Chaotic Image Encryption Algorithm Based on Dynamic Spiral Scrambling Transform and Deoxyribonucleic Acid Encoding Operation

XINGYUAN WANG AND SHENGNAN CHEN
School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China
Corresponding author: Xingyuan Wang (xywang@dlmu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672124, in part by the Password Theory Project of the 13th Five-Year Plan National Cryptography Development Fund under Grant MMJJ20170203, in part by the Liaoning Province Science and Technology Innovation Leading Talents Program Project under Grant XLYC1802013, and in part by the Key Research and Development Projects of Liaoning Province under Grant 2019020105-JH2/103.

ABSTRACT This article proposes a dynamic spiral scrambling algorithm, which is combined with the random pixel value filling operation, Deoxyribonucleic Acid (DNA) operation and used in image encryption. The random pixel value filling operation is to generate some random values and fill them around the plaintext image. These values can influence all the pixels after the confusion and diffusion operations. The proposed dynamic spiral scrambling algorithm dynamically combines the chaotic sequences with the plaintext image, and scrambles the pixel values of the plaintext image, this operation makes the scrambling algorithm not fixed, and position changes are highly sensitive to the chaotic sequences, any slight change will get a completely different result. Then, this article combines Deoxyribonucleic Acid (DNA) coding and manipulation to further confuse pixel values. Finally, the average entropy is 7.9974, the average NPCR is 99.6067 and the average UACI is 33.4544, experimental results and various security analyses show that the algorithm has good encryption effects and can resist common attacks such as select plaintext attacks, cropping, and noise attacks.

INDEX TERMS Chaotic image encryption, chaotic system, Deoxyribonucleic acid coding, dynamic spiral scrambling transform.

I. INTRODUCTION Nowadays, with the rapid development of internet technology and computer technology, the communication on the network is becoming more and more frequent. At present, it has become one of the most important means for humans to express information. As images are more and more widely used, the security of information transmission is receiving more and more attention, digital image encryption has become a hot topic for many scholars. The initial value sensitivity of chaotic systems, characteristics of pseudo-randomness, ergodicity, parameter sensitivity and unpredictability [1], which can be well combined with image encryption, therefore, the combination of chaotic systems and image encryption to form an encryption algorithm based on chaotic image cryptosystems has become one of the hot topics in the field of computer science [2]–[9]. Since the emergence of chaos theory, people’s research on chaotic systems has gradually increased from low-dimensional to high-dimensional, and has changed from a static model to a dynamic model. In recent years, spatiotemporal chaos has been favored by scholars for its complex dynamic behavior. A lot of researches have been done on one-dimensional chaotic map, like Logistic chaotic map, Tent chaotic map, two-dimensional chaotic map, like Cat chaotic map, three-dimensional chaotic map, like Lorenz chaotic map, and spatiotemporal chaotic system, like Logistic-Dynamics Coupled Map Lattices (LDCML) spatiotemporal chaotic system [10], all indicate that cryptography based on chaos theory has become a popular research direction for researchers [11]–[14]. With the deepening of scholars’ research, many chaotic systems and image encryption algorithms based on chaos theory have been proposed and achieved good results [15]–[23].
The encryption algorithm is mainly composed of two stages of scrambling and diffusion. In short, “scrambling” changes the position of the pixel value, let it is not in the original position, diffusion is divided into pixel-level and bit-level diffusion, through some specific methods, like parallel diffusion method, diffusion using matrix semi-tensor product, and so on [24]–[28], the pixel value of the plaintext image is changed. An image encryption scheme based on a combination of hybrid genetic algorithms and Deoxyribonucleic Acid (DNA) sequences has broadened scholars’ horizons, through DNA encoding, decoding and pairing rules [29]–[31], the diffusion effect of pixel value is better, and by blurring the information in the original plaintext further can enhances the security of the image.

This article combines two-dimensional-logical adjusted sine map (2D-LASM), one-dimensional Logistic chaotic map, DNA coding method, sorting scrambling and random pixel value addition, hash function, a dynamic spiral scrambling transform is proposed to encrypt and decrypt images. This article uses 2D-LASM and one-dimensional Logistic chaotic map can quickly generate chaotic sequences for encryption and decryption. After selecting appropriate parameters and initial values, chaotic maps can generate random chaotic sequences with good performance. The generated chaotic sequences are extremely sensitive to initial state and system parameters.

The definition of a Logistic chaotic map is:

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

The definition of a 2D-LASM [16] is:

$$\begin{align*}
g : y_{i+1} &= \sin(\pi \mu (z_{i+1} + 3) y_i (1 - y_i)), \\
h : z_{i+1} &= \sin(\pi \mu (y_{i+1} + 3) z_i (1 - z_i)).
\end{align*}$$  \hspace{1cm} (2)

Among them, the control parameter $\mu$ is between $(0, 4)$ and $x_n$ is between $(0, 1)$. A lot of previous studies have shown that in the one-dimensional Logistic chaotic map, when $\mu \in (3.5699456, 4]$, the Logistic chaotic map exhibits chaotic state, in order to achieve chaotic state, in practical application, the value range of $\mu$ should in $(3.5699456, 4]$. The generated chaotic sequence is aperiodic and non-converged, which is very sensitive to the initial value. To avoid the period window, the $\mu$ used in this article is in $(3.89, 4]$. Another chaotic map is 2D-LASM, for details, refer to “(2)”, when $\mu_1 \in [0.37, 0.38]$ or $[0.4, 0.42]$ or $[0.44, 0.93]$ or $\mu_1 = 1$, 2D-LASM enters chaotic state. In this article, we set $\mu_1 = 0.7582$.

B. SPIRAL SCRAMBLING TRANSFORMATION

The essence of the algorithm is to combine the chaotic sequences to change the pixel values position of the plaintext images. The general process of the algorithm is as follows. Firstly, the pixel value is selected according to the chaotic sequences generated by the chaotic maps. The selected pixel values are random. In a round of loops, each time a pixel value of a random position of the original plain image is selected, it is sequentially stored in a new one-dimensional matrix. After the end of the first round of the loop, the panning operation is performed, and then the second loop of the second loop is started, the operation is the same as the first round until all the pixel values of the original plaintext image are selected, finally, reduce it to a two-dimensional matrix of $m \times n$. The algorithm is described as follows:

For plain text image $P[m, n]$:

Step 1. Perform random pixel value filling operation on the original image to prepare for the next “spiral scrambling transform”. The matrix size after the operation is $M \times N$, the added value will affect all the ciphertext after the subsequent scrambling and diffusion operation, it can improve the algorithm security. If the original image is a non-square image, the shorter side is padded with random pixel values. In order to change the position of all pixel values at once, the operation further changes $M$ and $N$ into odd numbers, and transforms them into square images with odd edges.

Step 2. Select the pixel value of the middle point. Refer to “(3)” and “(4)”, take the pixel value at position $i, j$, that is $P(i, j)$. $M$ and $N$ are the length and width of the image after
performing the “pixel value filling operation”.

\[ i = \text{ceil} \left( \frac{M}{2} \right) \]  (3)

\[ j = \text{ceil} \left( \frac{N}{2} \right) \]  (4)

**Step 3.** According to the chaotic sequence \( A_1 \), the layers \( i_1 \) to be selected next is determined.

**Step 4.** Determining the entrance \( i_2 \) at which the pixel value selection operation starts according to another chaotic sequence \( A_2 \).

