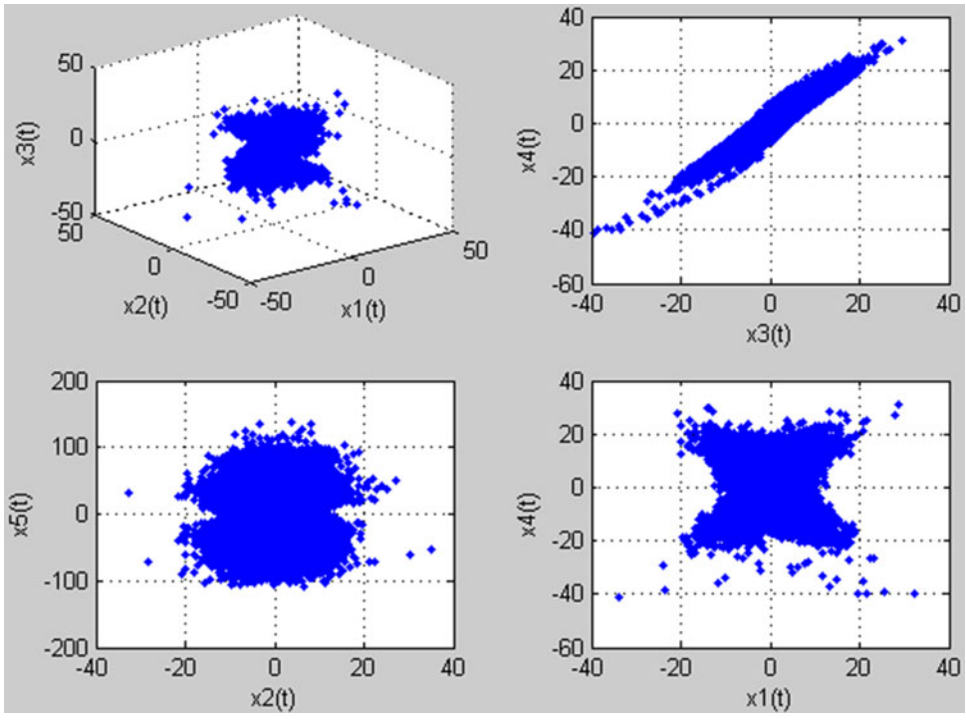
IEEE Photonics Journal
An IEEE Photonics Society Publication

A Novel Hyperchaotic Image Encryption Scheme Based on DNA Encoding, Pixel-Level Scrambling and Bit-Level Scrambling

Volume 10, Number 2, April 2018

Shuliang Sun



DOI: 10.1109/JPHOT.2018.2817550
1943-0655 © 2018 IEEE

A Novel Hyperchaotic Image Encryption Scheme Based on DNA Encoding, Pixel-Level Scrambling and Bit-Level Scrambling

Shuliang Sun

School of Electronics and Information Engineering, Fuqing Branch of Fujian Normal University, Fuqing 350300, China, and
Innovative Information Industry Research Center, Fuqing Branch of Fujian Normal University, Fuqing 350300, China

DOI:10.1109/JPHOT.2018.2817550

Manuscript received January 4, 2018; revised February 19, 2018; accepted March 15, 2018. Date of publication March 20, 2018; date of current version April 10, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61272469, in part by the Natural Science Foundation of Fujian Province of China under Grant 2016J05153, in part by the Outstanding Youth Scientific Research Training Program of Fujian Province of China in 2017, and in part by the Training Program for Talents from the Fuqing Branch of Fujian Normal University of China under Grant KY2017NS07. (e-mail: tjussl_07@126.com).

Abstract: In this paper, a novel image encryption scheme has been proposed using pixel-level scrambling, bit-level scrambling, and DNA encoding. First, initial conditions of five-dimensional hyperchaotic system are computed and chaotic sequences are generated. Then, pixel-level scrambling and bit-level scrambling are implemented to permute the plain image. Permuted image and generated pseudorandom sequence are executed decomposition operations in order to enhance security. DNA encoding, DNA XOR operation, and DNA complementary rules are also adopted to improve the security of the cryptosystem. Experiments results and theoretical analysis show that the proposed scheme is secure enough and can resist known plain text attack, statistical attacks, and differential attacks. It is suitable for practical application.

