A RGB image encryption algorithm based on DNA encoding and chaos map

Lili Liu, Qiang Zhang *, Xiaopeng Wei

Key Laboratory of Advanced Design and Intelligent Computing, Dalian University, Ministry of Education, Dalian 116622, China

ABSTRACT

In this paper, a RGB image encryption algorithm based on DNA encoding combined with chaotic map is proposed aiming at characteristics of RGB image. The algorithm firstly carries out DNA encoding for R, G, B components of RGB image; then realizes the addition of R, G, B by DNA addition and carries out complement operation by using the DNA sequence matrix controlled by Logistic; three gray images are got after decoding; finally gets the encrypted RGB images by reconstructing R, G, B components which use image pixels disturbed by Logistic chaotic sequence. Simulation result shows that the proposed algorithm has a large secret key space and strong secret key sensitivity. Meanwhile, it can resist exhaustive attack, statistical attack, and thus it is suitable for RGB image encryption.

1. Introduction

With the development of computer network technology, digital image is widely used in various fields of society. However, due to openness of the network, the security of image is threatened seriously, so the image encryption becomes the most effective way to guarantee transmit security of images [1]. Chaos is seemingly a random movement of deterministic system. Chaos system has the properties of ergodicity, boundedness, sensitivity to initial conditions. Therefore, using chaotic system in image encryption can meet certain security requirements. However, the chaotic encryption algorithms [2–4] which utilize one-dimensional chaos map, multi-dimensional chaos map and ultra-dimensional chaos map are all to transform the image pixel position and pixel values, a lot of Refs. [5–7] point out that using encryption algorithms constituted by a single chaos map are vulnerable to be interpreted.

Nowadays, DNA computing has permeated the domain of cryptography. DNA cryptogram utilizes DNA as information carrier and takes advantage of biological technology to achieve encryption [8–11]. Kang et al. had proposed a character encryption algorithm based on pseudo DNA operation [12], it made use of central dogma which belongs to molecular biology to implement encryption. However, DNA encryption methods have disadvantages such as expensive experimental equipment, complex operation and difficult to grasp its biotechnology, and still cannot be efficiently applied in encryption field.

In order to overcome the defects of the situation that a single chaos encryption is easy to be decoded and DNA encryption needs biological experiment, we presents RGB image encryption algorithm based on DNA encoding and chaos map. The proposed algorithm utilizes DNA addition to scramble the pixel values of image R, G, B components and then encrypt the scrambled images. Experimental results show that the algorithm which is simple to implement, can resist a variety of attacks, and can be easily applied to color image to scramble encryption, very suitable to use in secure communication. This paper is organized as follows. In Section 2, we introduce the basic theory of the proposed algorithm. The details of designing the...
proposed image encryption are proposed in Section 3. Section 4 is simulation results. In Section 5, security analysis is discussed. Section 6 gives the conclusion.

2. Basic theory of the proposed algorithm

The algorithm mainly uses the encoded matrix of DNA sequence to carry out DNA addition operation, and uses Logistic mapping function to implement image encryption.

2.1. DNA encoding

DNA computing is a form of computing which uses DNA, biochemistry and molecular biology, instead of the traditional silicon-based computer technologies. DNA computing, or more generally, bimolecular computing, is a fast developing interdisciplinary area. With the rapid development of DNA computing, the researchers presented many biological operations and algebra operations based on DNA sequence [13]. Single-strand DNA sequence is composed by four bases, they are A, C, G and T, where A and T are complement to each other, so are C and G. In the modern theory of electronic computer, all information is expressed by binary system. But in DNA coding theory, information is represented by DNA sequences. So we use binary numbers to express the four bases in DNA sequence and two bits binary number to represent a base. In the theory of binary system, 0 and 1 are complementary, so we can obtain that 00 and 11 and 10 and 01 are also complementary. We can use 00, 01, 10 and 11 to express four bases and the number of coding combination kinds is $4! = 24$. Due to the complementary relation between DNA bases, there are only eight kinds of coding combinations that satisfy the principle of complementary base pairing in 24 kinds of coding combinations. Table 1 gives eight encoding rules:

Example: The binary pixel value of an image is [00111010], so the corresponding DNA sequence is [ATGG] according to the first encoding rule, similarly according to the seventh decoding rule, the decoding sequence is [11001010]. In the proposed algorithm, we put the eight encoding and decoding rules mapped to the eight sub-region of (0,1), and using the seed generated by random number to choose different rules.