**Step 5.** According to the binary bits 0 and 1 obtained by the chaotic sequence \( A_3 \), it is determined whether the direction of the selected pixel value is clockwise or counterclockwise.

**Step 6.** After \( i_1, i_2 \) and \( i_3 \) are determined, the pixel value is selected from the original matrix in a spiral manner.

**Step 7.** After the first round of the loop, the panning operation is started.

**Step 8.** Enter the second round of the cycle, the operation is the same as the first round.

**Step 9.** Each time a pixel value is selected, it is stored in a one-dimensional matrix \( B \).

After the first spiral scrambling transformation operation is completed, the matrix \( B \) is obtained, and then the matrix \( B \) is sorted and scrambled according to the chaotic sequences, further confusing the pixel values. The semi-ciphertext image \( P_3 \) is obtained after the sorting and scrambling operation is completed.

**C. DNA CODING METHOD AND OPERATION**

Different DNA sequences have different nucleotides [32], [33]. Nucleotides contain four bases: A (adenine), G (guanine), C (cytosine), T (thymine). The bases are paired in pairs, A (adenine) and T (thymine) are complementary, G (guanine) and C (cytosine) are complementary. In the binary code, there are only 0 and 1, 0 and 1 are complementary, 00 and 11 are complementary, 01 and 10 are complementary. Therefore, A, G, C, and T can be used to encode 00, 11, 01, and 10 in binary. There is a total of 24 coding schemes, however, due to the complementary rules, only 8 encoding schemes are available. The eight coding schemes are shown in Table 1.

**TABLE 1. DNA coding rules.**

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

In the process of encrypting the image, the original pixel gray value is converted into an 8-bit binary number, and the gray value of each 8-bit pixel can be represented by four encoded DNA sequences. For example, a pixel grayscale value of 201, a binary sequence representation of 11001001, encoding it according to the DNA encoding rules on the table, can result in eight combinations: TACG, TAGC, ATCG, ATGC, CGTA, CGAT, GCTA, and GCAT. In addition, the encryption algorithm also performs addition, subtraction, and XOR operations on the base pairs of DNA. Addition, subtraction, and XOR operations are listed in Table 2 to Table 4, respectively.

**TABLE 2. Addition of DNA.**

|   | A | G | C | T |
|---|---|---|---|---|
| A | A | G | C | T |
| G | G | C | T | A |
| C | C | T | A | G |
| T | T | A | G | C |

**TABLE 3. Subtraction of DNA.**

|   | A | G | C | T |
|---|---|---|---|---|
| A | A | T | C | G |
| G | G | A | T | C |
| C | C | G | A | T |
| T | T | C | G | A |

**TABLE 4. XOR operation of DNA.**

|   | A | G | C | T |
|---|---|---|---|---|
| A | A | T | C | G |
| G | G | A | T | C |
| C | C | G | A | T |
| T | T | C | G | A |

**III. ENCRYPTION ALGORITHM**

The encryption algorithm proposed in this article consists of three parts. Firstly, the plain image is first scrambled by using the dynamic spiral scrambling transform. Then, according to the chaotic sequences, it is sorted and scrambled again, and the pixel values are further confused. Finally, the encoding, decoding, addition, subtraction and XOR operation of DNA combined with the chaotic sequences to change the pixel values, these operations can make the system have better encryption effect and higher security. Assume that for a plaintext image \( P \) of size \( m \times n \), where \( m \) is the number of rows and \( n \) is the number of columns. The specific process of encryption is divided into the following steps.

**A. KEY AND SYSTEM INITIAL VALUE GENERATION**

The hash function is irreversible and anti-collision, can effectively resist known plaintext attacks, select plaintext/ciphertext attacks, so in this article, the Secure Hash Algorithm-512(SHA-512) is selected [34], [35], the SHA-512...
algorithm connects the plaintext and the ciphertext, it can better guarantee the encryption effect and security performance of the system, so that the initial key \( K \) is not static, but is constantly changing.

In order to improve the security of the system, a combination of two chaotic maps is used to generate the chaotic sequences. These chaotic sequences are used 3 times, and the total number of used sequences is 8 groups. The initial key \( K \) is converted into a decimal between 0 and 0.5 to obtain a key, bring it into the 2D-LASM to get \( x_0 \), taking the other value in the initial key \( K \) as \( \mu_0 \), and the initial value \( \mu \) of the one-dimensional Logistic is obtained by taking equation, for details, refer to “(5)”. Iterate 200 times to obtain the first chaotic sequence \( A_1 \), use the last value of \( A_1 \) as the initial value of the next iteration, and continue the iterative operation until all the chaotic sequences required by the system are obtained. The chaotic maps used in this article are detailed in equation, refer to “(1)” and “(2)”.

\[
\mu = 3.89 + 0.11\mu_0 \quad (5)
\]

B. MATRIX SCRAMBLING OPERATION

In this article, the scrambling operation of the matrix is divided into two parts. Firstly, the plain image is scrambled by the dynamic spiral scrambling transform to obtain a matrix \( B \), and then the matrix \( B \) is sorted and scrambled with a chaotic sequence to further confuse the pixel values.

1) DYNAMIC SPIRAL SCRAMBLING TRANSFORM

It can be seen from the foregoing operation that if the original image is a “non-square odd edge” matrix, it is operated to become a “square odd edge” matrix, so that the dynamic spiral operation algorithm can change the position of all pixel values in the matrix at once. The specific scrambling operation steps are as follows:

**Step 1.** Starting from 1, a one-dimensional matrix \( h \) is generated in order, where the value is \((1, i−1)\), and the elements in the one-dimensional matrix are sorted and scrambled according to the chaotic sequence \( A_6 \). The value in the one-dimensional matrix \( h_1 \) is used to determine the number of layers \( i \) selected by the spiral. The value of \( i \) is shown in the formula, refer to “(3)”

\[
h_1 = \text{sort}(h) \quad (6)
\]

**Step 2.** After determining the number of layers, the chaotic sequence \( A_7 \) is rounded, and is controlled between \([1, 4]\) to select the entrance for starting the pixel value selection, the upper, left, lower, and right, entrances are numbered as 1, 2, 3, 4 respectively.

\[
a_7(i) = \text{ceil}(A_7(i) \times 4) \quad (7)
\]

**Step 3.** After determining the selected entrance, according to the chaotic sequence \( A_5 \), it is controlled within the range of \([0, 255]\), and then converted into a binary number, according to 0 and 1 to determine the direction of pixel value selection, this article stipulates that 0 represents select the pixel value clockwise, and 1 to select the pixel value in the counterclockwise direction.

\[
A_5 = \text{de2bi}(\text{uint8}(\text{ceil}(A_5 \times 255)), 8) \quad (8)
\]

**Step 4.** A simple example of the loop operation is as follows. From the middle to the outside, set the number of layers to 0, 1, 2, 3××, first select the pixel value of the innermost layer, that is, the 0th layer, and store it in the first position of the one-dimensional matrix \( B \), then select the number of layers \( i_1 \) according to the \( A_6 \) sorted by the chaotic sequence, assuming that \( i_1 \) is 2, that is, selecting the elements of the second layer, and then selecting the entrance \( i_2 \) according to \( A_7 \), assuming that \( i_2 \) is 4 here. Finally, the direction \( i_3 \) of the selected element is determined according to \( A_7 \), assuming that \( i_3 \) is 0 here, that is, clockwise selection. After selecting one layer, the next pixel value selection operation is started until each value in \( A_6 \) is traversed, and the first round of the loop ends. The \( 7 \times 7 \) two-dimensional matrix is taken as an example. The selection operation is as shown in the Fig. 1. In this figure, the innermost green color is counted as the 0th layer, and the outward yellow grid is counted as the first layer. Blue is the second layer, brown is the third layer... and so on.