Index Terms: 5-D hyperchaotic system, bit-level scrambling, DNA encoding, decomposition operation.

1. Introduction

With the development of Internet and communication technology, information security has been paid more and more attention. The traditional encryption algorithms are not suitable for image encryption [1]. Chaotic system is famous for sensitivity to initial conditions and parameters, pseudo-randomness, ergodicity and reproduction [2]. It is very suitable for image encryption and many chaotic encryption schemes have been proposed. Ravichandran *et al*. [3] presented a cryptosystem which could be suitable for both selective and full medical image encryption. Wu *et al*. [4] proposed a new image randomness measure using Shannon entropy over local image blocks. Wang and Zhang [5] put forward a novel color image encryption with heterogeneous bit-permutation and correlated chaos. Heterogeneous bit-permutation was performed to reduce computation cost and improve permutation efficiency. Correlated chaos could make fully use of chaotic maps. Liu *et al*. [6]

proposed a cryptosystem based on two-dimensional Sine ICMIC modulation map. The confusion and diffusion processes were combined together. Chaotic shift transform was proposed to efficiently change the image pixel positions. Wang *et al*. [7] presented a fast image encryption method which was based on rows and columns switch. Sivakumar and Venkatesan [8] proposed a new image encryption method based on knight's travel path and true random number. Authors in [9] presented an efficient image encryption using quaternary coding. Quaternary coding was used to o split the plain image into four sub-sections so that the cipher image could not be formed without any one sub-section.

Due to high parallelism, huge storage and ultra-low power consumption, some DNA-based encryption methods have been proposed nowadays. Khalifa and Atito [10] proposed a steganography algorithm based on Playfair cipher and two-by-two DNA complementary rule. Liu *et al*. [11] analyzed a RGB image encryption scheme based on DNA encoding and chaos map. They found that the scheme could be broken with only four chosen plain-images and corresponding cipher-images. Two other defeats were also pointed out. Wang *et al*. [12] presented a hybrid image encryption based on 2-D chaotic sequence and DNA encoding. Rehman *et al*. [13] proposed a method for gray images based on chaos and DNA complementary rules. The most significant and least significant parts of each block were encoded with different methods. Jain and Rajpal [14] designed a robust image encryption scheme using DNA and logistic chaotic maps. The original image was DNA encoded and a mask was generated with 1D chaotic map. DNA addition and DNA complementary rules were also adopted. A novel image encryption scheme based on DNA sequence operations and chaotic system was proposed in [15]. The plain image was confused using the pseudorandom sequences firstly. Then one of DNA encoding rules was used to obtain DNA matrix. Thirdly, the rows and columns of DNA matrix were permuted. Zhang *et al*. [16] put forward a novel image encryption using DNA addition and chaotic maps. DNA sequence matrix was divided into many equal blocks and these blocks were executed DNA addition operation. DNA complementary operation was also applied in this scheme.

The rest of the paper is organized as follows. Section 2 briefly describes 5-D hyperchaotic system and DNA coding. Section 3 presents the proposed encryption and decryption scheme. Section 4 shows the experimental results and analysis. Section 5 depicts security analysis and conclusion is described in Section 6.

2. Preliminary Works

2.1 5-D hyperchaotic System

5-D hyperchaotic system [17], [18] could be expressed as follows:

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) + x_2 x_3 x_4 \\ \dot{x}_2 = b(x_1 + x_2) + x_5 - x_1 x_3 x_4 \\ \dot{x}_3 = -cx_2 - dx_3 - ex_4 + x_1 x_2 x_4 \\ \dot{x}_4 = -fx_4 + x_1 x_2 x_3 \\ \dot{x}_5 = -g(x_1 + x_2) \end{cases}, \tag{1}$$

where a, b, c, d, e, f, g are system control parameters. If the parameters are assigned as $a = 30$, $b = 10$, $c = 15.7$, $d = 5$, $e = 2.5$, $f = 4.45$ and $g = 38.5$, the 5-D hyperchaotic system is in a chaotic state and could produce five chaotic sequences. Sequence trajectories of system (1) are displayed in Fig. 1.

