of $key_{x0}$ is changed by $10^{-15}$ while keeping the other four keys unchanged, so we can obtain another new set of keys denoted as $key(2) = \{1.100000000000001, 2.2, 3.3, 4.4, 2000\}$. Next, the original images, shown in Figure 5a, are encrypted with the key(1) and key(2) to obtain the encrypted images denoted as $C_3$ and $C_4$, which are shown in Figure 5b,c. The difference images of $C_3$ and $C_4$ are shown in Figure 5d. One can intuitively see that the encrypted images have significant changes; even the secret key makes a tiny change. Finally, encrypted images $C_4$ and $C_3$ are decrypted using key(1) and key(2) to get the decrypted images shown in Figure 5e,f. From Figure 5, we can see that the cipher image cannot be successfully restored, and any valid information in the original image cannot be obtained from the decrypted image even if the correct secret key has been changed slightly. Thus, our image cryptosystem possesses outstanding key sensitivity.

(a) The original image.  (b) The encrypted image with key(1).

Figure 5. Cont.
4.4. Statistical Attack Analysis

The correlation coefficient, histogram, and information entropy are three important evaluation criteria for statistical attack analysis.

4.4.1. Correlation Coefficient Test

Generally, the correlation between adjacent pixels is strong in the plaintext image and must be decreased close to zero after being encrypted by a secure encryption system (note that the theoretical value is zero). In the experiment, we calculate the correlation coefficients of all the adjacent pixels in the four directions of the plain image and the cipher image using Equation (11), and the calculation results are shown in Table 5. A comparison of the detailed results with similar algorithms is given in Table 6.

\[ r_{xy} = \frac{\text{cov}(u, v)}{\sqrt{D(u)} \times \sqrt{D(v)}} \]  

(11)

where \( \text{cov}(u, v) = \frac{1}{N} \sum_{i=0}^{N} (x_i - E(u))(y_i - E(v)) \), \( D(u) = \frac{1}{N} \sum_{i=0}^{N} (u_i - E(u))^2 \), \( E(u) = \frac{1}{N} \sum_{i=0}^{N} u_i \). \( N \) is the number of image pixels, and \( u, v \) are the gray values of adjacent pixels in the four directions mentioned above.

Furthermore, two-thousand pixels are randomly selected from the original image or its corresponding encrypted image. The correlation diagram among adjacent pixels in the four directions is shown in Figure 6. As Figure 6 shows, after encrypting, the values of the adjacent pixels are completely different.

---

![Figure 5. The secret key sensitivity test.](image_url)
To resist statistical attack, the histogram of encrypted image should be flat. In this section, four images are selected as the original image shown in Figure 7a. The histograms of the original images and their corresponding encrypted image are given in Figure 7b,c, respectively. Intuitively, the cipher image has a flat histogram, while original image histogram fluctuates greatly.

### Table 5. Correlation coefficients of plain images and the corresponding cipher images.

<table>
<thead>
<tr>
<th>Image</th>
<th>Original</th>
<th>Encrypted</th>
<th>Horizontal</th>
<th>Vertical</th>
<th>Diagonal</th>
<th>Antidiagonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>0.9593</td>
<td>0.0025</td>
<td>0.9719</td>
<td>0.0011</td>
<td>0.9850</td>
<td>0.9865</td>
</tr>
<tr>
<td>Baboon</td>
<td>0.9679</td>
<td>0.0019</td>
<td>0.9631</td>
<td>0.0019</td>
<td>0.9274</td>
<td>0.9366</td>
</tr>
<tr>
<td>Peppers</td>
<td>0.9860</td>
<td>0.00001</td>
<td>0.9853</td>
<td>0.0013</td>
<td>0.9784</td>
<td>0.9706</td>
</tr>
<tr>
<td>Airplane</td>
<td>0.9301</td>
<td>0.0026</td>
<td>0.8752</td>
<td>0.0014</td>
<td>0.8748</td>
<td>0.0028</td>
</tr>
</tbody>
</table>

### Table 6. Correlation coefficients of the encrypted Lena image with different encryption schemes.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Plain image</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>0.9593</td>
<td>0.0019</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.9679</td>
<td>0.0007</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.9860</td>
<td>0.0038</td>
</tr>
</tbody>
</table>

Figure 6. Correlation analysis of the original image and the corresponding encrypted image.