1. Assume that \( i_1 \) is 2, \( i_2 \) is 4, and \( i_3 \) is 0.
2. Assume that \( i_1 \) is 3, \( i_2 \) is 3, and \( i_3 \) is 1.
3. Assume that \( i_1 \) is 1, \( i_2 \) is 2, and \( i_3 \) is 0. The operation of selecting pixel values is as FIGURE 1.

**Step 5.** After the end of the first loop, the remaining pixel values are transformed. Careful attention can be paid to the FIGURE 2. The blank area in the upper left corner is exactly the same as the shape of the blue selected area in the lower right corner. The pixel value of the blank area in the upper left corner is shifted to the blue area to cover the original pixel value, and the pixel value of the blank area in the lower left corner is translated to the orange area to cover the original pixel value, and the pixel value of the blank area in the upper right corner is shifted to the green area. Overwrite the original pixel value, and then shift the pixel value of the blank area in
the lower right corner to the upper left corner to cover the original pixel value.

**Step 6.** After the end of the panning operation, the operation of selecting the pixel values in the step 2, step 3, step 4, is performed again.

**Step 7.** The gray value of the selected pixel is stored in the one-dimensional matrix \( B \) in order, and the initial confusion of the plaintext image is completed, and finally the semi-ciphertext image \( P_2 \) of \( M \times N \) is obtained.

2) **SORT SCRAMBLING OPERATION**

Next, the semi-ciphertext image \( P_2 \) is further confused by sorting and scrambling. The chaotic sequence \( A_2 \) sorts and scrambles the semi-ciphertext image \( P_2 \) to obtain \( P_3 \).

\[
A_2' = \text{sort}(A_2)
\]

First convert image \( P_2 \) into a one-dimensional sequence \( P_2' = (P_1, P_2, P_3, \ldots, P_M \times N) \).

\( \text{Sort}() \) is a sorting function that sorts \( M \times N \) decimals in the chaotic sequence \( A_2 \) from small to large. After the sorting is completed, the sorted position is obtained. The obtained positions are arranged, and the corresponding subscript of each element in \( A_2' \) in \( A_2 \) is stored as the sequence \( D \). It is easy to know that \( D \) is an unordered scramble of the integer sequence from 1 to \( M \times N \). The elements in \( P_2' \) are established from the first element \( P_1 \) to \( P_{Di} \), from the first to the end, and the positions are exchanged one by one, and then converted into the \( M \times N \) image \( P_3 \). So far, all the scrambling parts are completed. Figure 4 simply shows the steps of sorting scrambling.

![FIGURE 3. Matrix before and after sort scrambling transform: (a) is the matrix before transform; (b) is the matrix after transform.](image)

**C. DNA REPLACEMENT AND DNA MANIPULATION**

The scrambling operation of transforming only the pixel value position is not sufficient to ensure the security of the encryption. The statistical analysis and other analysis methods can still obtain a lot of effective information in plaintext, in order to crack the encrypted image, so it is necessary to diffuse the pixel values. Changing the pixel values in the image can more effectively hide the plaintext information and better resist attacks, this article uses DNA encoding, decoding, addition, subtraction and XOR operations to solve the problem.

The DNA manipulation used in this article is divided into the following steps: First, the original image is encoded according to the chaotic sequence, and the DNA sequence \( C_1 \) is obtained, then the other two chaotic sequences are encoded to generate two other DNA sequences \( C_2 \) and \( C_3 \). The DNA sequences \( C_1, C_2 \) and \( C_3 \) complete a series of addition, subtraction and XOR operations to obtain the DNA sequence \( Z_1 \) to be decoded, finally decode the processed DNA sequence to complete the replacement process \([30], [34], [35]\). The specific operation process is as follows.

According to the chaotic sequence \( A_1 \) generated in the foregoing, the original image is encoded, the DNA sequence \( Z_1 \) after the operation is decoded according to \( A_2 \), the DNA sequence to be operated is generated according to \( A_3 \) and \( A_4 \), and the coding rule is generated according to the formula and used for DNA operation, the formula refer to ‘‘(10)’’. The sequences \( e_1, e_2, e_3 \) and the sequence \( d_1 \) used for decoding have values ranging from 0-7 to select which of the eight encoding methods of DNA.

\[
\begin{align*}
  e_1 &= \text{mod}(\text{uint8}(255 \times A_1), 8) \\
  e_2 &= \text{mod}(\text{uint8}(255 \times A_2), 8) \\
  e_3 &= \text{mod}(\text{uint8}(255 \times A_3), 8)
\end{align*}
\]
\[ d_1 = \text{mod}(\text{uint8}(255 \times A_4), 8) \]  

(11)

Converting each pixel value of the scrambled image \( P_3 \) into binary, each original pixel value corresponds to the converted eight-bit binary bit, then encoding \( P_3 \) according to \( e_1 \), and the encoded semi-ciphertext DNA sequence \( C_1 \) is obtained.

Chaotic sequences \( A_3 \) and \( A_4 \) are processed according to the formula, refer to \( "(10)-(11)" \), and their values range from 0 to 255. They are also converted into octet binary values, then encoded \( A_3 \) and \( A_4 \) according to the chaotic sequences \( e_2 \) and \( e_3 \) to obtain DNA sequences \( C_2 \) and \( C_3 \).

After the DNA sequences \( C_1 \), \( C_2 \) and \( C_3 \) are obtained, the addition, subtraction and XOR operations of the DNA are started. There is a total of 9 possible operation between the three DNA sequences, see the details, refer to \( "(13)" \). The chaotic sequence \( A_5 \times 9 \) is rounded up to obtain \( a_5 \), and its value is between 1 to 9, which is used to select the 9 different situations, each operation corresponding to different \( e_1 \), \( e_2 \), \( e_3 \), according to \( e_1 \), \( e_2 \), \( e_3 \) to determine different sequence rules. Addition, subtraction and XOR operation results in the result \( Z_1 \).

\[ a_5 = \text{ceil}(A_5 \times 9) \]  

(12)

The nine addition, subtraction, and XOR operation rules in DNA manipulation are as follows.

\[
\begin{align*}
Z_1(i,j) &= C_1 + C_2 + C_3 \\
Z_1(i,j) &= C_1 + C_2 - C_3 \\
Z_1(i,j) &= C_1 + C_2 \oplus C_3 \\
Z_1(i,j) &= C_1 - C_2 - C_3 \\
Z_1(i,j) &= C_1 - C_2 + C_3 \\
Z_1(i,j) &= C_1 - C_2 \oplus C_3 \\
Z_1(i,j) &= C_1 \oplus C_2 + C_3 \\
Z_1(i,j) &= C_1 \oplus C_2 - C_3 \\
Z_1(i,j) &= C_1 \oplus C_2 \oplus C_3
\end{align*}
\]  

(13)

Finally, \( Z_1 \) is decoded according to \( d_4 \), and the decoded binary sequence is obtained, converted into a decimal pixel value, and then restored to an image \( P_4 \) of \( M \times N \) size.

