Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A novel color image encryption algorithm based on a fractional-order discrete chaotic neural network and DNA sequence operations*

Li-ping CHEN†‡[1], Hao YIN[1], Li-guo YUAN[2], António M. LOPES[3], J. A. Tenreiro MACHADO[4], Ran-chao WU[5]

[1]*School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, China*

[2]*College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China*

[3]*UISPA–LAETA/INEGI, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, Porto 4200-465, Portugal*

[4]*Department of Electrical Engineering, Polytechnic Institute of Porto, R. Dr. António Bernardino de Almeida, 431, Porto 4249-015, Portugal*

[5]*School of Mathematical Sciences, Anhui University, Hefei 230601, China*

†E-mail: lip_chenhut@126.com

Received Dec. 18, 2019; Revision accepted Feb. 20, 2020; Crosschecked Apr. 16, 2020

Abstract: A novel color image encryption algorithm based on dynamic deoxyribonucleic acid (DNA) encoding and chaos is presented. A three-neuron fractional-order discrete Hopfield neural network (FODHNN) is employed as a pseudo-random chaotic sequence generator. Its initial value is obtained with the secret key generated by a five-parameter external key and a hash code of the plain image. The external key includes both the FODHNN discrete step size and order. The hash is computed with the SHA-2 function. This ensures a large secret key space and improves the algorithm sensitivity to the plain image. Furthermore, a new three-dimensional projection confusion method is proposed to scramble the pixels among red, green, and blue color components. DNA encoding and diffusion are used to diffuse the image information. Pseudo-random sequences generated by FODHNN are employed to determine the encoding rules for each pixel and to ensure the diversity of the encoding methods. Finally, confusion II and XOR are used to ensure the security of the encryption. Experimental results and the security analysis show that the proposed algorithm has better performance than those reported in the literature and can resist typical attacks.

Key words: Fractional-order discrete systems; Neural networks; Deoxyribonucleic acid (DNA) encryption; Color image encryption

https://doi.org/10.1631/FITEE.1900709 **CLC number:** TP37

1 Introduction

With the rapid development of computers and the Internet, the total amount and transmission speed of multimedia information have considerably increased. However, due to the openness of the network, it is difficult to guarantee the security of information which is generated during user communication. Because digital images are important parts of multimedia information, the security and integrity of digital image data in the transmission process has become a growing concern. Image encryption techniques are effective means of ensuring image

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 11971032) and the Science and Technology Program of Guangzhou, China (No. 201707010031)

ORCID: Li-ping CHEN, https://orcid.org/0000-0002-8110-5378

© Zhejiang University and Springer-Verlag GmbH Germany, part of Springer Nature 2020

transmission security. However, image information has special properties, such as bulk data capacity and strong correlation between pixels. Therefore, traditional encryption methods, such as the Advanced Encryption Standard (AES), Data Encryption Standard (DES), and International Data Encryption Algorithm (IDEA), are not suitable for image data (Zhou et al., 2014; Toughi et al., 2017). The design of new image encryption algorithms satisfying both confusion and diffusion requirements is still a challenge (Deng et al., 2011; Zhang GJ and Liu, 2011; Zhu et al., 2011; Zhang LY et al., 2012).

Recently, image encryption based on chaos has gradually matured and been widely used (Guan et al., 2005; Enayatifar et al., 2014; Chen JX et al., 2015; Zhang YQ and Wang, 2015; Zhou et al., 2015; Wang XY et al., 2016b; Zhang Y, 2018; Wu GC et al., 2019b). Chaotic systems are typical nonlinear dynamic systems, characterized by sensitivity to initial values, ergodicity, and high complexity. Such systems are suitable for pseudo-random number sequence generators in image encryption algorithms. Furthermore, fractional-order (FO) chaotic systems have advantages over the classical integer-order ones, because they allow the selection of a broader range of parameters and unveil more complex dynamic behaviors (Miller and Ross, 1988; Atıcı and Şengül, 2010; Zhang R et al., 2010; Kaslik and Sivasundaram, 2012; Goodrich and Peterson, 2015; Wu GC et al., 2019a; Huang et al., 2020). Therefore, the introduction of FO chaotic systems into the field of image encryption emerged as a new research topic (Chen LP et al., 2015, 2017). Wu GC et al. (2016) first proposed an FO image encryption algorithm, demonstrating that the rich dynamic behavior of chaotic systems and the varying FO could improve the security of the encryption algorithms. Recently, Abdeljawad et al. (2019) proposed image encryption based on double FO parameters. Nevertheless, the use of encryption based on chaotic systems without considering the structure of the algorithm can also lead to security problems in dealing with several typical attacks (Chen GR et al., 2004; Chen JL et al., 2015; Li CQ et al., 2017).

The deoxyribonucleic acid (DNA) technology has the advantages of high parallelism and high information density (Zheng et al., 2009; Enayatifar et al., 2015). Some researchers have combined the DNA technology with chaotic encryption methods to improve security. However, the resulting encryption algorithms still reveal several limitations. First, all image pixels are encoded/decoded using the same rules (Zhang Q and Wei, 2013; Norouzi and Mirzakuchaki, 2017), or the encoding/decoding rules are just regarded as part of the secret key (Zhang Q et al., 2014; Guesmi et al., 2016; Zhang YQ et al., 2016; Zhang LM et al., 2017). This results in a small key space that makes the encryption algorithm unable to withstand brute force or plaintext attacks. Second, the encryption processes are insensitive to changes in the plain image and the secret key. To solve these problems, some researchers proposed improved methods. Wu XJ et al. (2017) adopted a chaotic color image encryption algorithm based on DNA encryption and entropy to improve its sensitivity to the plain image. However, the same encoding rules were used for all pixels. Chen JX et al. (2018) used random encoding rules to generate DNA sequences and obtained a balanced information distribution in the encoding DNA matrix. However, because different images may have the same encoding rules, the security performance of the encryption is still compromised.

Motivated by the above findings, a novel FO chaotic color image encryption algorithm based on DNA encoding and the SHA-2 hash function is proposed. A new three-neuron fractional-order discrete Hopfield neural network (FODHNN) with rich dynamic behavior and a wide parameter selection space is employed to extend the secret key space. The five-parameter user input data and the first half of the 256-bit hash of the plain image are adopted to obtain the initial value of FODHNN. The remaining hash value is further considered to improve the sensitivity of the algorithm to the plain image. The three dimensions of FODHNN correspond to the image color components. The pseudo-random chaotic sequence generated by FODHNN is used to select the encoding rules for each pixel and, consequently, to avoid the problems associated with single DNA encoding methods. Finally, a new three-dimensional (3D) projection confusion method is proposed to scramble the pixels of the three color components of the plain image.

2 Preliminaries

2.1 Fractional-order discrete Hopfield neural network

The FODHNN used for image encryption is derived from a three-neuron integer-order continuous Hopfield-type neural network given by (Hopfield, 1982)

$$\boldsymbol{x}' = -\boldsymbol{x} + \boldsymbol{W} f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^3, \tag{1}$$

where $\boldsymbol{W} = \begin{pmatrix} 2 & 1 & -9 \\ -9 & 2 & 1 \\ 1 & -9 & 2 \end{pmatrix}$ and $f(\boldsymbol{x}) = \sin \boldsymbol{x}$. This yields the following system:

$$\begin{cases} x' = -x + 2\sin x + \sin y - 9\sin z, \\ y' = -y - 9\sin x + 2\sin y + \sin z, \\ z' = -z + \sin x - 9\sin y + 2\sin z. \end{cases} \tag{2}$$

Replacing the integer derivative by the FO operator in system (2) gives

$$\begin{cases} D^\nu x(t) = -x(t) + 2\sin(x(t)) + \sin(y(t)) \\ \qquad\qquad - 9\sin(z(t)), \\ D^\nu y(t) = -y(t) - 9\sin(x(t)) + 2\sin(y(t)) \\ \qquad\qquad + \sin(z(t)), \\ D^\nu z(t) = -z(t) + \sin(x(t)) - 9\sin(y(t)) \\ \qquad\qquad + 2\sin(z(t)), \end{cases} \tag{3}$$

where

$$D^\nu f(t) = \frac{1}{\Gamma(n-\nu)} \int_{t_0}^{t} (t-\tau)^{n-\nu-1} f^{(n)}(s) \mathrm{d}s$$

denotes the Caputo FO derivative, $t > 0$, $\nu \in (0, 1)$, the initial condition is $[x(0), y(0), z(0)]$, and $\Gamma(s) = \int_0^\infty t^{s-1} \mathrm{e}^{-t} \mathrm{d}t$ stands for the Gamma function.