2.2 DNA Coding

Deoxyribonucleic Acid (DNA) is two twisted strands composed of four bases, adenine (A), cytosine (C), thymine (T) and guanine (G). It's known that A and T are complementary, and G and C are also complementary according to Watson-Crick complement rule [19]. A, C, G and T could be

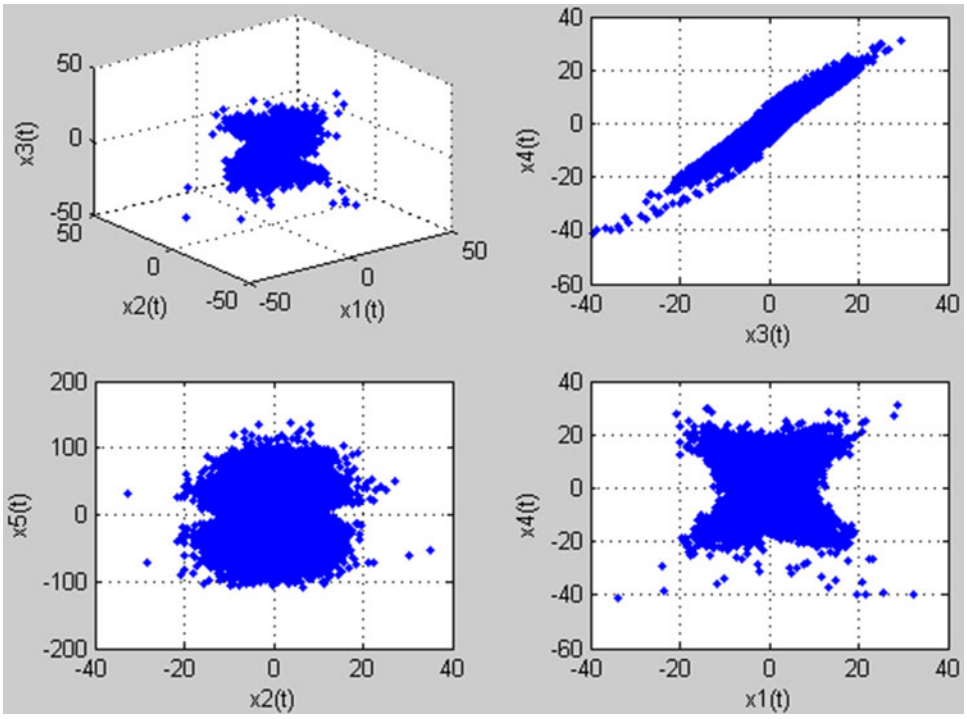


Fig. 1. Sequence trajectories of system (1) with parameters $a = 30$, $b = 10$, $c = 15.7$, $d = 5$, $e = 2.5$, $f = 4.45$ and $g = 38.5$.

TABLE 1
DNA Ex-OR Operation

⊕	A	C	G	T
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A

represented as 00 (0), 01(1), 10(2) and 11(3). In the 8-bit grayscale image, each pixel is denoted by a DNA sequence of length 4. DNA Ex-OR operation is shown as Table 1.

The DNA complementary rule must satisfy that [20]:

$$\begin{cases} x \neq E(x) \neq E(E(x)) \neq E(E(E(x))) \\ x = E(E(E(E(x)))) \end{cases} \tag{2}$$

where $E(x)$ is the base pair of x which is different from x at least one bit.

There are six major complementary rules for each letter of DNA sequence. For all letter x, $E(x)$, $E(E(x))$, $E(E(E(x)))$ is not equal.