At this point, the encryption process is all over. The decryption process is the reverse process of the encryption process, similar to the encryption process, and will not be described here. The encryption process is as FIGURE 5.
IV. EXPERIMENTAL RESULTS AND SAFETY ANALYSIS

A. EXPERIMENTAL RESULTS

The plaintext images selected in this article are five images of “Lenna”, “Plane”, “Pepper”, “Black” and “White”, the size of the image is 256 × 256. FIGURE 6 - 10 gives the plaintext images, the encrypted images and the decryption images of the above five grayscale images.

B. SECURITY ANALYSIS

1) KEY SECURITY ANALYSIS

The initial key $K$ generated by SHA-512 in this article is 128-bit char type data, and the generated key is related to the plaintext image, then a series of operations on the key $K$ are performed to generate the initial value of chaotic systems, the one-dimensional Logistic chaotic map and 2D-LASM.
are iterated 200 times to make the system reach a state of full chaos, so as to obtain the key used for encryption. The key space of the algorithm is large enough and has a high correlation with the plaintext image. Even with a very small change to the initial key, the final result can change a lot. The following is a test of key sensitivity, after the image is encrypted, it is decrypted with the correct key, the correct original plaintext image can be obtained, only the first bit of the key is changed very slightly, by decrypting the ciphertext image, the correct original plaintext image cannot be obtained. FIGURE 11 shows the decryption test results of the correct key and the wrong key respectively. The experimental results show that although the key change is small, the result obtained by the decryption algorithm is completely different. The initial key is extremely sensitive. Let's take “Lenna” as an example:

Correct key K:
7290ac041911f857e18384f87333337c0258057723bb56a
dfad3549ad3bbe5d7f70066beccfa8dcd20fb22d2c6188f1a24c4b766be65bf729e05646462ba349e

Error key K’:
6290ac041911f857e18384f87333337c0258057723bb56a
dfad3549ad3bbe5d7f70066beccfa8dcd20fb22d2c6188f1a24c4b766be65bf729e05646462ba349e

b: ADJACENT PIXEL CORRELATION
In the unencrypted plaintext image, there is a strong correlation between adjacent pixels, which makes the security performance insufficient. In order to resist the attack of statistical analysis, the correlation between adjacent pixels must be reduced. The following formula is used to calculate the correlation between the plain image and the adjacent pixels of the ciphertext image, refer to “(14)-(17)”. In this article, 3000 pairs of pixels are selected in the plaintext image and ciphertext image of “Lenna”, the results are compared in three related directions: horizontal, vertical and diagonal. FIGURE 14 and FIGURE 15 show the correlation between adjacent pixels in the horizontal direction, vertical direction and diagonal direction of the “Lenna” image before and after encryption. It can be seen that the image before encrypting has a strong correlation in three directions, but that the pixel value distribution of the ciphertext image encrypted by the encryption algorithm in three directions is very uniform, and the correlation is greatly reduced.

\[ r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)}} \] (14)

2) VARIOUS STATISTICAL ANALYSIS
After the image encryption program is completed, it is very important and necessary to perform statistical analysis. When cracking the ciphertext image, the attacker can sometimes use the statistical analysis method to find the ciphertext digital image, statistical anomaly, revealing the existence of hidden information, in order to achieve the purpose of cracking the encrypted image, in which the analysis of image histogram and the analysis of adjacent pixel correlation are two very important statistical characteristics in image encryption algorithm indicators, so we need to analyze them.

a: HISTOGRAM AND ITS ANALYSIS
The horizontal axis of the histogram represents the pixel value, the vertical axis represents the sum of the number of the pixel values in the figure, the distribution of pixel values in plaintext images is usually uneven. The ciphertext image processed by the encryption algorithm, the more uniform the distribution of pixel values, the better the performance against statistical analysis attacks, encrypted images are more secure. As shown in FIGURE 12 - 13, the three images of “Lenna”, “Plane” and “Pepper” are displayed before and after encryption, and the encrypted three histograms can be seen. The histogram distribution is uneven before encryption, the histograms of the three images after encryption are uniform.
Information entropy is an abstract concept in mathematics, which reflects the degree of ordering of information. The information entropy of a system is higher, it means that the system is more disordered. Conversely, the information entropy of a system is lower, it means that the system is more orderly. The value of information entropy can be calculated by the equation, refer to “(18)”.

\[
H(s) = \sum_{i=0}^{2^L-1} p(s_i) \log_2 \frac{1}{p(s_i)}
\]

where \(p(s_i)\) represents the probability of occurrence. The closer the information entropy is to 8, the more disordered the pixel values of the encrypted image are, and the better the encryption effect is. Table 7 shows the information entropy of the three images “Lenna”, “Plane” and “Pepper”. The experimental results show that the information entropy of the image encrypted by the encryption algorithm is closer to 8, which can better resist statistical attacks.

3) RESISTANCE DIFFERENTIAL ATTACK ANALYSIS

Differential attacks are a choice of chosen plaintext attacks. The encryption algorithm can be attacked by comparing the
X. Wang, S. Chen: Chaotic Image Encryption Algorithm Based on Dynamic Spiral Scrambling Transform

FIGURE 15. Correlation between adjacent pixels of the image after encryption: (a) is the horizontal direction of ciphertext; (b) is the vertical direction of ciphertext; (c) is the diagonal direction of ciphertext.

TABLE 5. Correlation coefficients of images.

| Image     | Plain | Proposed |
|-----------|-------|----------|
|           | Level | Vertical | Diagonal | Level | Vertical | Diagonal |
| Lena      | 0.9538 | 0.9721 | 0.9375 | −0.0017 | −0.0132 | 0.0084 |
| Plane     | 0.9078 | 0.9106 | 0.8405 | −0.0043 | −0.0154 | −0.0257 |
| Pepper    | 0.9668 | 0.9729 | 0.9444 | −0.0062 | −0.0236 | −0.0047 |
| Image-0   | 1      | 1      | 1      | 0.0104  | −0.0260 | −0.0099 |
| Image-255 | 1      | 1      | 1      | −0.0020 | −0.0118 | −0.0037 |

TABLE 6. Comparison of image correlation coefficients.

| Image     | Proposed | Ref. [37] | Ref. [33] |
|-----------|----------|-----------|-----------|
|           | Level | Vertical | Diagonal | Level | Vertical | Diagonal | Level | Vertical | Diagonal |
| Lena      | −0.0017 | −0.0132 | 0.0084  | 0.0003 | 0.0025  | 0.0004  | 0.0047 | 0.0028  | 0.0043  |
| Plane     | −0.0043 | −0.0154 | −0.0257 | 0.0002 | 0.0049  | 0.0008  | 0.0023 | 0.0060  | 0.0069  |
| Pepper    | −0.0062 | −0.0236 | −0.0047 | 0.0004 | 0.0064  | 0.0025  | 0.0038 | 0.0057  | 0.0112  |