2.2. The addition and subtraction operation of DNA sequence

Since the development of DNA computing, scholars have proposed using algebraic operation of DNA sequence to replace the traditional computer algebraic operation. Based on this, we use DNA addition operation to realize DNA sequence matrix computing for R, G, B components. The algorithm of this paper finds out DNA addition and subtraction rules by using mod 2 operations of binary figure when 01 – A, 10 – T, 00 – C, 11 – G, and you can find the rules in Table 2.

The Logistic map is a polynomial map (equivalently, recurrence relation) of degree 2, often cited as an archetypal example of how complex chaotic behavior can arise from very simple non-linear dynamical equations. The map was popularized in a seminal 1976 paper by the biologist Robert May, in part as a discrete-time demographic model analogous to the Logistic of how complex chaotic behavior can arise from very simple non-linear dynamical equations. The map was popularized in a seminal 1976 paper by the biologist Robert May, in part as a discrete-time demographic model analogous to the Logistic equation first created by Pierre François Verhulst [14]. One-dimensional Logistic map is the most widely applied chaos map currently. The chaotic models generated by it are also known as insect amount model. Its mathematical define is as follows:

$$x_{n+1} = \mu x_n (1 - x_n)$$  \hspace{1cm} (1)

where $\mu \in [0,4], x_n \in (0,1), n = 0, 1, 2, \ldots$, and we can find from the bifurcation diagram of Fig. 1 that when $2.6 < \mu \leq 4$ this dynamic system forks and generates two times period or four times period from the steady state. Vast multi-periods appear in the interval of smaller $\mu$, and after forking $n$ times, the length of period is $2^n$; when $3.569945 < \mu \leq 4$, the system accesses chaos state and the chaotic sequences generated by it are non-convergent, aperiodic and having certain sensitivity to initial value; when $\mu = 4$, the Logistic map is surjection and while $x_n \in (0,1)$ the chaotic sequences have ergodicity [15].

The chaotic sequence in the algorithm of this paper has generation ways as follows:

(1) The real value sequence is generated by Logistic map while initial value is $g_0$, $\mu_0$; namely $\{x_n, n = 0, 1, 2, \ldots\}$, it is formed by the chaotic map track points.

(2) Binary sequences: The real value sequence generated by chaotic map while initial value is $g_0, \mu_0$, through defining a threshold function $f(x)$ and the threshold is 0.5, the definition is as follows:

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Eight kinds of DNA map rules.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>00 – A</td>
<td>00 – A</td>
</tr>
<tr>
<td>01 – C</td>
<td>01 – G</td>
</tr>
<tr>
<td>10 – G</td>
<td>10 – C</td>
</tr>
<tr>
<td>11 – T</td>
<td>11 – G</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>00 – C</td>
<td>00 – C</td>
</tr>
<tr>
<td>01 – A</td>
<td>01 – A</td>
</tr>
<tr>
<td>10 – T</td>
<td>10 – T</td>
</tr>
<tr>
<td>11 – G</td>
<td>11 – G</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>00 – G</td>
<td>00 – G</td>
</tr>
<tr>
<td>01 – T</td>
<td>01 – T</td>
</tr>
<tr>
<td>10 – A</td>
<td>10 – A</td>
</tr>
<tr>
<td>11 – C</td>
<td>11 – C</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>00 – T</td>
<td>00 – T</td>
</tr>
<tr>
<td>01 – C</td>
<td>01 – C</td>
</tr>
<tr>
<td>10 – G</td>
<td>10 – G</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>00 – T</td>
<td>00 – T</td>
</tr>
<tr>
<td>01 – C</td>
<td>01 – C</td>
</tr>
<tr>
<td>10 – G</td>
<td>10 – G</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>01 – G</td>
<td>01 – G</td>
</tr>
<tr>
<td>10 – C</td>
<td>10 – C</td>
</tr>
<tr>
<td>11 – A</td>
<td>11 – A</td>
</tr>
</tbody>
</table>
$f(x) = \begin{cases} 0, & 0 < x_n \leq 0.5 \\ 1, & 0.5 < x_n < 1 \end{cases}$

We can transform the chaotic sequence of real value to chaotic sequences of binary.