1. A → T, T → C, C → G, G → A
2. A → T, T → G, G → C, C → A
3. A → C, C → T, T → G, G → A
4. A → C, C → G, G → T, T → A
5. A → G, G → T, T → C, C → A
6. A → G, G → C, C → T, T → A

3. The Proposed Image Encryption and Decryption Scheme

The chaotic sequences are generated by 5-D hyperchaotic system (1). Suppose the size of the plain image P is $M \times N$. Then, a pixel-level and bit-level scrambling are adopted to permute the plain image. DNA encoding operation is utilized and cipher image H is obtained finally.

3.1 Pixel-Level Scrambling

Step 1: Compute the initial values x_1, x_2, x_3, x_4 and x_5 of 5-D hyperchaotic system (1) as follows:

$$\begin{cases} x_1(1) = \mathrm{mod}\left(\sum_{j=1}^{5} x_j^0, 1\right) \\ x_i(1) = \mathrm{mod}\left(x_{i-1}(1) + x_i^0, 1\right) \quad i = 2, 3, 4, 5 \end{cases}, \tag{3}$$

where x_1^0, x_2^0, x_3^0, x_4^0, x_5^0 are the initial keys, and mod (x, y) means the residue of x divided by y.

Step 2: Iterate 5-D hyperchaotic system N_0 times to avoid the transient effect. Continue to iterate 5-D hyperchaotic system MN times and get 3 chaotic sequences k_1, k_2 and k_3. Especially $k_l = [k_l(1), k_l(2), \ldots, k_l(MN)]$, $l = 1, 2, 3$.

$$N_0 = 200 + \mathrm{mod}\left(\left(\left(\sum_{i=1}^{5} x_i^0\right) - \left\lfloor \sum_{i=1}^{5} x_i^0 \right\rfloor\right) \times 10^{15}, 200\right), \tag{4}$$

Step 3: Suppose (i, j) and (i', j') are the positions of original plain image P. Corresponding scrambling image is denoted as P', and it could be computed as follows:

$$\begin{cases} i' = i + \mathrm{mod}\left((abs(k_1(i)) - \lfloor abs(k_1(i)) \rfloor) \times 10^{15}, M - i\right) \\ j' = j + \mathrm{mod}\left((abs(k_2(j)) - \lfloor abs(k_2(j)) \rfloor) \times 10^{15}, N - j\right) \end{cases}, \tag{5}$$

where $\lfloor x \rfloor$ rounds x to the nearest integer less than or equal to x.

Step 4: The scrambling operation is operated as

$$P'(i, j) = P(i', j'), \; P(i', j') = P(i, j), \tag{6}$$

Where $P'(i, j)$ is the scrambling image positioned at (i, j), $P(i', j')$ and $P(i, j)$ are the original image positioned at (i', j') and (i, j), $i = 1, 2, \ldots, M$; $j = 1, 2, \ldots, N$.

3.2 Bit-Level Scrambling

Step 1: Convert diffused image matrix P' to one-dimensional sequence $P' = [P'(1), P'(2), \ldots, P'(MN)]$ from upper-left corner to lower-right corner.

Step 2: The chaotic sequence k_3 is transformed as formula (7):

$$k_3'(r) = \mathrm{mod}\left((abs(k_3(r)) - \lfloor abs(k_3(r)) \rfloor) \times 10^{15}, 8\right), \tag{7}$$

where $k_3'(r) \in [0, 7]$ and $r = 1, 2, \ldots, MN$.

Step 3: Transform the decimal sequences P' and k_3' into corresponding binary sequences.

Step 4: The scrambled sequence C will be computed as (8).

$$C(r) = \mathrm{circshift}[P'(r), LSB(k_3'(r)), k_3'(r)], \tag{8}$$

where circshift $[u, q, v]$ means v-bit cyclic shift on the binary sequences u. $LSB(z)$ means the least bit of z. A right cyclic shift or a left cyclic shift will be decided by $q = 1$ or $q = 0$ [21].

Step 5: Convert binary sequence C to its decimal sequence.