Discretizing system (3) with piece-wise constant arguments (Agarwal et al., 2013; El Raheem and Salman, 2014), one obtains

$$\begin{cases} D^\nu x(t) = -x\left(\left(\frac{t}{h}\right)h\right) + 2\sin\left(x\left(\left(\frac{t}{h}\right)h\right)\right) \\ \qquad + \sin\left(y\left(\left(\frac{t}{h}\right)h\right)\right) - 9\sin\left(z\left(\left(\frac{t}{h}\right)h\right)\right), \\ D^\nu y(t) = -y\left(\left(\frac{t}{h}\right)h\right) - 9\sin\left(x\left(\left(\frac{t}{h}\right)h\right)\right) \\ \qquad + 2\sin\left(y\left(\left(\frac{t}{h}\right)h\right)\right) + \sin\left(z\left(\left(\frac{t}{h}\right)h\right)\right), \\ D^\nu z(t) = -z\left(\left(\frac{t}{h}\right)h\right) + \sin\left(x\left(\left(\frac{t}{h}\right)h\right)\right) \\ \qquad - 9\sin\left(y\left(\left(\frac{t}{h}\right)h\right)\right) + 2\sin\left(z\left(\left(\frac{t}{h}\right)h\right)\right), \end{cases} \tag{4}$$

$$\begin{cases} x(n+1) = x(n) + \frac{h^\nu}{\Gamma(1+\nu)}[-x(n) \\ \qquad + 2\sin\big(x(n)\big) + \sin\big(y(n)\big) \\ \qquad - 9\sin\big(z(n)\big)], \\ y(n+1) = y(n) + \frac{h^\nu}{\Gamma(1+\nu)}[-y(n) \\ \qquad - 9\sin\big(x(n)\big) + 2\sin\big(y(n)\big) \\ \qquad + \sin\big(z(n)\big)], \\ z(n+1) = z(n) + \frac{h^\nu}{\Gamma(1+\nu)}[-z(n) \\ \qquad + \sin\big(x(n)\big) - 9\sin\big(y(n)\big) \\ \qquad + 2\sin\big(z(n)\big)], \end{cases} \tag{5}$$

where $n \in \mathbb{N}_0$ and $h \in \mathbb{R}_+$ denotes the discretization step size.

Remark 1 As pointed out by Angstmann et al. (2017), the discretization method presented in Agarwal et al. (2013) and El Raheem and Salman (2014), for a class of initial-value problems involving the Caputo derivative, results in an incorrect first-order difference equation. This means that the difference system (5), being of first order, cannot capture the dynamics of the original system (3). However, it has no influence on the results, and we will show that the FO discrete system (5) is chaotic itself.

2.2 Chaotic behavior of FODHNN

To verify the pseudo-randomness of the chaotic sequence generated by FODHNN (5), the chaotic dynamic behavior of system (5) will be considered.

The 0–1 test was introduced by Gottwald and Melbourne (2004) to test the dynamic behavior of a system. The method does not need to reconstruct the phase space, using the system time series directly. An index k is calculated, and its value is used to decide the system dynamics. When the value of k is close to 1, the system dynamic behavior is chaotic, whereas when its value is close to zero, the system is regular (periodic or quasiperiodic). We apply the 0–1 test when the time series obtained for the initial value $\big(x(0), y(0), z(0)\big) = (0.08, 0.8, -6.2)$, FO ($\nu = 0.6$), and step size $h = 0.05$. The obtained value of k is 0.9983, showing that the dynamic behavior of the system is chaotic. Fig. 1 depicts the phase diagram of FODHNN. Fig. 2a illustrates the dynamics of the translation components (p, q) of the system (for details, see Gottwald and Melbourne (2004)). Fig. 2b shows the transformation

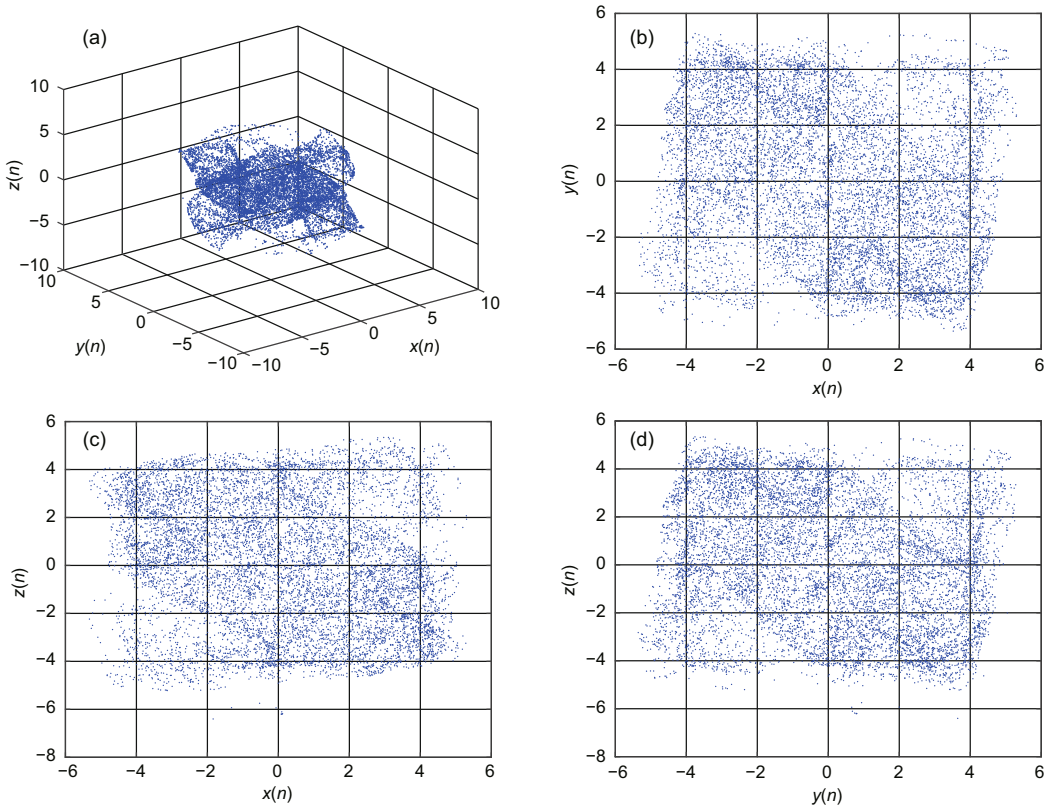

Fig. 1 Phase diagrams of the FODHNN defined in system (5) for the initial value $(x(0), y(0), z(0)) = (0.08, 0.8, -6.2)$, fractional order ($\nu = 0.6$), and step size $h = 0.05$: (a) x-y-z space; (b) x-y plane; (c) x-z plane; (d) y-z plane

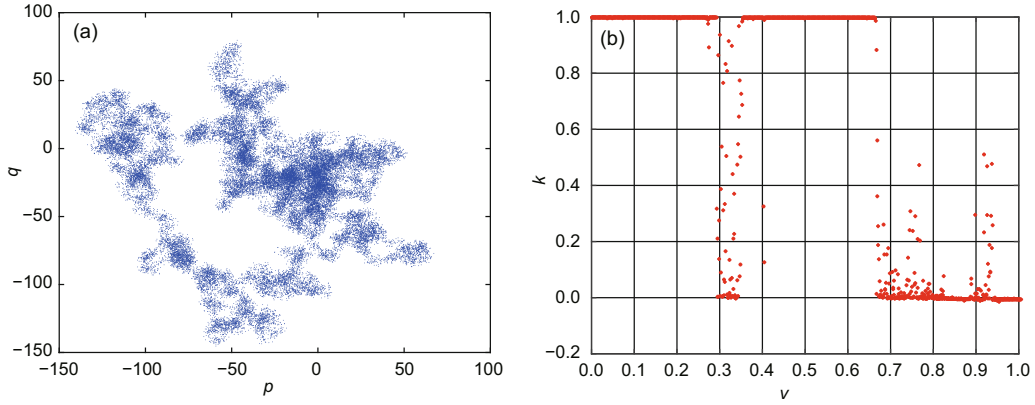

Fig. 2 Dynamics of translation components (p, q) of the system when k=0.9983 (a) and k under different orders of ν (b)

of the k value under different orders of ν, indicating the order range when the system shows the chaotic behavior.

2.3 DNA encoding and decoding

DNA is a unique molecule with a double helix structure that works as a storage medium of the genetic information of various organisms. There are four different nucleic acids in a DNA sequence, i.e., adenine (A), cytosine (C), guanine (G), and thymine (T). The nucleic acid A always pairs with T, and G always pairs with C (Machado et al., 2011; Machado, 2015, 2017). Similarly, in the binary system, the digits 0 and 1 are complementary. Therefore, we can

infer that 10 and 01, and 11 and 00 are also complementary. There are 24 kinds of encoding schemes (permutations) if we code A, T, G, C with 11, 10, 01, 00; however, only the eight rules shown in Table 1 satisfy the Watson-Crick complement rule.