3. Algorithm description

In this paper, we firstly split the RGB image into R, G, B components. Secondly, transform the R, G, B components into DNA code, and get three DNA sequences matrix, in order to disturb the correlation between the pixels in spatial domain and scramble the pixels of the different components, we carry out addition operation (Fig. 2) for three DNA sequence matrixes according to the rules and then complement with DNA sequences by the new chaotic sequences. Thirdly, get three gray images through decoding, and then disturb the three image pixels by using Logistic chaos map. Finally, combine R, G, B components to get encrypted RGB images. Block diagram of the algorithm is as Fig. 3.

The encryption processes of the specific steps are as follows:

(1) Input 8-bit color image $A(m,n,3)$, where $m$, $n$ are the image dimensionalities of rows and columns, respectively.

<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>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>C</td>
<td>A</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>T</td>
<td>C</td>
<td>T</td>
<td>C</td>
<td>A</td>
<td>T</td>
<td>C</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>G</td>
<td>C</td>
<td>T</td>
<td>A</td>
<td>T</td>
<td>G</td>
<td>C</td>
<td>T</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>T</td>
<td>G</td>
<td>A</td>
<td>C</td>
<td>A</td>
<td>T</td>
<td>G</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>

Fig. 1. The bifurcation diagram of the Logistic map.

Table 2
DNA addition and subtraction operation.

<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>
</tr>
</thead>
<tbody>
<tr>
<td>... CAAG TGGC...</td>
<td>... CAAG TGGC...</td>
<td>... AGCT TGAC...</td>
<td>... TCAG CATC...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>... AGCT GAGC...</td>
<td>... AGCT GAGC...</td>
<td>... TTCC GAGA...</td>
<td>... GCAG CTTA...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>... GTCC GAGA...</td>
<td>... GTCC GAGA...</td>
<td>... AGGA ATTT...</td>
<td>... ACAC CGAA...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>... ACAC CGGT...</td>
<td>... ACAC CGGT...</td>
<td>... AGCT CAGT...</td>
<td>... TCAC CAGT...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. DNA addition.
We carry out addition operation by using DNA pseudo operation for DNA sequence matrixes $R(m,n \times 8)$, $G(m,n \times 8)$ and $B(m,n \times 8)$, then encoded respectively in accordance with the DNA encoding rules selected by seed $key1$ ($key1$ is a random figure and $key1 \in [1,8]$) and get three DNA sequence matrixes $Ar(m,n \times 4)$, $Ag(m,n \times 4)$ and $Ab(m,n \times 4)$.

(3) We carry out addition operation by using DNA pseudo operation for DNA sequence matrixes $Ar(m,n \times 4)$, $Ag(m,n \times 4)$ and $Ab(m,n \times 4)$ according to Table 3. The addition rules are $Ar(i,j) = Ar(i,j) + Ag(i,j)$, $Ag(i,j) = Ag(i,j) + Ab(i,j)$ and $Ab(i,j) = Ag(i,j) + Ab(i,j)$.

(4) Generate the chaotic sequence $x_n$ whose length is $l = m \times n \times 8/2$ by using Logistic chaos in the condition of initial value is $g_0$ and the system parameter is $\mu_0$. Then we use the following threshold function $f(x)$ (Eq. (2)) to get a binary sequence $z(1,l)$, and transform it to $z(m,n \times 4)$.

(5) When $z(i,j) = 1$ the bases of the corresponding locations of the $Ar(i,j)$, $Ag(i,j)$ and $Ab(i,j)$ DNA matrix are complement, otherwise unchanged.

(6) On the results of (5), according to the DNA map rules selected by seed $key2$ ($key2$ is a random figure and $key2 \in [1,8]$) carry out binary recovering operation, then we will obtain binary matrixes $R'(m,n \times 8)$, $G'(m,n \times 8)$ and $B'(m,n \times 8)$.

(7) Produce a chaotic sequence $c$ whose length is $m \times n$ by using Logistic chaos map system in condition of initial value is $g_1$ and system parameter is $\mu_1$. Map the elements of $c$ from $(0,1)$ to $(0,1,2, \ldots, 255)$ and transform $c$ to binary matrix $c(1,m \times n \times 8)$, then restructure to $c(m,n \times 8)$.

(8) Carry out exclusive or operation for $c$ with $R'$, $G'$ and $B'$, respectively, get three new matrix $R''$, $G''$ and $B''$.

(9) Finally, recover RGB image and that is the encrypted color images.