#### 4.4.2. Histogram Analysis

The image histogram reflects the frequency of the different grayscale throughout the digital image. To resist statistical attack, the histogram of encrypted image should be flat. In this section, four images are selected as the original image shown in Figure 7a. The histograms of the original images and their corresponding encrypted image are given in Figure 7b,c, respectively. Intuitively, the cipher image has a flat histogram, while original image histogram fluctuates greatly.

Furthermore, the uniform distribution of the cipher image histogram is analyzed by histogram variance in quantitative analysis, which is defined as follows:

$$Var(H) = \frac{\sum_{i=0}^{255} (h_i - \frac{M \times N}{256})^2}{256}$$  \hspace{1cm} (12)
where $H = \{h_0, h_1, \ldots, h_{255}\}$ is a one-dimensional array, $h_i$ represents the frequency of the pixel in the image, and $M$ and $N$ are the size of the test images. The smaller the value of the variance is, the more difficult it is to get the statistical information of the cipher image. The variance results of the cipher image histogram encrypted by different encryption algorithms are shown in Table 7. One can see that, compared with similar algorithms, the image histogram encrypted by our proposed scheme is flatter.

![Image](image_url)

(a) The original image.

(b) The histogram of the original image.

(c) The encrypted image.

(d) The histogram of the corresponding encrypted image.

Figure 7. The histogram analysis of the proposed scheme.

<table>
<thead>
<tr>
<th></th>
<th>Proposed</th>
<th>[3]</th>
<th>[5]</th>
<th>[22]</th>
<th>[13]</th>
<th>[28]</th>
<th>[37]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>957.2106</td>
<td>1104.711</td>
<td>1043.214</td>
<td>977.02</td>
<td>973.2578</td>
<td>1027.593</td>
<td>5118.094</td>
</tr>
<tr>
<td>Baboon</td>
<td>909.6752</td>
<td>1145.922</td>
<td>1061.038</td>
<td>–</td>
<td>1071.297</td>
<td>1058.132</td>
<td>5119.29</td>
</tr>
<tr>
<td>Peppers</td>
<td>1046.5407</td>
<td>1289.859</td>
<td>1105.716</td>
<td>–</td>
<td>1046.734</td>
<td>946.67</td>
<td>5403.283</td>
</tr>
<tr>
<td>Camera</td>
<td>890.0863</td>
<td>1035.719</td>
<td>976.015</td>
<td>–</td>
<td>1057.492</td>
<td>941.062</td>
<td>5381.261</td>
</tr>
</tbody>
</table>

Table 7. Variance of the encrypted gray image histogram with different algorithms.
4.4.3. Information Entropy Analysis

The greater the entropy of the system, the more uncertain the system. Information entropy is often used to measure the uncertainty of image information. The entropy is defined as follows:

\[
H(m) = \sum_{i=0}^{L} p(m_i) \log \frac{1}{p(m_i)}
\]  

(13)

where \( L \) is the grayscale of image (\( i \in L \)) and \( p(m_i) \) is the probability of the occurrence of grayscale \( m_i \). Assuming that an image with a grayscale of \( L = 256 \) and all the grayscale appears with the same probability, according to Equation (13), the theoretical value of information entropy \( H(m) \) is eight, according to Equation (12). Therefore, the bigger the information entropy of the encrypted image, the more secure the image cryptosystem is. In this section, many standard images are used to measure our scheme’s information entropy performance. The information entropy analysis results of the cipher images of Lena encrypted by our scheme and similar algorithms are also given. As shown in Tables 8 and 9, the experimental results of our scheme are closer to the theoretical value of eight than similar algorithms. Therefore, the proposed cryptosystem possesses high security when facing statistical attacks.

Table 8. Experimental results of information entropy.

<table>
<thead>
<tr>
<th>Image</th>
<th>Lena</th>
<th>Fruits</th>
<th>Baboon</th>
<th>Sailboat</th>
<th>Pepper</th>
<th>Flower</th>
<th>Satellite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain</td>
<td>7.4455</td>
<td>7.3644</td>
<td>7.3585</td>
<td>7.4853</td>
<td>7.5937</td>
<td>7.4107</td>
<td>6.5578</td>
</tr>
<tr>
<td>Encrypted</td>
<td>7.9993</td>
<td>7.9993</td>
<td>7.9993</td>
<td>7.9993</td>
<td>7.9993</td>
<td>7.9993</td>
<td>7.9993</td>
</tr>
</tbody>
</table>

Table 9. Experimental results of the information entropy of encrypted Lena.