Table 1 DNA encoding rules satisfying the Watson-Crick complement rule

Nucleic acid	1	2	3	4	5	6	7	8
A	00	00	01	01	10	10	11	11
T	11	11	10	10	01	01	00	00
C	01	10	00	11	00	11	01	10
G	10	01	11	00	11	00	10	01

Herein, the encrypted objects are color pictures that can be represented by three color channels, red (R), green (G), and blue (B). For example, if there is a pixel with a gray level 173 in the red channel, then its binary code is [10101101]. Based on the DNA encoding rule 1 in Table 1, we can obtain the DNA sequence [GGTC]. If we use rule 1 to decode the DNA sequence, then the original [10101101] is retrieved. However, if we choose rule 2, then the wrong binary value [01011110] is obtained.

With the rapid development of DNA computing, several researchers proposed basic operations based on DNA sequences. Tables 2 and 3 summarize the addition and subtraction respectively, which will be used in the following sections.

3 The proposed image encryption scheme

Fig. 3 depicts a schematic of the complete encryption system, including the chaotic secret code stream generator, 3D projection confusion, random DNA encoding, plaintext-unrelated diffusion, and confusion II and XOR modules. The implementation method is described in the next subsections.

3.1 Chaotic secret code stream generator based on FODHNN and SHA-2

The initial value of FODHNN depends on both a five-parameter user input vector (k_1, k_2, k_3, h, ν) and an external 256-bit key K generated by the SHA-256 hash function. First, we divide K into 4-bit blocks, K_i, $i = 1, 2, \ldots, 64$, so that $K = \{K_1, K_2, \ldots, K_{64}\}$, and we express each K_i in decimal. Then, the initial value of FODHNN (5) is given by

$$\begin{cases} x(0) = k_1 + \dfrac{\mathrm{mod}(K_1 + K_2 + \ldots + K_{12}, 256)}{256}, \\ y(0) = k_2 + \dfrac{\mathrm{mod}(K_{13} + K_{14} + \ldots + K_{24}, 256)}{256}, \\ z(0) = k_3 + \dfrac{\mathrm{mod}(K_{25} + K_{26} + \ldots + K_{36}, 256)}{256}, \end{cases}$$

where $\mathrm{mod}(a, b)$ represents the modular operation of a for b.

Table 2 Addition operation for DNA sequences

Nucleic acid (+)	A	C	G	T
A	T	A	C	G
C	A	C	G	T
G	C	G	T	A
T	G	T	A	C

Table 3 Subtraction operation for DNA sequences

Nucleic acid (−)	A	C	G	T
A	C	A	T	G
C	G	C	A	T
G	T	G	C	A
T	A	T	G	C

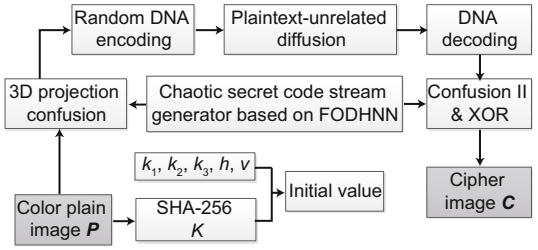

Fig. 3 Schematic representation of the new encryption scheme

3.2 3D projection confusion

A new 3D projection confusion is proposed to scramble the pixels between the R, G, and B components. For a color image $\boldsymbol{P}$ with $M \times N$ pixels ($\boldsymbol{P} \in \mathbb{R}^{M \times N}$), this is accomplished by Algorithm 1.

3.3 Random DNA encoding and plaintext-unrelated diffusion

In most DNA sequence based encryption methods, just one of the encoding schemes in Table 1 is chosen, leading to limited security of the encryption

Algorithm 1 3D projection confusion

1: Separate the plain image $\boldsymbol{P}$ ($\boldsymbol{P} \in \mathbb{R}^{M\times N}$) into its R, G, and B color component matrices, $\boldsymbol{R}$, $\boldsymbol{G}$, and $\boldsymbol{B}$ ($\in \mathbb{R}^{M\times N}$), respectively.

2: If $M = N$, then go to step 3; otherwise, fill with "0" to obtain a square plain image.

3: Iterate FODHNN (5) $O = 4MN + C$ times, where $C = (K_{37} + K_{38} + \ldots + K_{64}) \times 10$, discard the first C values, and retain the pseudo-random sequence $x(i), y(i), z(i)$, $i = 1, 2, \ldots, 4MN$ to minimize the impact of the transient.

4: Consider $\boldsymbol{R}$, $\boldsymbol{G}$, and $\boldsymbol{B}$ as three sides of a cube "A" (Fig. 4), and map the pseudo-random sequence $x(i)$, $y(i)$, $z(i)$ in step 3 to the cube by means of the expression

$$\begin{cases} X(i) = \text{floor}(\text{mod}(x(i) \times 10^{14}, 256)), \\ Y(i) = \text{floor}(\text{mod}(y(i) \times 10^{14}, 256)), \\ Z(i) = \text{floor}(\text{mod}(z(i) \times 10^{14}, 256)), \end{cases}$$

where floor(a) represents the maximum integer not greater than a. It results in that there is a point $a(X(i), Y(i), Z(i))$ in the cube "A," and that the points projected on surfaces $\boldsymbol{R}$, $\boldsymbol{G}$, and $\boldsymbol{B}$ are $R(X(i), Z(i))$, $G(Y(i), Z(i))$, and $B(X(i), Y(i))$, respectively.

5: Determine the scrambling method based on the projection variable $P_1(i) = \text{floor}(\text{mod}(z(i) \times 10^{14}, 5)) + 1$, as follows:

Case 1: when $P_1(i) = 1$, exchange the positions of $R(X(i), Z(i))$ and $B(X(i), Y(i))$.

Case 2: when $P_1(i) = 2$, exchange the positions of $R(X(i), Z(i))$ and $G(Y(i), Z(i))$.

Case 3: when $P_1(i) = 3$, first exchange the positions of $R(X(i), Z(i))$ and $B(X(i), Y(i))$, and then exchange the positions of $R(X(i), Z(i))$ and $G(Y(i), Z(i))$.

Case 4: when $P_1(i) = 4$, exchange the positions of $G(Y(i), Z(i))$ and $B(X(i), Y(i))$.

Case 5: when $P_1(i) = 5$, first exchange the positions of $G(Y(i), Z(i))$ and $B(X(i), Y(i))$, and then exchange the positions of $R(X(i), Z(i))$ and $B(X(i), Y(i))$.

Then, reshape the three matrices after scrambling, i.e., $\boldsymbol{R}_1$, $\boldsymbol{G}_1$, and $\boldsymbol{B}_1$ ($\in \mathbb{R}^{M\times N}$), to $\boldsymbol{R}_1$, $\boldsymbol{G}_1$, and $\boldsymbol{B}_1$ ($\in \mathbb{R}^{MN\times 1}$), respectively.

process. Herein, we propose a robust DNA encoding process based on the value of the variable

$$P_2(i) = \text{floor}\big(\text{mod}(y(i) \times 10^{14}, 8)\big) + 1,$$

which is used to select one of the encoding methods of Table 1 for each pixel in $\boldsymbol{R}_1, \boldsymbol{G}_1$, and $\boldsymbol{B}_1$ ($\in \mathbb{R}^{MN\times 1}$). For example, if $P_2(i) = 2$, then the DNA encoding rule 2 is adopted to encode the gray values of $R_1(i, 1), G_1(i, 1)$, and $B_1(i, 1)$. After encoding, DNA sequence matrices $\boldsymbol{R}_2$, $\boldsymbol{G}_2$, and $\boldsymbol{B}_2$ ($\in \mathbb{R}^{MN\times 1}$) are obtained.

To improve the security of the encryption algorithm, plaintext-unrelated diffusion is proposed through DNA operations to hide the image information. The procedure is given in Algorithm 2.

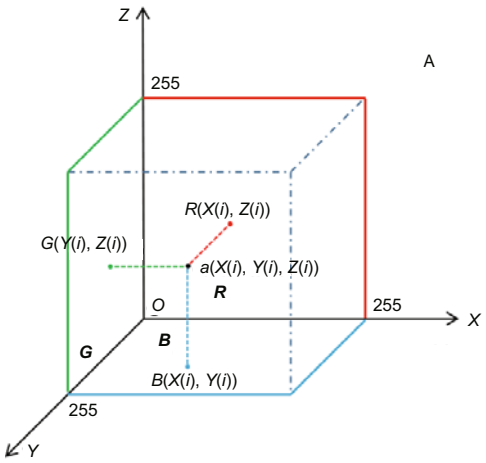

Fig. 4 Cube "A"

Algorithm 2 DNA diffusion

for $i = 1, 2, \ldots, MN$ **do**
  if $i = 1$ **then**
    $R_3(i) = R_2(i)$, $G_3(i) = G_2(i)$, $B_3(i) = B_2(i)$
  else
    $R_3(i) = R_2(i) + R_2(i-1)$
    $G_3(i) = G_2(i) + G_2(i-1)$
    $B_3(i) = B_2(i) + B_2(i-1)$
  end if
end for

After the diffusion process, we use P_2 with the rules in Table 1 to decode the DNA sequence, yielding $\boldsymbol{R}_3$, $\boldsymbol{G}_3$, and $\boldsymbol{B}_3$ ($\in \mathbb{R}^{MN\times 1}$).