TABLE 7. Information entropy of images.

| Image     | Plain | Proposed | Ref. [5] | Ref. [9] | Ref. [30] |
|-----------|-------|----------|----------|----------|-----------|
| Lena      | 7.3218 | 7.9972  | 7.9967  | 7.9964  | 7.9972   |
| Plane     | 6.8421 | 7.9974  | 7.9974  | 7.9945  | 7.9970   |
| Pepper    | 7.3800 | 7.9975  | 7.9973  | 7.9913  | 7.9971   |

TABLE 8. Local information entropy of images.

|          | Proposed | Lena | Plane | Pepper | Object | House |
|----------|----------|------|-------|--------|--------|-------|
| Local information entropy | 7.9019 | 7.9029 | 7.9020 | 7.9021 | 7.9028 |
| Pass or Fail | pass | pass | pass | pass | pass |

The propagation of the plaintext image after the encryption. The minimal change is made to the plaintext, so that the plaintext image before encryption and the ciphertext image after the encryption operation are obtained, and then the acquired data of the two images before and after the encryption are analyzed to obtain the key. Therefore, a good encryption algorithm should be able to resist differential attack analysis, that is, even if the change of plaintext is extremely small, the ciphertext will be greatly changed, so that it can resist the differential attack well. Here, the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI) are calculated. The detail is in equation, refer to “(19)”.

\[
NPCR(C_1, C_2) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} D(i,j)}{M \times N} \times 100\%
\]

\[
UACI = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} |c_1(i,j) - c_2(i,j)|}{M \times N \times 255} \times 100\% 
\]

where \(C_1, C_2\) represent two different images. \(M \times N\) represents the size of image. If \(C_1(i,j) \neq C_2(i,j)\), then \(D(i,j) = 1\), otherwise, \(D(i,j) = 0\). In addition, Wu et al. establish a mathematical model for ideal encryption images and propose statistical hypothesis of NPCR and UACI [36]. The hypothesis of NPCR with \(\alpha\)-level significance is

\[
H_0 : NPCR(C_1, C_2) = \mu_N
\]

\[
H_1 : NPCR(C_1, C_2) < \mu_N
\]
TABLE 9. NPCR values of ciphertext images and its comparison.

| File name | Our algorithm | Ref. [40] | Ref. [44] | Ref. [45] | Ref. [13] | Ref. [46] |
|-----------|---------------|-----------|-----------|-----------|-----------|-----------|
| 5.1.09    | 99.6003       | 99.5804   | 99.614    | 99.6094   | 99.6246   | 99.5575   |
| 5.1.10    | 99.5882       | 99.5865   | 99.588    | 99.5956   | 0.0092    | 99.5544   |
| 5.1.11    | 99.6033       | 99.5972   | 99.6033   | 99.6109   | 99.6445   | 99.8123   |
| 5.1.12    | 99.6215       | 99.6201   | 99.5651   | 99.6063   | 99.5972   | 99.6109   |
| 5.1.13    | 99.6457       | 99.6414   | 99.5789   | 99.6140   | 99.6582   | 99.7421   |
| 5.1.14    | 99.5927       | 99.5773   | 99.6765   | 99.5987   | 99.6993   |
| 5.2.08    | 99.6090       | 99.6300   | 99.6037   | 99.6101   | 99.6216   | 99.6101   |
| 5.2.09    | 99.6329       | 99.6346   | 99.6029   | 99.6019   | 99.6048   | 99.7025   |
| 5.2.10    | 99.6120       | 99.6178   | 99.6124   | 99.5979   | 99.5861   | 99.612    |
| 7.1.01    | 99.6052       | 99.5861   | 99.6082   | 99.5842   | 99.6162   | 99.419    |
| 7.1.02    | 99.6037       | 99.6178   | 99.6174   | 99.6117   | 99.6025   | 99.72     |
| 7.1.03    | 99.6257       | 99.6117   | 99.612    | 99.6021   | 99.5998   | 99.4072   |
| 7.1.04    | 99.6204       | 99.5808   | 99.5911   | 99.6201   | 99.6033   | 99.6037   |
| 7.1.05    | 99.6151       | 99.5998   | 99.6178   | 99.6185   | 99.6070   | 99.4572   |
| 7.1.06    | 99.6120       | 99.6006   | 99.6174   | 99.6346   | 99.6015   | 99.5213   |
| 7.1.07    | 99.5942       | 99.6059   | 99.5922   | 99.5926   | 99.6029   | 99.5007   |
| 7.1.08    | 99.6556       | 99.5918   | 99.6052   | 99.6223   | 99.612    | 99.6902   |
| 7.1.09    | 99.6120       | 99.6010   | 99.6086   | 99.6166   | 99.6048   | 99.7181   |
| 7.1.10    | 99.5980       | 99.6002   | 99.5941   | 99.6086   | 99.6212   | 99.5163   |
| boat.512  | 99.6037       | 99.6037   | 99.6101   | 99.5800   | 99.6067   | 99.7128   |
| gray.512  | 99.6098       | 99.6075   | 99.6159   | 99.6235   | 993.6094  | 99.612    |
| ruler.512 | 99.6329       | 99.6147   | 99.6212   | 99.6132   | 99.6113   | 99.3118   |

| pass rate | 100%          | 90.91%      | 95.45%      | 90.91%      | 90.91%      | 59.09%     |

where $\mu_N = \frac{F}{F + 1}$ and it is the ideal value of NPCR. When NPCR ($C_1, C_2$) < NPCR*, we reject $H_0$, otherwise, we accept $H_0$. The critical value NPCR* is defined as Eq. (21).

$$NPCCR^*_\alpha = \frac{(F - \Phi^{-1}(\alpha)\sqrt{\frac{F}{MN}})}{F + 1}$$

where $F$ denotes the largest supported pixel value, and $M \times N$ is the size of image. $\Phi^{-1}(\alpha)$ is the inverse cumulative density function of the standard normal distribution. On the other hand, the hypothesis of UACI with $\alpha$-level significance is

$$H_0 : UACI(C_1, C_2) = \mu_U$$
$$H_1 : UACI(C_1, C_2) < \mu_U$$

where $\mu_U = \frac{F(F + 2)}{3F + 3}$ and it is the ideal value of UACI. When UACI ($C_1, C_2$) $\neq$ (UACI*-, UACI*+), we reject $H_0$, otherwise, we accept $H_0$. The critical value UACI* and UACI*+ is defined as Eq. (23).

$$UACI'_{\alpha} = \frac{F + 2}{3F + 3} - \Phi^{-1}(\alpha)\sqrt{\frac{(F + 2)(F^2 + 2F + 3)}{18(F + 1)^2MNF}}$$
$$UACI_{\alpha}^+ = \frac{F + 2}{3F + 3} + \Phi^{-1}(\alpha)\sqrt{\frac{(F + 2)(F^2 + 2F + 3)}{18(F + 1)^2MNF}}$$