From the above algorithm analysis, the attacker must possess Logistic chaos map parameters $g_0$, $\mu_0$, $g_1$, $\mu_1$, keys of encoding and decoding rules $key1$, $key2$, which are totally six keys. The encryption algorithm greatly improves the security of images. The algorithm decryption process can be seen as the inverse of the encryption process.

In decryption process, we must have six secret keys: $g_0$, $\mu_0$, $g_1$, $\mu_1$, $key1$, $key2$. First read the RGB image and then split its component R, G, B, and use Logistic chaotic sequence generated by $g_1$ and $\mu_1$ to change the pixel values of R, G, B, components, and encode the permutated matrix according to DNA rule selected by $key2$. In the control of Logistic chaos sequence

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>This paper</th>
<th>Ref. [16]</th>
<th>Ref. [17]</th>
<th>Ref. [12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color image</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Security analysis</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Biological experiment</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
</tbody>
</table>
generated by $g_0$ and $\mu_0$, we do complement operation, and do subtract to three DNA sequence matrixes, then decode these gray images according to DNA rule selected by $key1$. At last, we recover RGB image and that is the decrypted image.

4. Simulate results

In MATLAB 7.1 environment, we implement simulate experiment for the $128 \times 128$ RGB image Lena in the condition of $g_0 = 0.501, \mu_0 = 3.81, g_1 = 0.401, \mu_1 = 3.68, key1 = 1, key2 = 7$. From the Fig. 4, we can find that the proposed algorithm has a good effect to encrypt color image.

Compared with traditional image encryption algorithm based on chaos, the security of image encryption algorithm we proposed is determined by the chaotic system and biological knowledge involved with DNA computing, which makes the encryption system proposed by us has the dual security and more difficult to decode.

From the above Table 3 we found that the algorithms of the Refs. [16] and [17] are hard to be realized due to the limitation of their needs for complex biological experiment, and the algorithm of Ref. [12] can not encrypt color image. Only the algorithm in this paper can encrypt color image, do not need biological experiment and has good security, so the algorithm in this paper is superior to other DNA encryption algorithm.

5. The security analysis

5.1. Ability of resisting exhaustive attack

5.1.1. Secret key’s space analysis

A good encryption algorithm should have a large key space to make it resist exhaustively attack effectively. The encryption algorithm consists of six key parameters in this paper, they are $g_0$, $\mu_0$, $g_1$, $\mu_1$, $key1$ and $key2$, where $g_0 \in [0,1]$. 

Fig. 4. The images of simulate experiment. (a) Original image (b) encrypted image (c) decrypted image with error keys (d) decrypted image with correct keys.
key1 and key2 are random number. If the calculation precision is $10^{-14}$, the secret key space is $10^{14} \times 10^{14} \times 10^{14} = 10^{56}$ in addition to random number. So the key space is large enough to resist exhaustive attack [18].

5.1.2. Secret key’s sensitivity analysis

Chaotic sequences have high sensitivity to initial value and rapid diffusibility. In implementation of the algorithm, a slight change of arbitrary parameter in secret key will affects the results of encryption and decryption. In order to test the sensitivity of the secret key, we use slight difference keys to decryption. Fig. 5 is the decrypted image and its histogram where $g_0 = 0.501$ changes to $g_0 = 0.501000000000001$ in (a), $\mu_0 = 3.68$ changes to $\mu_0 = 3.680000000000001$ in (b) and other secret keys are unchanged.

Fig. 5 illustrates that only when the decryption keys and the encryption keys are consistent can we correctly decrypt the image. Otherwise, as long as there are minor differences in the key, you can not correctly extract the original image, and the error decrypted image can not reflect the information of original image. Thus, we can see that the proposed algorithm has the secret key sensitivity, and can resist exhaustive attack effectively.

5.2. Ability of resisting statistical attack

5.2.1. The gray histogram analysis

Fig. 6 is the histograms of Lena and Pepper in the same condition. Through comparison of histograms, we can see that the image histograms change greatly after encryption. Before encryption the image pixels are concentrated, and they are relatively even after encryption. The similarities of two images are reduced, thus it shows that the proposed algorithm has a better ability of resisting statistical attacks [19].
5.2.2. Correlation coefficient analysis

Due to the high correlation between adjacent pixels in original image, we must reduce that in encryption images to resist statistical attack. We randomly select 2000 pairs of adjacent pixels from the original image and encryption image, and then use the following equations to calculate the correlation coefficients [20].

\[
E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i
\]  

(3)

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

(4)

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

(5)

Fig. 6. The histograms of original and encrypted image.