3.4 Confusion II and XOR

The confusion II scheme is used to separately shuffle the elements in the three signal channels. This is accomplished in Algorithm 3. Additionally, the XOR is implemented by Algorithm 4.

Finally, we reshape $\boldsymbol{C}'_{\mathrm{R}}$, $\boldsymbol{C}'_{\mathrm{G}}$, and $\boldsymbol{C}'_{\mathrm{B}}$ into the $M \times N$ dimensional matrices $\boldsymbol{C}_{\mathrm{R}}, \boldsymbol{C}_{\mathrm{G}}$, and $\boldsymbol{C}_{\mathrm{B}}$, which represent the red, green, and blue components of

the ciphered image, respectively. Decryption is the inverse process of the encryption scheme, performed with the inverse DNA addition operation shown in Table 3.

Remark 2 Existing algorithms either use high-dimensional chaotic systems (Hu et al., 2018; Li Z et al., 2018; Al-Hazaimeh et al., 2019), or try to expand and improve existing low-dimensional chaotic systems (Ye et al., 2018; Essaid et al., 2019; Hanis and Amutha, 2019; Hua et al., 2019; Li RZ et al., 2019). In the proposed encryption scheme, FODHNN is used as a pseudo-random number sequence generator. It has a larger parameter selection space and stronger chaotic characteristics than other systems, which greatly improves the secret key space and ensures system security.

Remark 3 In the process of DNA encoding, the same method is commonly adopted for all pixels (Ravichandran et al., 2017; Wang JS et al., 2017; Ahgue et al., 2018; Sun, 2018). In the proposed algorithm, a pseudo-random sequence generated by FODHNN is used to select a unique encoding method for each pixel of the color image. This makes it difficult for hackers to obtain useful information from the DNA matrix, thus improving the security performance of encryption.

Remark 4 The novel 3D projection confusion algorithm (1) links the three color components of the image through the spatial stereo structure, (2) maps the pseudo-random number sequence generated by FODHNN into the space body, and (3) confuses the three projection points by random selection. This strategy can significantly confuse the pixels between different color channels, considerably reducing the correlation between pixels, and improving the security of the encryption algorithm.

Algorithm 3 Confusion II

1: Sort the sequence $x(j)$, $y(j)$, $z(j)$, $j = 3MN + 1, 3MN + 2, \ldots, 4MN$, using the pseudo-random stream $x(i)$, $y(i)$, $z(i)$ from FODHNN, $i = 1, 2, \ldots, MN$, and the formula

$$\begin{cases} [x_1(i), T_x] = \mathrm{sort}(x(j)), \\ [y_1(i), T_y] = \mathrm{sort}(y(j)), \\ [z_1(i), T_z] = \mathrm{sort}(z(j)), \end{cases}$$

where sort$(\cdot)$ is the sort function in MATLAB. Obtain position index sequences T_x, T_y, and T_z.

2: Rearrange every element of $\boldsymbol{R}_3$ ($\boldsymbol{R}_3 \in \mathbb{R}^{MN\times 1}$) with sequence T_x. This means that the $(T_x(1))^{\mathrm{th}}$ element in $\boldsymbol{R}_3$ exchanges its location with the first one, the $(T_x(2))^{\mathrm{th}}$ element exchanges its location with the second one, and so on. When the last element completes the exchanging process, the confused sequence $\boldsymbol{R}_4$ ($\boldsymbol{R}_4 \in \mathbb{R}^{MN\times 1}$) is generated.

3: Use the method in step 2 to rearrange $\boldsymbol{G}_3$ with sequence T_y and $\boldsymbol{B}_3$ with sequence T_z. Then, sequences $\boldsymbol{G}_4$ and $\boldsymbol{B}_4$ are obtained.

Algorithm 4 XOR

1: Using the mapping sequences $X(i)$, $Y(i)$, $Z(i)$ obtained in step 4 of the 3D projection confusion scheme, the XOR matrices are obtained as

$$\begin{cases} X_{\mathrm{R}}(1:MN) = X(2MN+1:3MN), \\ X_{\mathrm{G}}(1:MN) = Y(2MN+1:3MN), \\ X_{\mathrm{B}}(1:MN) = Z(2MN+1:3MN). \end{cases}$$

2: The cipher matrices are derived as follows:

$$\begin{cases} \boldsymbol{C}'_{\mathrm{R}}(i) = \mathrm{bitxor}(R_4(i), X_{\mathrm{R}}(i)), \\ \boldsymbol{C}'_{\mathrm{G}}(i) = \mathrm{bitxor}(G_4(i), X_{\mathrm{G}}(i)), \\ \boldsymbol{C}'_{\mathrm{B}}(i) = \mathrm{bitxor}(B_4(i), X_{\mathrm{B}}(i)), \end{cases}$$

where bitxor$(\cdot)$ denotes the bitwise XOR function.

4 Simulation results and security analysis

4.1 Simulations

To verify the performance of the proposed encryption method, we carried out several simulations using the software MATLAB 2016a on a laptop equipped with Intel® Core™ i5-8300H (2.3 GHz) with 8 GB memory.

4.2 Security analysis

We tested the algorithm with four 256×256 color images, namely Lena, Baboon, all black, and all white. Fig. 5 depicts the plain, ciphered, and deciphered images of Lena, and the corresponding histograms, obtained with the proposed scheme for the five-parameter user-supplied key (0.07, 0.8, −6.2, 0.05, 0.6). Fig. 6 (on P.874) shows the plain and ciphered images of all black, all white, and Baboon.

4.2.1 Key space

The secret key space is an important metric to measure the ability of the encryption algorithms to

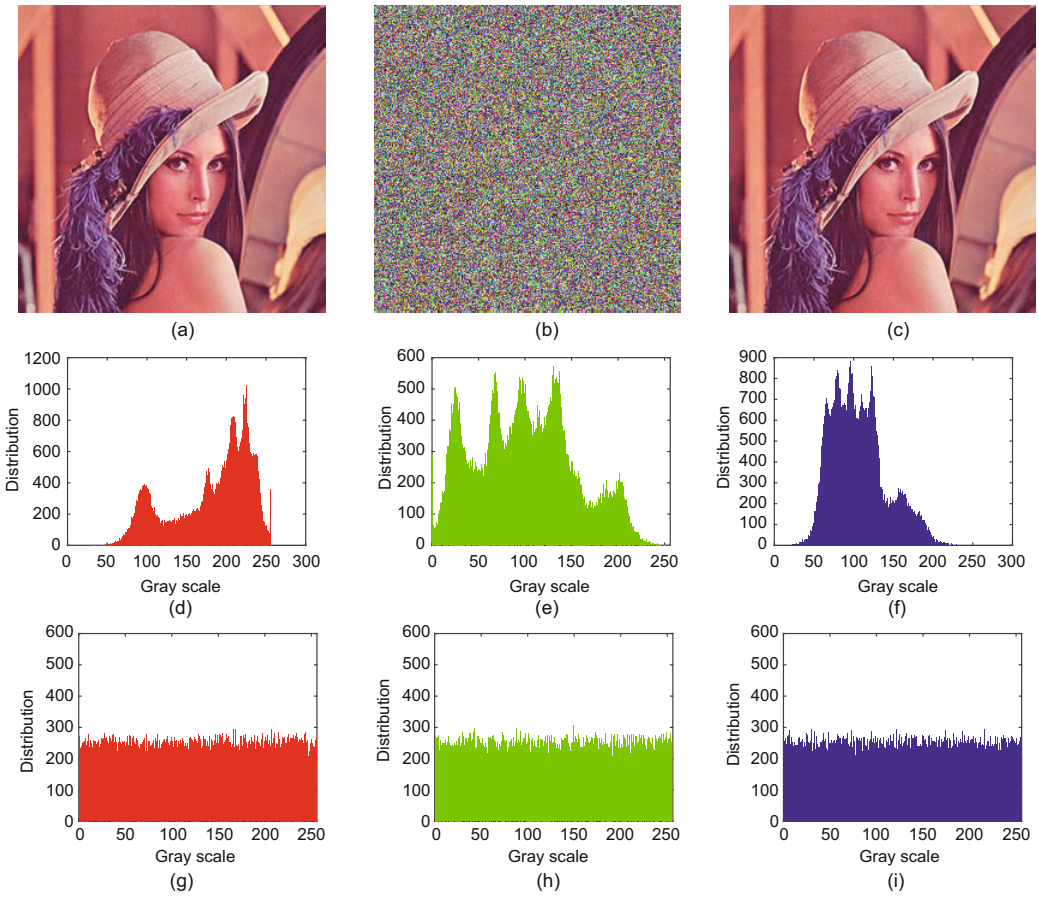

Fig. 5 Lena: (a) plain image; (b) ciphered image; (c) deciphered image; (d) histogram of the plain image (red component); (e) histogram of the plain image (green component); (f) histogram of the plain image (blue component); (g) histogram of the ciphered image (red component); (h) histogram of the ciphered image (green component); (i) histogram of the ciphered image (blue component)

resist brute-force attacks. Generally, algorithms are designed so that their secret key space is larger than 2^{100}. In the proposed algorithm, the secret key consists of a five-parameter user-supplied vector and a 256-bit hash. It is known that the precision of the user input data is 10^{15}, while the precision of FO and the discrete step are also 10^{15}. Considering the order range of FODHNN when it has the chaotic behavior, the size of the key space due to the order is 10^{14}, yielding a key space size of 10^{74}. Furthermore, the key space due to the hash value is 2^{128}. Therefore, the total secret key space assumes the value $10^{74} \times 2^{128}$, making any brute-force attack to the algorithm computationally infeasible.