Here, we set $\alpha = 0.05$. Then, we choose 22 grayscale images from the USC-SIPI database: 6 images of size 256 x 256, 16 images of size 512 x 512, and other commonly used images. For the images in the USC-SIPI database, we calculate the pass rate and compare it with other references, for the images of size 256 x 256, $N_\alpha^* = 99.5693\%$ and $[U_{\alpha}^-, U_{\alpha}^+] = [33.2824\%, 33.6447\%]$; for the images of size 512 x 512, $N_\alpha^* = 99.5839\%$ and $[U_{\alpha}^-, U_{\alpha}^+] = [33.3730\%, 33.5541\%]$. For the other commonly used images, we choose 6 images of size 256 x 256, these images are “Lenna” “Fabio” “Peppers” “Plane” “Black” and “White”, we calculate the average value of NPCR and UACI for all pictures and compare with other references. It can be seen from Table 9 and Table 10 that the pass rate of this article is 100%, while the pass rate of references
TABLE 10. UACI values of ciphertext images and its comparison.

| File name | Our algorithm | Ref. [40] | Ref. [41] | Ref. [42] | Ref. [43] | Ref. [45] |
|-----------|---------------|-----------|-----------|-----------|-----------|-----------|
| 5.1.09    | 33.4948       | 33.5253   | 33.14     | 33.5980   | 33.4652   | 33.5096   |
| 5.1.10    | 33.4250       | 33.3938   | 33.24     | 33.5366   | 33.5448   | 33.5587   |
| 5.1.11    | 33.4839       | 33.8600   | 33.24     | 33.4398   | 33.4162   | 33.4315   |
| 5.1.12    | 33.5738       | 33.6150   | 33.56     | 33.4228   | 33.2152   | 33.5379   |
| 5.1.13    | 33.4366       | 33.7250   | 33.56     | 33.4205   | 33.4699   | 33.6163   |
| 5.1.14    | 33.4855       | 33.4491   | 33.21     | 33.4696   | 33.5554   | 33.4617   |
| 5.2.08    | 33.5399       | 33.3933   | 33.31     | 33.4720   | 33.4575   | 33.4323   |
| 5.2.09    | 33.4489       | 33.5346   | 33.62     | 33.4921   | 33.4175   | 33.3405   |
| 5.2.10    | 33.5227       | 33.5265   | 33.31     | 33.4914   | 33.4315   | 33.4687   |
| 7.1.01    | 33.5386       | 33.4789   | 33.25     | 33.5212   | 33.5150   | 33.5444   |
| 7.1.02    | 33.4802       | 33.5416   | 33.27     | 33.4846   | 33.5221   | 33.4910   |
| 7.1.03    | 33.4531       | 33.4062   | 33.27     | 33.4647   | 33.4777   | 33.5553   |
| 7.1.04    | 33.4818       | 33.4845   | 33.21     | 33.5202   | 33.4721   | 33.4806   |
| 7.1.05    | 33.5400       | 33.4852   | 33.30     | 33.5400   | 33.4757   | 33.3965   |
| 7.1.06    | 33.5435       | 33.4453   | 33.30     | 33.5254   | 33.5035   | 33.5651   |
| 7.1.07    | 33.5397       | 33.4535   | 33.15     | 33.5205   | 33.4317   | 33.4363   |
| 7.1.08    | 33.4855       | 33.4760   | 33.26     | 33.5678   | 33.4274   | 33.4769   |
| 7.1.09    | 33.5173       | 33.4875   | 33.26     | 33.5223   | 33.4452   | 33.4085   |
| boat.512  | 33.4881       | 33.4754   | 33.23     | 33.4325   | 33.4434   | 33.5247   |
| gray.512  | 33.5468       | 33.3743   | 33.37     | 33.3993   | 33.4554   | 33.4770   |
| ruler.512 | 33.5394       | 33.3807   | 33.43     | 33.5129   | 33.4795   | 33.4166   |

is lower than 100%. The comparison of the average values in Table 11 also shows that the algorithm in this article has NPCR and UACI values closer to the ideal value. So the encryption algorithm proposed in this article has better ability to resist differential attacks.

TABLE 11. The average of NPCR and UACI and comparison with other algorithms.

| Image     | #Mean | Ref. [5]     | Ref. [37]     | Ref. [38]     | Ref. [39]     |
|-----------|------|--------------|--------------|--------------|--------------|
| NPCR (%)  | 99.6067 | 99.5900 | 99.6175 | 99.6077 | 99.6137 |
| UACI (%)  | 33.4544 | 33.4000 | 33.4152 | 33.4570 | 33.5489 |

4) RESISTANCE TO CROPPING ATTACKS AND NOISE ATTACKS
In many image encryption systems, the cropping attack and the noise attack need to be tested to verify that the algorithm has robustness against cropping and noise attacks. Because the attacker cannot correctly decrypt the encrypted image during the information transmission process. In this case, the attacker usually uses the means of interfering communication to achieve the effect of preventing the recipient from decrypting, thereby achieving the purpose of the attack.

In the experiment, this article takes “Lenna” as an example, and cuts 1/16, 1/8, 1/4, 1/2 respectively. The image recognition performance after decryption operation is good, and the image after noise attack can also be recognized after decryption. These images can be restored to recognizable plaintext images, the experimental results show that the algorithm is very resistant to cropping attacks and noise attacks. Different levels of cropping and noise attacks and their recovery results are shown in FIGURE 16-19, and FIGURE 20.

5) \( \chi^2 \) TEST RESULTS
\( \chi^2 \) is defined in equation, refer to “(24)”. It is used to quantitatively describe the extent to which an image deviates from the absolutely uniform distribution. Where \( \bar{p} \) represents the average frequency of all pixels and is taken for \( (M \times N)/256 \), \( p_i \) represents frequency of pixel \( i \) in the image. The smaller
the $\chi^2$, the more uniform the pixel distribution. Table 12 lists $\chi^2$ values of all cipher images are much lower than that of plain images.

$$
\chi^2 = \sum_{i=0}^{255} \frac{(p_i - \bar{p})^2}{\sigma} 
$$

(24)

6) PSNR ANALYSIS
The full name of PSNR is Peak Signal to Noise Ratio, which is an evaluation index of the quality of the full reference image. PSNR is often used to calculate the degree of difference between the ciphertext image and the ciphertext image after cropping. Objectively evaluate the degree of error between images based on corresponding pixel values. That is, based on error-sensitive image quality evaluation. The PSNR algorithm is simple and the inspection speed is fast, but the difference value it presents is sometimes not proportional to the subjective feeling of the person. The results of the PSNR are shown in Table 12.

In equation (26), $P_{max}$ is the maximum gray value of the plain text image, and $P_{max} = 255$. $P(i, j)$ is the pixel value of $(i, j)$ after the plain text image is encrypted. $C(i, j)$ is the pixel value of $(i, j)$ after cropping the ciphertext.