Fig. 7. Horizontal direction correlation of Lena image. (a), (c) and (e) are the correlation of original image in R, G and B, respectively; (b), (d) and (f) are the correlation of encrypted image in R, G and B, respectively.
where \( x \) and \( y \) are grey values between adjacent pixels, \( \text{cov}(x,y) \) is covariance, \( D(x) \) is variance and \( E(x) \) is mathematical expectation.

Fig. 7 shows the horizontal direction correlation of the adjacent pixels in R, G, B components of original and encrypted image. From the contrast diagrams we can discover that the correlation between pixels of original image is much larger than the correlation between pixels of encryption image.

Ref. [21] uses one-time keys and robust chaotic maps to encrypt color image. From Table 4 in follows we can draw the conclusion that the correlation coefficient of encrypted image is close to 0 no matter in horizontal, vertical and diagonal directions, but original image is almost close to 1; the correlation coefficient of encrypted image in this paper is lower than that in the Ref. [21]. From images and data, we can easily find that adjacent pixels of original image have very strong linear correlation, while the correlation between adjacent pixels of encrypted image is very small. It has damaged the linear correlation of original image. Therefore the encrypted algorithm can effectively resist pixel correlation statistical attack [22].

5.3. Information entropy

The information entropy is defined to express the degree of uncertainties in the system [23]. We can also use it to express uncertainties of the image information. The information entropy can measure the distribution of gray value in image. If the uncertainty of image is greater, the entropy is bigger and the decrypting process of the image requires more information too. On the contrary, the more orderly the encrypted image is, the smaller the information entropy is. The information entropy equation is as follows:

\[
H(m) = -\sum_{i=0}^{M-1} p(m_i) \log_2 \frac{1}{p(m_i)}
\]

where \( M \) is the total number of symbols \( m_i \in m \); \( p(m_i) \) represents the probability of occurrence of symbol \( m_i \) and \( \log \) denotes the base 2 logarithm so that the entropy is expressed in bits.

The information entropy of an ideal random image is 8, from Table 5 we can know that information entropy of encryption image is very close to 8 and better than the flower entropies in the Ref. [21]. So this means that the encrypted images are close to a random source and the proposed algorithm is secure against the entropy attack.

5.4. Speed performance

Apart from the security consideration, running speed of the algorithm is also an important aspect for a good encryption algorithm. Because the algorithm only use the operation of comparison, or, complement, the complexity of the proposed algorithm is low (the time complexity of the proposed algorithm \( T(n) = O(n^2) \)) and it is suit to modern encryption technology. The simulator for the proposed scheme is implemented using MATLAB 7.1. Performance was measured on a 2.70 GHz Pentium Dual-Core with 1.96 GB RAM running Windows XP. Simulation results show that the average running speed is 21.03 MB/s for encryption and 22.04 MB/s for decryption. Now, we simulate the proposed algorithm in electronic computer, and the time cost is negligible when we using ultra-large-scale parallel computing power of DNA computing.

6. Conclusion

This paper puts forward a RGB image encryption algorithm based on DNA encoding and chaos map. RGB image has a high pixel correlation in spatial domain, but traditional encryption algorithm is mostly used to process it on the R, G and B layers,
respectively, and it is difficult to eliminate the pixel correlation in space domain. Aiming to the characteristics of RGB image, we use binary DNA encoding to make the mathematical problems into biological problems and introduce the biological knowledge of DNA computing into the proposed algorithm. The proposed algorithm effectively eliminates the pixels correlation of the RGB image in the spatial domain by using DNA addition operation, and in order to increase security, we combine chaotic system to disturb the value of the pixels. So the algorithm security is decided by chaotic system and DNA operation to obtain a certain security, and has the dual security. The simulation experiment and results show that the encryption algorithm is effective, simple to implement, and difficult to eliminate the pixel correlation in space domain. Aiming to the characteristics of RGB image, it is not difficult that using ultra-large-scale parallel computing power of DNA computing to implement the algorithm, the time cost is negligible.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Nos. 31170797, 30870573), Program for Changjiang Scholars and Innovative Research Team in University (No. IRT1109), the Program for Liaoning Innovative Research Team in University (No. LT2011018), the Program for Liaoning Innovative Talents in University (No. LR201003), the Program for Liaoning Innovative Research Team in University (No. LT2011018) and by the Program for Liaoning Science and Technology Research in University (No. LS2010179).

References