4.2.2 Correlation coefficient test

The correlation coefficient r of two N-dimensional random variables $\boldsymbol{u}$ and $\boldsymbol{v}$ is a measure of their linear dependence and is given by

$$r = \frac{\mathrm{cov}(\boldsymbol{u}, \boldsymbol{v})}{\sigma(\boldsymbol{u}) \cdot \sigma(\boldsymbol{v})}, \tag{6}$$

where $\mathrm{cov}(\boldsymbol{u}, \boldsymbol{v})$ denotes the covariance of $\boldsymbol{u}$ and $\boldsymbol{v}$ and $\sigma(\cdot)$ is the standard deviation.

We calculated the correlation coefficient for three types of adjacent points, namely horizontal, vertical, and diagonal adjacent pixels. Table 4 shows the results obtained for the images Lena, Baboon, all black, and all white, when randomly choosing 10 000 pairs of adjacent pixels from the three color channels of the original and the encrypted image. Fig. 7 (on P.875) depicts the correlation distribution of vertical, horizontal, and diagonal adjacent pixels of the original and ciphered images of Lena. We can verify from Table 4 and Fig. 7 that the pixels in the plain image have strong correlation and that the correlation coefficient of the encrypted image is close to zero.

4.2.3 Information entropy test

Shannon entropy (Gray, 2011) is an important indicator for assessing the degree of information randomness. For an L-level grayscale image, the entropy is given by

$$H = -\sum_{i=0}^{L} p(i)\log_2 p(i),$$

where $p(i)$ is the probability of occurrence of the i^{th} level. For a random 256-level (i.e., $L = 256$) grayscale image, we have $H = 8$.

Table 5 presents the entropy values of the ciphered images Lena, Baboon, all black, and all white obtained using the proposed algorithm. We can verify that H is always close to the limit value $H = 8$, meaning that the plain image information is well concealed. Table 6 compares the performance of the new algorithm with that exhibited by the methods presented in Zhang YS and Xiao (2014), Liu and Kadir (2015), Wu XJ et al. (2015), and Chai et al. (2019). The results show that the novel algorithm is more efficient than its counterparts.

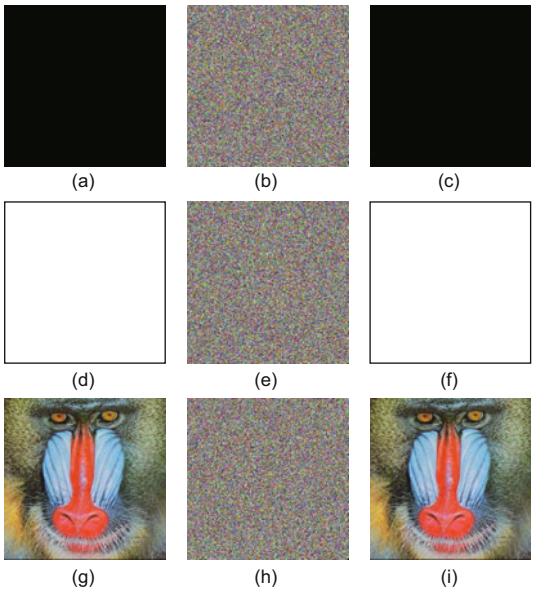

Fig. 6 All black, all white, and Baboon: (a) all black plain image; (b) ciphered image of (a); (c) deciphered image of (b); (d) all white plain image; (e) ciphered image of (d); (f) deciphered image of (e); (g) Baboon plain image; (h) ciphered image of (g); (i) deciphered image of (h)

4.2.4 Differential attack test

To measure the ability of the encryption algorithm to resist differential attacks, we examined two indices, namely (1) the number of pixels change rate (NPCR) and (2) the unified average changing intensity (UACI). NPCR compares the values of corresponding pixels in two images, yielding the number

Table 4 Values of r obtained for the images Lena, Baboon, all black, and all white, when randomly choosing 10 000 pairs of horizontal, vertical, and diagonal adjacent pixels

Image	Channel	r		
		Horizontal	Vertical	Diagonal
Lena plaintext	R	0.9622	0.9323	0.8958
	G	0.9628	0.9305	0.8963
	B	0.9273	0.8767	0.8310
Lena ciphertext	R	0.0001	0.0091	−0.0023
	G	−0.0025	−0.0061	0.0058
	B	−0.0074	−0.0059	0.0015
Baboon ciphertext	R	−0.0015	−0.0010	−0.0083
	G	0.0070	0.0014	0.0053
	B	−0.0087	0.0030	0.0032
All black ciphertext	R	0.0057	0.0048	0.0016
	G	−0.0063	−0.0040	0.0021
	B	0.0025	0.0084	−0.0095
All white ciphertext	R	−0.0082	−0.0056	−0.0074
	G	0.0086	0.0045	−0.0056
	B	−0.0062	−0.0050	−0.0091

Table 5 The entropy H values of the ciphered images Lena, Baboon, all black, and all white obtained using the proposed algorithm

Image	H		
	R channel	G channel	B channel
Lena plaintext	7.3147	7.6391	7.0542
Lena ciphertext	7.9974	7.9971	7.9975
Baboon ciphertext	7.9971	7.9971	7.9972
All black ciphertext	7.9971	7.9969	7.9972
All white ciphertext	7.9974	7.9974	7.9973

Table 6 Values of H for the ciphered image Lena obtained by the proposed algorithm and those yielded by the competitive methods

Algorithm	H		
	R channel	G channel	B channel
Ours	7.9974	7.9971	7.9975
Chai et al. (2019)	7.9973	7.9969	7.9971
Wu XJ et al. (2015)	7.9893	7.9896	7.9903
Zhang YS and Xiao (2014)	7.9973	7.9972	7.9969
Liu and Kadir (2015)	7.9896	7.9893	7.9896

of different pairs as a percentage of the total number of pixels. UACI captures the average of differences between the corresponding pixels. NPCR and UACI are given by

$$\begin{cases} \text{NPCR} = \dfrac{\sum_{i,j} D(i,j)}{MN} \times 100\%, \\ \text{UACI} = \dfrac{\sum_{i,j} |C(i,j) - C'(i,j)|}{MN} \times 100\%, \end{cases}$$

where M and N denote the size of the image, and $\boldsymbol{C}$ and $\boldsymbol{C'}$ represent two ciphered images whose plain versions differ just by one pixel. Therefore, if $C(i,j) \neq C'(i,j)$, then $D(i,j) = 1$; otherwise, $D(i,j) = 0$. Ideally, we should have NPCR=99.6094% and UACI=33.4635%. In the experiments, we randomly selected one pixel in the image Lena and added value 1 to that pixel, obtaining a new image denoted as Lena'. The two images, Lena and Lena', were then encrypted, and the NPCR and UACI of the R, G, and B channels were calculated, yielding 99.61%, 99.61%, and 99.62%, for the first, and 33.48%, 33.47%, and 33.43%, for the second. Table 7 compares the results obtained using the proposed algorithm and the methods described in Wu XJ et al. (2015), Wang XY et al. (2016a), ur Rehman et al. (2018), and Chai et al. (2019). We verify that the new encryption scheme reveals a superior resilience to differential attacks, as measured by the indices NPCR and UACI.