<table>
<thead>
<tr>
<th>Image</th>
<th>Proposed</th>
<th>[1]</th>
<th>[3]</th>
<th>[4]</th>
<th>[8]</th>
<th>[13]</th>
<th>[21]</th>
<th>[38]</th>
<th>[27]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain</td>
<td>7.4455</td>
<td>7.4455</td>
<td>7.4455</td>
<td>7.4455</td>
<td>7.4455</td>
<td>7.4455</td>
<td>7.4455</td>
<td>7.4455</td>
<td>7.4455</td>
</tr>
</tbody>
</table>

4.5. Encrypted Time Analysis

In practical applications, in addition to the security performance, the cryptosystem’s running speed is also an important indicator. In this section, the execution time is evaluated in seconds. Table 10 shows the execution time of the proposed image cryptosystem and similar schemes [3,13,28,29]. One can see that our scheme has a better performance in encryption speed. Besides, we analyze the time complexity denoted by \( \Theta \). The computational complexity is related to the process of completing encryption/decryption. In our proposed scheme, the time consumption mainly includes the key streams’ generation, DNA encoding/decoding, permutation, and diffusion stages. For a grayscale image with size \( M \times N \), the time complexity of DNA encoding and decoding is \( \Theta(8 \times M \times N) \). In the permutation and diffusion phase, the manipulated object is the DNA sequence, and its complexity is \( \Theta(4 \times M \times N) \). Furthermore, the hyper-chaotic Lorenz system is solved to obtain chaotic sequences, in which the time complexity is \( \Theta(M \times N) \). Thus, the time complexity of the proposed scheme is \( \Theta(8 \times M \times N) \). Through time complexity analysis, the DNA encoding and decoding part has higher time complexity. However, in the future, with the advent of DNA computers, a DNA-based image encryption system can be massively executed in parallel mode.

Table 10. Encryption speed of the proposed algorithm and other encryption schemes.

<table>
<thead>
<tr>
<th>Image</th>
<th>Proposed Algorithm</th>
<th>[3]</th>
<th>[13]</th>
<th>[28]</th>
<th>[29]</th>
</tr>
</thead>
<tbody>
<tr>
<td>512 × 512</td>
<td>0.7449</td>
<td>2.6379</td>
<td>3.6208</td>
<td>1.3857</td>
<td>1.2843</td>
</tr>
<tr>
<td>256 × 256</td>
<td>0.1824</td>
<td>0.2985</td>
<td>0.9317</td>
<td>0.4397</td>
<td>0.2985</td>
</tr>
</tbody>
</table>
5. Conclusions

In this paper, a novel plaintext-related mechanism based on the peculiarity of plaintext DNA coding (PPDC) is proposed. The original image is firstly encoded to obtain the DNA encoding and the value of PPDC. Then, PPDC is used to process the chaotic sequences to obtain the key streams used in the permutation stage. Such key stream generation method is different from the one used in the previous DNA-based chaotic image encryption algorithms. Because the key streams are related to both the random sequences generated by the Lorenz hyper-chaotic system and plaintext image, our scheme can realize higher key sensitivity and plaintext sensitivity to resist chosen plaintext attacks or differential attacks effectively. Furthermore, the DNA level permutation operation not only disorganizes the position of the plain image pixels, but also changes the value of the pixel, which improve the efficiency and security. Especially, the encryption algorithm only needs to carry out one round of the permutation-diffusion operation to get a better ability of resisting various attacks. Secret key space analysis, histogram analysis, correlation analysis, sensitivity analysis, and information entropy analysis are given. The analysis results prove that the proposed algorithm is secure to resist all kinds of attacks, including the powerful chosen plaintext attack.

Author Contributions: Y.H., L.H., and Y.K. conceived of and designed the experiments; Y.K. and L.H. performed the experiments; X.X., S.C., and H.Z. analyzed the data; X.X. contributed reagents/materials/analysis tools; Y.K., Y.H., and L.H. wrote the paper. All authors read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Planning Project of Guangdong Province, China (No.2020B111110002, No.2019B010140002, No.2017B010124003, No.2017B09090001, No.2017B09090004).

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

References