As you can see from Table 13, the algorithm proposed in this article has a smaller PSNR. It shows that there is a great difference between plaintext and ciphertext, so the algorithm proposed in this article has higher security.

$$
MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||P(i, j) - C(i, j)||^2
$$

(25)

$$
PSNR = 20 \times \log_{10} \left( \frac{P_{max}}{\sqrt{MSE}} \right)
$$

(26)

7) ENCRYPTION SPEED
The proposed algorithm is implemented in matlab 2017 and deployed on the computer with the Intel Core i5-5200 CPU,
TABLE 14. Encryption speed and comparison.

| Algorithm         | Key generation time (s) | Encryption time (s) | Decryption time (s) | Platform & System characteristics |
|-------------------|-------------------------|---------------------|---------------------|-----------------------------------|
| Our proposed      | 0.028485                | 2.819177            | 2.910840            | Matlab R2017a, CPU 2.7GHZ, 4 GB memory |
| Ref. [37]         | -                       | 2.8716              | -                   | Matlab R2012b, CPU 2.93GHZ, 2 GB memory |
| Ref. [47]         | -                       | 3.5015              | -                   | -                                  |
| Ref. [48]         | -                       | 7.73                | -                   | Matlab R201a, CPU 2.7GHZ, 2 GB memory |
| Ref. [49]         | -                       | 2.9968              | -                   | Matlab 7.1                         |
| Ref. [50]         | -                       | 3.5341              | -                   | Matlab 7.1, CPU 2.7GHZ, 2 GB memory |

4GB memory and Windows 10 operation system. Take the Lena image (256 × 256) as an example. The time to generate the key sequence is 0.028485 seconds, the encryption time is 2.819177 seconds, and the decryption time is 2.910840 seconds. Table 14 shows the key generation time, encryption time, decryption time of this algorithm and its comparison with other references.

8) KEY SPACE
The key space should be large enough to make brute-force attack infeasible. In the proposed algorithm, The initial values $y \in (0, 1)$, $z \in (0, 1)$, the parameter $\mu \in (3.89, 4]$, the parameter $\mu_1 \in [0.37, 0.38] \cup [0.4, 0.42] \cup [0.44, 0.93]$ and $\mu_1 = 1$. Assuming that the system has $10^{-14}$ accuracy, the size of each key can be calculated as follows.

$$(10^{14} \times 10^{14} \times 0.11 \times 10^{14} \times 0.52 \times 10^{14}) = 572 \times 10^{48}$$

(27)

So the key space of this algorithm is greater than $2^{100}$, the key space is large enough to resist brute-force attack.

9) TIME COMPLEXITY
This algorithm will make unconventional images into regular images through “random pixel value filling operation”, so the time complexity of this algorithm is $O(\max(M, N) \times \max(M, N))$, $M$ and $N$ are the length and width of the image respectively.

10) NIST TEST
NIST test includes 15 tests. It can be used to estimate the randomness of a binary sequence. Each test gives a significance level $\alpha$ ($\alpha = 0.001$). The probabilistic value (P-value) is obtained after statistical results. If P-value $\geq \alpha$, the sequence is random. If P-value $< \alpha$, the sequence is non-random. For an ideal encryption algorithm, the P-value of the cipher image should be greater than $\alpha$. Table 15 lists NIST test of plain and cipher image of “Lenna”.

V. COLOR IMAGE PERFORMANCE ANALYSIS
In this chapter, the encryption algorithm of color image is briefly introduced and experimented, and some simple security analysis is also carried out.

A. COLOR IMAGE ENCRYPTION AND DECRYPTION RESULTS
The color image encryption process is actually the same as the gray image process. The color image is decomposed into three gray images of R, G, and B. The three gray images are encrypted separately, and then decrypted. Synthesize the three images, that is, the color image after the encryption is successful. In this article, the “Fruits” are used as an example to perform the encryption and decryption operations. The encryption and decryption results are shown in FIGURE 21-24.
B. COLOR IMAGE PERFORMANCE ANALYSIS

The performance analysis of a color image is to decompose the color image into three images of R, G, and B. After analyzing the performance of these three images, the following is a histogram analysis, correlation analysis, anti-cutter, noise attack analysis of three images of R, G, and B.

1) COLOR IMAGE STATISTICAL ANALYSIS

a: HISTOGRAM ANALYSIS

Here, the color image is decomposed into three images of R, G, and B. The histograms of the R image before and after encryption, the histogram of the G image, and the histogram of the B image are given respectively. It is found that the histograms of the first three images are obviously uneven. The histograms of the cipher text after encryption are obviously even. See details in FIGURE 25 and FIGURE 26.

b: CORRELATION ANALYSIS

Here, the color image is decomposed into three images: R, G, and B. The horizontal correlation of the R image before encryption, the vertical correlation of the G image, and the diagonal correlation of the B image are given. It can be seen that the pixel distribution of the first three images is very compact and the correlation between the pixel values is strong. At the same time, the horizontal correlation of the encrypted image R image, the vertical correlation of the G image, and B are also given. The diagonal correlation of the image, it can be seen from the figure that the...
distribution of pixel values is more uniform, and the correlation between pixel values is significantly reduced. See details in FIGURE 27 and FIGURE 28.

**c: CROPPING AND DECRYPTION RESULTS**

The cropping and noise attack analysis of color images are to perform a cropping or noise attack on the three images R,
TABLE 16. NPCR and UACI for color images.

|     | R               |     | G               |     | B               |     |
|-----|-----------------|-----|-----------------|-----|-----------------|-----|
|     | NPCR            | UACI| NPCR            | UACI| NPCR            | UACI|
| Plain| 99.6429         | 33.5139| 99.5880         | 33.4390| 99.5850         | 33.3688|
|      | 99.5758         | 33.4965| 99.6017         | 33.4868| 99.5941         | 33.4591|

FIGURE 29. Crop 1/8 and its decryption result.

FIGURE 30. 0.05 salt and pepper noise attack and its decryption result.

TABLE 17. $\chi^2$ test of color image.

|         | R   | G   | B   |
|---------|-----|-----|-----|
| Plain   | 33775 | 51415 | 11463 |
| Proposed| 267  | 224  | 294  |

VI. CONCLUSION

This article proposes a dynamic spiral scrambling algorithm based on chaotic system, which can be implemented from special to general. According to different plaintexts, each encryption operation generates different key, and the scrambling operations on the plaintext images are not the same, at the same time, this article combines with the random pixel value filling operation, different random pixel values will be generated during each encryption process, and these random pixel values will affect other pixel values during the encryption process. In theory, this makes the cracking more difficult. The experimental results also show that the algorithm can encrypt and decrypt black and white images, grayscale images and even color images, and the analysis of the experimental results also shows that the algorithm not only has good effects in encryption, but also resists common attacks. It also has good performance. It can prove the performance of the algorithm through key sensitivity test, histogram analysis, correlation analysis and differential attack. At present, the algorithm is still in the theoretical stage. The experimental results and various security analysis show that the algorithm has good encryption effect and can resist common attacks, such as clear-text attack, cropping, noise attack, etc. The algorithm can be used for image encryption.