4.2.5 Sensitivity analysis

In general, a good encryption algorithm should be sensitive to the secret key. Minimum changes in just one pixel of the plain image result in completely different hash values, and thus the secret key also changes dramatically due to its hash value component K. In the following we discuss the impact of

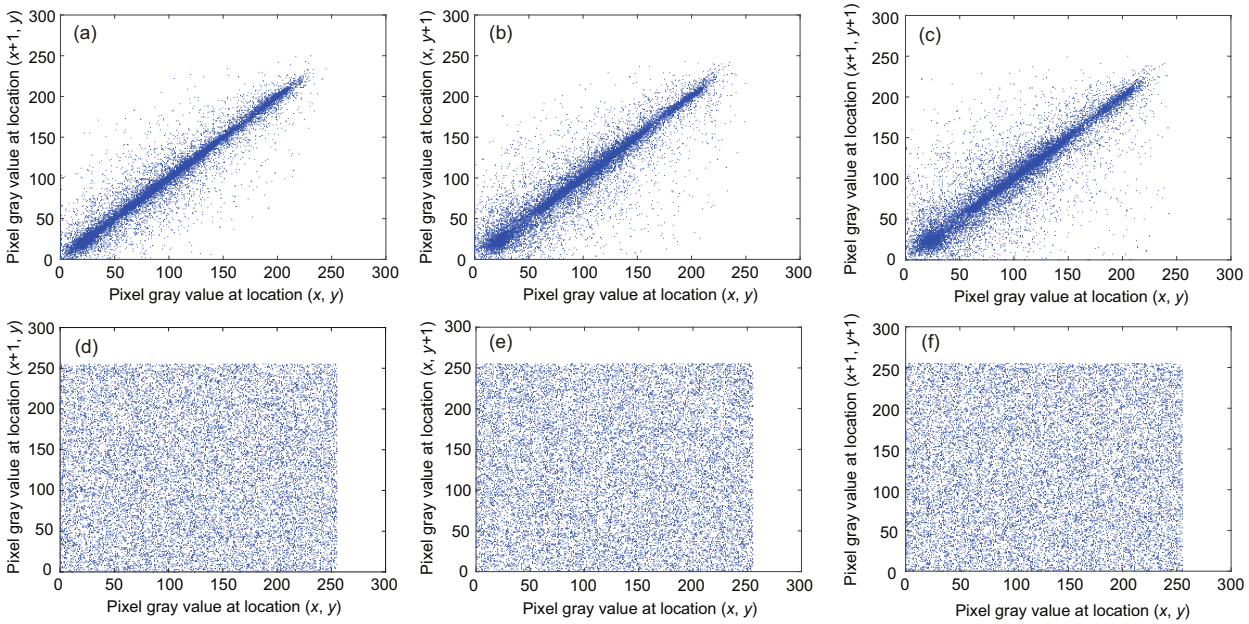

Fig. 7 Correlation distribution of vertical, horizontal, and diagonal adjacent pixels of the original and ciphered images of Lena: (a) horizontal, plain image; (b) vertical, plain image; (c) diagonal, plain image; (d) horizontal, ciphered image; (e) vertical, ciphered image; (f) diagonal, ciphered image

Table 7 NPCR and UACI obtained using the proposed algorithm and the competitive methods

Algorithm	NPCR (%)				UACI (%)			
	R channel	G channel	B channel	Average	R channel	G channel	B channel	Average
Ours	99.61	99.61	99.62	99.613	33.48	33.47	33.43	33.460
Chai et al. (2019)	99.60	99.61	99.61	99.607	33.56	33.46	33.49	33.503
Wu XJ et al. (2015)	99.61	99.60	99.60	99.603	33.46	33.50	33.47	33.477
ur Rehman et al. (2018)	99.60	99.60	99.60	99.600	33.36	33.43	33.37	33.387
Wang XY et al. (2016a)	99.63	99.60	99.60	99.610	33.60	33.30	33.40	33.433

minor changes of the secret key on the encryption and decryption processes. The image Lena and the following seven secret keys $\boldsymbol{D}_i$ $(i = 1, 2, \ldots, 7)$ were tested (component K is expressed in hexadecimal).

$\boldsymbol{D}_1$ =[0.8, 0.81, −6.2, 0.05, 0.6, CA595AB9743521DE72E65630613837627ACEAC0A67944180B1095E09D1D3847C],

$\boldsymbol{D}_2$ =[0.8 + 10^{-15}, 0.81, −6.2, 0.05, 0.6, CA595AB9743521DE72E65630613837627ACEAC0A67944180B1095E09D1D3847C],

$\boldsymbol{D}_3$ =[0.8, 0.81 + 10^{-15}, −6.2, 0.05, 0.6, CA595AB9743521DE72E65630613837627ACEAC0A67944180B1095E09D1D3847C],

$\boldsymbol{D}_4$ =[0.8, 0.81, −6.2 + 10^{-15}, 0.05, 0.6, CA595AB9743521DE72E65630613837627ACEAC0A67944180B1095E09D1D3847C],

$\boldsymbol{D}_5$ =[0.8, 0.81, −6.2, 0.05 + 10^{-15}, 0.6, CA595AB9743521DE72E65630613837627ACEAC0A67944180B1095E09D1D3847C],

$\boldsymbol{D}_6$ =[0.8, 0.81, −6.2, 0.05, 0.6 + 10^{-15}, CA595AB9743521DE72E65630613837627ACEAC0A67944180B1095E09D1D3847C],

$\boldsymbol{D}_7$ =[0.8, 0.81, −6.2, 0.05, 0.6, DA595AB9743521DE72E65630613837627ACEAC0A67944180B1095E09D1D3847C].

First, keys $\boldsymbol{D}_2$, $\boldsymbol{D}_3$, and $\boldsymbol{D}_4$ were obtained from the original $\boldsymbol{D}_1$ by adding 10^{-15} to the parameters k_1, k_2, and k_3, respectively. Key $\boldsymbol{D}_5$ was derived from key $\boldsymbol{D}_1$ by adding 10^{-15} to the discrete step h. Key $\boldsymbol{D}_6$ was obtained by adding 10^{-15} to FO ν. Key $\boldsymbol{D}_7$ was obtained by adding 1 to the K_1 component of K. Table 8 shows the NPCR and UACI values of the Lena images ciphered with the secret keys $\boldsymbol{D}_i$ $(i = 2, 3, \ldots, 7)$. It can be seen that the secret keys change only slightly, but yield changes of more than 99% on the encrypted images' pixels. Therefore, the sensitivity of the proposed algorithm to the secret key is considerable.

Second, key $\boldsymbol{D}_1$ was used to encrypt the plain image Lena, and keys $\boldsymbol{D}_i$ $(i = 1, 2, \ldots, 7)$ were used to decrypt the ciphered image. Fig. 8 depicts the resulting decrypted images. We can verify that the secret key cannot be decrypted successfully with only minor changes.

In summary, the proposed encryption algorithm has good key sensitivity in what concerns the encryption and decryption processes.

4.2.6 Noise and occlusion attack analysis

In real-world applications, the image may be affected with noise and occlusion in the process of transmission. To resist these types of attacks, the encryption algorithm should be able to restore the information.

We tested the proposed algorithm with the

Table 8 NPCR and UACI of the Lena image ciphered with different secret keys

Secret key	NPCR (%)			UACI (%)		
	R channel	G channel	B channel	R channel	G channel	B channel
$\boldsymbol{D}_2$	99.61	99.66	99.64	33.50	33.42	33.41
$\boldsymbol{D}_3$	99.63	99.59	99.60	33.36	33.40	33.43
$\boldsymbol{D}_4$	99.58	99.67	99.61	33.46	33.48	33.63
$\boldsymbol{D}_5$	99.59	99.62	99.63	33.51	33.42	33.37
$\boldsymbol{D}_6$	99.63	99.62	99.61	33.45	33.47	33.56
$\boldsymbol{D}_7$	99.63	99.58	99.61	33.32	33.54	33.54

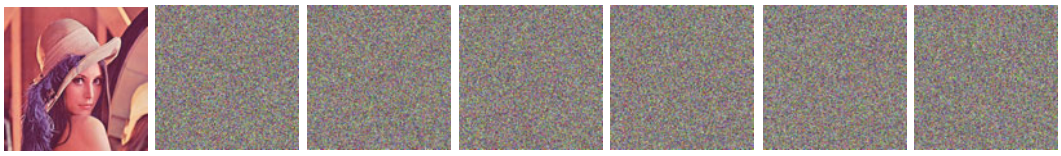

Fig. 8 Images of Lena encrypted with the secret key D_1 and decrypted with D_1 (a), D_2 (b), D_3 (c), D_4 (d), D_5 (e), D_6 (f), and D_7 (g)

image Lena. Fig. 9 depicts several ciphered images with different occlusion effects and their corresponding decrypted versions. We can verify that the original image information is restored, meaning that the algorithm has a good ability to resist occlusion attacks. Similarly, Fig. 10 shows ciphered images corrupted by the Gaussian and Salt&Pepper types of noise, as well as their corresponding decrypted versions. It can be seen that the encryption algorithm can also deal with noise attacks.

5 Conclusions

A new discrete chaotic encryption algorithm based on DNA encoding and SHA-256 has been presented. First, a new FODHNN model was obtained and adopted to generate pseudo-random sequences. The five-parameter user input data, including FO and the step size, and the hash value of the original image obtained with the SHA-256 function, were adopted to generate the initial value of FODHNN. Second, a 3D projection confusion method based on a pseudo-random sequence has been proposed to scramble the pixels between the R, G, and B components of the plain image. Third, a new random DNA encoding method and DNA encoding based diffusion have been proposed to further diffuse the information of the plain image. Finally, DNA sequence decoding, confusion II, and XOR have been implemented to improve the encryption method. The simulation results and security analysis showed that the proposed algorithm is suitable for color image encryption.

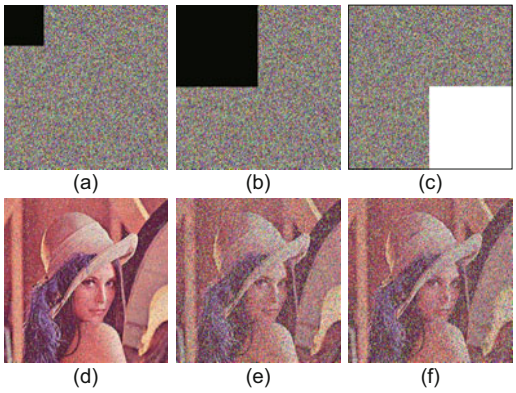

Fig. 9 Occlusion attack on the image Lena. Ciphered images with different occlusion effects and their corresponding decrypted versions: (a) cutting 1/16 on the top left corner; (b) cutting 1/4 on the top left corner; (c) cutting 1/4 on the bottom right corner; (d) decrypted version of (a); (e) decrypted version of (b); (f) decrypted version of (c)

Contributors

Li-ping CHEN designed the research. Ran-chao WU processed the data. Hao YIN drafted the manuscript. Li-guo YUAN helped organize the manuscript. Li-ping CHEN, António M. LOPES, and J. A. Tenreiro MACHADO revised and finalized the paper.

Compliance with ethics guidelines

Li-ping CHEN, Hao YIN, Li-guo YUAN, António M. LOPES, J. A. Tenreiro MACHADO, and Ran-chao WU declare that they have no conflict of interest.

References

Abdeljawad T, Banerjee S, Wu GC, 2019. Discrete tempered fractional calculus for new chaotic systems with short memory and image encryption. *Optik*, in press. https://doi.org/10.1016/j.ijleo.2019.163698

Agarwal RP, El-Sayed AMA, Salman SM, 2013. Fractional-order Chua's system: discretization, bifurcation and chaos. *Adv Differ Equat*, 2013:320. https://doi.org/10.1186/1687-1847-2013-320

Ahgue AO, de Nkapkop JD, Effa JY, et al., 2018. A DNA-based chaos algorithm for an efficient image encryption application. Int Symp on Electronics and Telecommunications, p.1-4. https://doi.org/10.1109/ISETC.2018.8583850

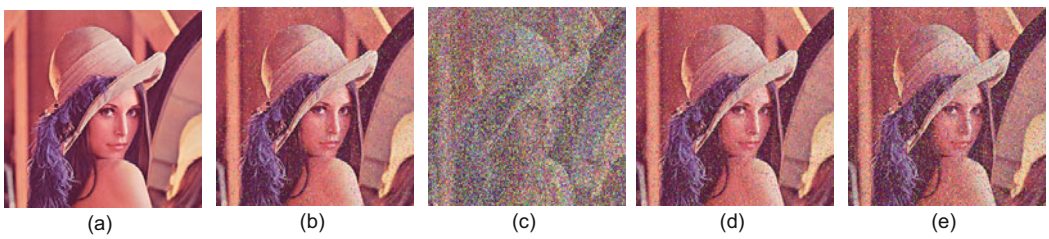

Fig. 10 Noise attack on the image Lena. Ciphered images corrupted with noise and their corresponding decrypted versions: (a) original image; (b) corrupted image with Gaussian noise with density 10^{-6}; (c) corrupted image with Gaussian noise with density 10^{-5}; (d) corrupted image with Salt&Pepper noise with density 0.05; (e) corrupted image with Salt&Pepper noise with density 0.1

Al-Hazaimeh OM, Al-Jamal MF, Alhindawi N, et al., 2019. Image encryption algorithm based on Lorenz chaotic map with dynamic secret keys. *Neur Comput Appl*, 31(7):2395-2405. https://doi.org/10.1007/s00521-017-3195-1

Angstmann CN, Henry BI, Jacobs BA, et al., 2017. Discretization of fractional differential equations by a piecewise constant approximation. *Math Model Nat Phenom*, 12(6):23-36. https://doi.org/10.1051/mmnp/2017063

Atıcı FM, Şengül S, 2010. Modeling with fractional difference equations. *J Math Anal Appl*, 369(1):1-9. https://doi.org/10.1016/j.jmaa.2010.02.009

Chai XL, Fu XL, Gan ZH, et al., 2019. A color image cryptosystem based on dynamic DNA encryption and chaos. *Signal Process*, 155:44-62. https://doi.org/10.1016/j.sigpro.2018.09.029

Chen GR, Mao YB, Charles KC, 2004. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Sol Fract*, 21(3):749-761. https://doi.org/10.1016/j.chaos.2003.12.022

Chen JL, Lei C, Lin SL, et al., 2015. Preparation and structural characterization of a partially depolymerized beta-glucan obtained from *Poria cocos* sclerotium by ultrasonic treatment. *Food Hydrocoll*, 46:1-9. https://doi.org/10.1016/j.foodhyd.2014.12.005

Chen JX, Zhu ZL, Fu C, et al., 2015. An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach. *Commun Nonl Sci Numer Simul*, 23(1-3):294-310. https://doi.org/10.1016/j.cnsns.2014.11.021

Chen JX, Zhu ZL, Zhang LB, et al., 2018. Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption. *Signal Process*, 142:340-353. https://doi.org/10.1016/j.sigpro.2017.07.034

Chen LP, Wu RC, He YG, et al., 2015. Robust stability and stabilization of fractional-order linear systems with polytopic uncertainties. *Appl Math Comput*, 257:274-284. https://doi.org/10.1016/j.amc.2014.12.103

Chen LP, Cao JD, Wu RC, et al., 2017. Stability and synchronization of fractional-order memristive neural networks with multiple delays. *Neur Netw*, 94:76-85. https://doi.org/10.1016/j.neunet.2017.06.012

Deng SJ, Zhan YP, Xiao D, et al., 2011. Analysis and improvement of a hash-based image encryption algorithm. *Commun Nonl Sci Numer Simul*, 16(8):3269-3278. https://doi.org/10.1016/j.cnsns.2010.12.016

El Raheem ZF, Salman SM, 2014. On a discretization process of fractional-order logistic differential equation. *J Egypt Math Soc*, 22(3):407-412. https://doi.org/10.1016/j.joems.2013.09.001

Enayatifar R, Abdullah AH, Isnin IF, 2014. Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. *Opt Lasers Eng*, 56:83-93. https://doi.org/10.1016/j.optlaseng.2013.12.003

Enayatifar R, Sadaei HJ, Abdullah AH, et al., 2015. A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata. *Opt Lasers Eng*, 71:33-41. https://doi.org/10.1016/j.optlaseng.2015.03.007

Essaid M, Akharraz I, Saaidi A, et al., 2019. A novel image encryption scheme based on permutation/diffusion process using an improved 2D chaotic system. Int Conf on Wireless Technologies, Embedded and Intelligent Systems, p.1-6. https://doi.org/10.1109/WITS.2019.8723717

Goodrich C, Peterson AC, 2015. Discrete Fractional Calculus. Springer, New York, USA. https://doi.org/10.1007/978-3-319-25562-0

Gottwald GA, Melbourne I, 2004. A new test for chaos in deterministic systems. *Proc R Soc Lond Ser A*, 460(2042):603-611. https://doi.org/10.1098/rspa.2003.1183

Gray RM, 2011. Entropy and Information Theory (2nd Ed.). Springer, New York, USA. https://doi.org/10.1007/978-1-4419-7970-4

Guan ZH, Huang FJ, Guan WJ, 2005. Chaos-based image encryption algorithm. *Phys Lett A*, 346(1-3):153-157. https://doi.org/10.1016/j.physleta.2005.08.006

Guesmi R, Farah MAB, Kachouri A, et al., 2016. A novel chaos-based image encryption using DNA sequence operation and secure Hash algorithm SHA-2. *Nonl Dynam*, 83(3):1123-1136. https://doi.org/10.1007/s11071-015-2392-7

Hanis S, Amutha R, 2019. A fast double-keyed authenticated image encryption scheme using an improved chaotic map and a butterfly-like structure. *Nonl Dynam*, 95(1):421-432. https://doi.org/10.1007/s11071-018-4573-7

Hopfield JJ, 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci USA*, 79(8):2554-2558. https://doi.org/10.1073/pnas.79.8.2554