REFERENCES

[1] L. Kocarev, “Chaos-based cryptography: A brief overview,” IEEE Circuits Syst. Mag., vol. 1, no. 2, pp. 1359–1375, Jan. 2001.
[2] H.-M. Yuan, Y. Liu, L.-H. Gong, and J. Wang, “A new image cryptosystem based on 2D hyper-chaotic system,” Multimedia Tools Appl., vol. 76, no. 6, pp. 8087–8108, Mar. 2017.
[3] X. Li, C. Li, and I.-K. Lee, “Chaotic image encryption using pseudorandom masks and pixel mapping,” Signal Process., vol. 125, pp. 48–63, Aug. 2016.
[4] A.-V. Diaconu, “Circular inter-intra pixels bit-level permutation and chaos-based image encryption,” Inf. Sci., vols. 355–356, pp. 314–327, Aug. 2016.
[5] C. Song and Y. Qiao, “A novel image encryption algorithm based on DNA encoding and spatiotemporal chaos,” Entropy, vol. 17, no. 12, pp. 6954–6968, Oct. 2015.
[6] O. Mirzaei, M. Yaghooob, and H. Irani, “A new image encryption method: Parallel sub-image encryption with hyper chaos,” Nonlinear Dyn., vol. 67, no. 1, pp. 557–566, Jan. 2012.
[7] A. Kanso and M. Ghebleh, “A novel image encryption algorithm based on a 3D chaotic map,” Commun. Nonlinear Sci. Numer. Simul., vol. 17, no. 7, pp. 2943–2959, Jul. 2012.
[8] X.-Y. Wang, L. Yang, R. Liu, and A. Kadir, “A chaotic image encryption algorithm based on perceptron model,” Nonlinear Dyn., vol. 62, no. 3, pp. 615–621, Nov. 2010.
[9] R. Enayatifar, A. H. Abdullah, and I. F. Isin, “Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence,” Opt. Lasers Eng., vol. 56, pp. 83–93, May 2014.
[10] W. Xingyuan, F. Le, W. Shibing, C. Zhang, and Z. Yingjian, “Spatiotemporal chaos in coupled logistic map lattice with dynamic coupling coefficient and its application in image encryption,” IEEE Access, vol. 6, pp. 39705–39724, Jul. 2018.

[11] X.-Y. Wang, S.-X. Gu, and Y.-Q. Zhang, “Novel image encryption algorithm based on cycle shift and chaotic system,” Opt. Lasers Eng., vol. 68, pp. 126–134, May 2015.

[12] X. Wang, L. Liu, and Y. Zhang, “A novel chaotic block image encryption algorithm based on dynamic random growth technique,” Opt. Lasers Eng., vol. 66, pp. 10–18, Mar. 2015.

[13] L. Xu, Z. Li, J. Li, and W. Hua, “A novel bit-level image encryption algorithm based on chaotic maps,” Opt. Lasers Eng., vol. 78, pp. 17–25, Mar. 2016.

[14] L. Wang, H. Song, and P. Liu, “A novel hybrid color image encryption algorithm using two complex chaotic systems,” Opt. Lasers Eng., vol. 77, pp. 118–125, Feb. 2016.

[15] W. Liu, K. Sun, and C. Zhu, “A fast image encryption algorithm based on chaotic map,” Opt. Lasers Eng., vol. 84, pp. 26–36, Sep. 2016.

[16] C.-X. Zhu and K.-H. Sun, “Cryptography and improving a novel color image encryption algorithm using RT-Enhanced chaotic tent maps,” IEEE Access, vol. 6, pp. 18770–188759, Mar. 2018.

[17] C. Chen, K. Sun, and Q. Xu, “A color image encryption algorithm based on 2D-CIMM chaotic map,” China Commun., vol. 17, no. 5, pp. 12–20, May 2020.

[18] C. Chen, K. Sun, and S. He, “An improved image encryption algorithm with finite computing precision,” Signal Process., vol. 168, Mar. 2020, Art. no. 107340.

[19] N. Chidambaram, P. Raj, T. Karuppaswamy, and R. Amirtharajan, “An advanced framework for highly secure and cloud-based storage of colour images,” IET Image Process., Jun. 2020, doi: 10.1049/iet-ipr.2018.5654.

[20] S. R. R. Sundaramaran, J. B. B. Rayappan, and R. Amirtharajan, “Ring oscillator as confusion—Diffusion agent: A complete TRNG driven image security,” IET Image Process., May 2020, doi: 10.1049/iet-ipr.2019.0168.

[21] S. R. S. Rajagopalan, and R. Amirtharajan, “FGPA based generic RO TRNG architecture for image confusion,” Multimedia Tools Appl., vol. 79, nos. 19–20, pp. 13841–13868, May 2020.

[22] C. Lakshmi, K. Thennmozhi, J. B. B. Rayappan, and R. Amirtharajan, “Hopfield attractor-trusted neural network: An attack-resistant image encryption,” Neural Comput. Appl., vol. 32, no. 15, pp. 11477–11489, Aug. 2020.

[23] S. Rajagopalan, S. Sharma, S. Arumugham, H. N. Upadhyay, J. B. B. Rayappan, and R. Amirtharajan, “YRBS coding with logistic map—A novel sanskrit aphorism and chaos for image encryption,” Multimedia Tools Appl., vol. 78, no. 8, pp. 10513–10541, Apr. 2019.

[24] L. Teng and X. Wang, “A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive,” Opt. Commun., vol. 285, nos. 19–20, pp. 4048–4054, Sep. 2012.

[25] X. Wang, L. Feng, and H. Zhao, “Fast image encryption algorithm based on parallel computing system,” Inf. Sci., vol. 486, pp. 340–358, Jun. 2019.

[26] X. Wang and S. Gao, “Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory,” Inf. Sci., vol. 507, pp. 16–36, Jan. 2020.

[27] B. Murugan and A. G. Nanjappa Gounder, “Image encryption scheme based on block-based confusion and multiple levels of diffusion,” IET Comput. Vis., vol. 10, no. 6, pp. 593–602, Sep. 2016.

[28] X.-Y. Wang and S. Gao, “Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network,” Inf. Sci., vol. 539, pp. 195–241, 2020.

[29] M. Kumar, A. Iqbal, and P. Kumar, “A new RGB image encryption algorithm based on DNA encoding and elliptic curve Diffie–Hellman cryptography,” Signal Process., vol. 125, pp. 187–202, Aug. 2016.

[30] A. Jain and N. Rajpal, “A robust image encryption algorithm resistant to attacks using DNA and chaotic logistic maps,” Multimedia Tools Appl., vol. 75, no. 10, pp. 5455–5472, May 2016.

[31] S. Kulkosn, D. Xiao, A. U. Rehman, and S. A. Abbas, “An efficient and noise resistant selective image encryption scheme for gray images based on chaotic maps and DNA complementary rules,” Multimedia Tools Appl., vol. 75, no. 1, pp. 1–23, Jan. 2016.

[32] P. Zhou, G. Zhao, L. Min, and X. Jin, “Chaos-based image encryption scheme combining DNA coding and entropy,” Multimedia Tools Appl., vol. 75, no. 11, pp. 6303–6319, Jun. 2016.

[33] X. Li, C. Zhou, and N. Xu, “A secure and efficient image encryption algorithm based on DNA coding and spatiotemporal chaos,” Int. J. Netw. Secur., vol. 20, no. 1, pp. 110–120, Jun. 2018.