Hu GY, Kou WL, Dong JE, et al., 2018. A novel image encryption algorithm based on cellular neural networks hyper chaotic system. IEEE 4th Int Conf on Computer and Communications, p.1878-1882. https://doi.org/10.1109/CompComm.2018.8780725

Hua ZY, Zhou YC, Huang HJ, 2019. Cosine-transform-based chaotic system for image encryption. *Inform Sci*, 480:403-419. https://doi.org/10.1016/j.ins.2018.12.048

Huang LL, Park JH, Wu GC, et al., 2020. Variable-order fractional discrete-time recurrent neural networks. *J Comput Appl Math*, 370:112633. https://doi.org/10.1016/j.cam.2019.112633

Kaslik E, Sivasundaram S, 2012. Nonlinear dynamics and chaos in fractional-order neural networks. *Neur Netw*, 32:245-256. https://doi.org/10.1016/j.neunet.2012.02.030

Li CQ, Lin DD, Li JH, 2017. Cryptanalyzing an image-scrambling encryption algorithm of pixel bits. *IEEE Multim*, 24(3):64-71. https://doi.org/10.1109/MMUL.2017.3051512

Li RZ, Liu Q, Liu LF, 2019. Novel image encryption algorithm based on improved logistic map. *IET Image Process*, 13(1):125-134. https://doi.org/10.1049/iet-ipr.2018.5900

Li Z, Peng CG, Li LR, et al., 2018. A novel plaintext-related image encryption scheme using hyper-chaotic system. *Nonl Dynam*, 94(2):1319-1333. https://doi.org/10.1007/s11071-018-4426-4

Liu HJ, Kadir A, 2015. Asymmetric color image encryption scheme using 2D discrete-time map. *Signal Process*, 113:104-112. https://doi.org/10.1016/j.sigpro.2015.01.016

Machado JAT, 2015. Fractional order description of DNA. *Appl Math Model*, 39(14):4095-4102. https://doi.org/10.1016/j.apm.2014.12.037

Machado JAT, 2017. Bond graph and memristor approach to DNA analysis. *Nonl Dynam*, 88(2):1051-1057. https://doi.org/10.1007/s11071-016-3294-z

Machado JAT, Costa AC, Quelhas MD, 2011. Entropy analysis of the DNA code dynamics in human chromosomes. *Comput Math Appl*, 62(3):1612-1617. https://doi.org/10.1016/j.camwa.2011.03.005

Miller KS, Ross B, 1988. Fractional difference calculus. Proc Int Symp on Univalent Functions, Fractional Calculus and Their Applications, p.139-152.

Norouzi B, Mirzakuchaki S, 2017. An image encryption algorithm based on DNA sequence operations and cellular neural network. *Multim Tools Appl*, 76(11):13681-13701. https://doi.org/10.1007/s11042-016-3769-4

Ravichandran D, Praveenkumar P, Rayappan JBB, et al., 2017. DNA chaos blend to secure medical privacy. *IEEE Trans NanoBiosci*, 16(8):850-858. https://doi.org/10.1109/TNB.2017.2780881

Sun SL, 2018. A novel hyperchaotic image encryption scheme based on DNA encoding, pixel-level scrambling and bit-level scrambling. *IEEE Photon J*, 10(2):7201714. https://doi.org/10.1109/JPHOT.2018.2817550

Toughi S, Fathi MH, Sekhavat YA, 2017. An image encryption scheme based on elliptic curve pseudo random and Advanced Encryption System. *Signal Process*, 141:217-227. https://doi.org/10.1016/j.sigpro.2017.06.010

ur Rehman A, Liao XF, Ashraf R, et al., 2018. A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2. *Optik*, 159:348-367. https://doi.org/10.1016/j.ijleo.2018.01.064

Wang JS, Long F, Ou WH, 2017. CNN-based color image encryption algorithm using DNA sequence operations. Int Conf on Security, Pattern Analysis, and Cybernetics, p.730-736. https://doi.org/10.1109/SPAC.2017.8304370

Wang XY, Zhang HL, Bao XM, 2016a. Color image encryption scheme using CML and DNA sequence operations. *Biosystems*, 144:18-26. https://doi.org/10.1016/j.biosystems.2016.03.011

Wang XY, Liu CM, Zhang HL, 2016b. An effective and fast image encryption algorithm based on chaos and interweaving of ranks. *Nonl Dynam*, 84(3):1595-1607. https://doi.org/10.1007/s11071-015-2590-3

Wu GC, Baleanu D, Lin ZX, 2016. Image encryption technique based on fractional chaotic time series. *J Vibr Contr*, 22(8):2092-2099. https://doi.org/10.1177/1077546315574649

Wu GC, Abdeljawad T, Liu JL, et al., 2019a. Mittag-Leffler stability analysis of fractional discrete-time neural networks via fixed point technique. *Nonl Anal Model Contr*, 24(6):919-936. https://doi.org/10.15388/NA.2019.6.5

Wu GC, Deng ZG, Baleanu D, et al., 2019b. New variable-order fractional chaotic systems for fast image encryption. *Chaos*, 29(8):083103. https://doi.org/10.1063/1.5096645

Wu XJ, Kan HB, Kurths J, 2015. A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps. *Appl Soft Comput*, 37:24-39. https://doi.org/10.1016/j.asoc.2015.08.008

Wu XJ, Wang KS, Wang XY, et al., 2017. Lossless chaotic color image cryptosystem based on DNA encryption and entropy. *Nonl Dynam*, 90(2):855-875. https://doi.org/10.1007/s11071-017-3698-4

Ye GD, Pan C, Huang XL, et al., 2018. An efficient pixel-level chaotic image encryption algorithm. *Nonl Dynam*, 94(1):745-756. https://doi.org/10.1007/s11071-018-4391-y

Zhang GJ, Liu Q, 2011. A novel image encryption method based on total shuffling scheme. *Opt Commun*, 284(12):2775-2780. https://doi.org/10.1016/j.optcom.2011.02.039

Zhang LM, Sun KH, Liu WH, et al., 2017. A novel color image encryption scheme using fractional-order hyper-chaotic system and DNA sequence operations. *Chin Phys B*, 26(10):100504. https://doi.org/10.1088/1674-1056/26/10/100504

Zhang LY, Li CQ, Wong KW, et al., 2012. Cryptanalyzing a chaos-based image encryption algorithm using alternate structure. *J Syst Softw*, 85(9):2077-2085. https://doi.org/10.1016/j.jss.2012.04.002

Zhang Q, Wei XP, 2013. A novel couple images encryption algorithm based on DNA subsequence operation and chaotic system. *Optik*, 124(23):6276-6281. https://doi.org/10.1016/j.ijleo.2013.05.009

Zhang Q, Liu LL, Wei XP, 2014. Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps. *AEU Int J Electron Commun*, 68(3):186-192. https://doi.org/10.1016/j.aeue.2013.08.007

Zhang R, Qi DW, Wang YZ, 2010. Dynamics analysis of fractional order three-dimensional Hopfield neural network. Proc 6th Int Conf on Natural Computation, p.3037-3039. https://doi.org/10.1109/ICNC.2010.5582371

Zhang Y, 2018. The unified image encryption algorithm based on chaos and cubic S-Box. *Inform Sci*, 450:361-377. https://doi.org/10.1016/j.ins.2018.03.055

Zhang YQ, Wang XY, 2015. A new image encryption algorithm based on non-adjacent coupled map lattices. *Appl Soft Comput*, 26:10-20. https://doi.org/10.1016/j.asoc.2014.09.039

Zhang YQ, Wang XY, Liu J, et al., 2016. An image encryption scheme based on the MLNCML system using DNA sequences. *Opt Lasers Eng*, 82:95-103. https://doi.org/10.1016/j.optlaseng.2016.02.002

Zhang YS, Xiao D, 2014. Self-adaptive permutation and combined global diffusion for chaotic color image encryption. *AEU Int J Electron Commun*, 68(4):361-368. https://doi.org/10.1016/j.aeue.2013.10.002

Zheng XD, Xu J, Li W, 2009. Parallel DNA arithmetic operation based on n-moduli set. *Appl Math Comput*, 212(1):177-184. https://doi.org/10.1016/j.amc.2009.02.011

Zhou YC, Bao L, Chen CLP, 2014. A new 1D chaotic system for image encryption. *Signal Process*, 97:172-182. https://doi.org/10.1016/j.sigpro.2013.10.034

Zhou YC, Hua ZY, Pun CM, et al., 2015. Cascade chaotic system with applications. *IEEE Trans Cybern*, 45(9):2001-2012. https://doi.org/10.1109/TCYB.2014.2363168

Zhu ZL, Zhang W, Wong KW, et al., 2011. A chaos-based symmetric image encryption scheme using a bit-level permutation. *Inform Sci*, 181(6):1171-1186. https://doi.org/10.1016/j.ins.2010.11.009