3.3 DNA Encoding Scheme

Step 1: Compute the initial values x'_1, x'_2, x'_3, x'_4, and x'_5 of 5-D hyperchaotic system (1) as follows:

$$\begin{cases} x'_1(1) = \mathrm{mod}\left(\sum_{j=1}^{6} x_j^0, 1\right) \\ x'_i(1) = \mathrm{mod}\left(x'_{i-1}(1) + x_i^0, 1\right) \quad i = 2, 3, 4, 5 \end{cases}, \tag{9}$$

where x_j^0 is the initial key, $j = 1, 2, \ldots, 6$.

Step 2: Iterate 5-D hyperchaotic system N'_0 times to avoid the transient effect. Continue to iterate 5-D hyperchaotic system $4MN$ times and get 4 chaotic sequences a_1, a_2, a_3 and a_4. Especially $a_l = [a_l(1), a_l(2), \ldots, a_l(4MN)]$, $l = 1, 2, 3, 4$.

$$N'_0 = 200 + \mathrm{mod}\left(\left(\left(\sum_{i=1}^{6} x_i^0\right) - \left\lfloor \sum_{i=1}^{6} x_i^0 \right\rfloor\right) \times 10^{15}, 200\right), \tag{10}$$

Step 3: The chaotic sequences a_1, a_2, a_3 and a_4 are performed as (11)–(14).

$$a_1(i) = \mathrm{mod}\left(\left(abs(a_1(i)) - \lfloor abs(a_1(i))\rfloor\right) \times 10^{15}, 6\right) + 1, \tag{11}$$

$$a_2(i) = \mathrm{mod}\left(\left(abs(a_2(i)) - \lfloor abs(a_2(i))\rfloor\right) \times 10^{15}, 4\right), \tag{12}$$

$$a_3(i) = \mathrm{mod}\left(\left(abs(a_3(i)) - \lfloor abs(a_3(i))\rfloor\right) \times 10^{15}, 256\right), \tag{13}$$

$$a_4(i) = \mathrm{mod}\left(\left(abs(a_4(i)) - \lfloor abs(a_4(i))\rfloor\right) \times 10^{15}, 256\right), \tag{14}$$

where $a_1 \in [1, 6]$, $a_2 \in [0, 3]$, $a_3 \in [0, 255] \in$, $a_4 \in [0, 255] \in$, $i = 1, 2, \ldots, 4MN$.

Step 4: For each $C(r)$ and $a_3(r)$ implement the following decomposition operation:

$$C(r) = \sum_{s=0}^{3} c_{4r-s}(r) \cdot 4^s,\ c_{4r-s} \in \{0, 1, 2, 3\}$$

$$a_3(r) = \sum_{s=0}^{3} d_{4r-s}(r) \cdot 4^s,\ d_{4r-s} \in \{0, 1, 2, 3\}, \tag{15}$$

where $r = 1, 2, \ldots, MN$. Then, the sequences $\{c(i)\}_{i=1}^{4MN}$ and $\{d(i)\}_{i=1}^{4MN}$ can be constructed.

Step 5: Convert $\{c(i)\}_{i=1}^{4MN}$ and $\{d(i)\}_{i=1}^{4MN}$ into DNA sequences $\{c'(i)\}_{i=1}^{4MN}$ and $\{d'(i)\}_{i=1}^{4MN}$.
Step 6: Perform the DNA EX-OR to get the DNA sequence F.

$$F(i) = c'(i) \oplus d'(i), \tag{16}$$

where $i = 1, 2, \ldots, 4MN$.

Step 7: Select a rule from six complementary rules according $a_1(i)$. Based on $a_2(i)$ and selected complementary rule, perform DNA replacement operation on DNA sequence $F(i)$ and obtain DNA complementary sequence $F'(i)$.

$$F'(i) = E^{a_2(i)}(F(i)) = \begin{cases} F(i), & \text{if} a_2(i) = 0 \\ E(F(i)), & \text{if } a_2(i) = 1 \\ E(E(F(i))), & \text{if } a_2(i) = 2 \\ E(E(E(F(i)))), & \text{if } a_2(i) = 3 \end{cases}, \tag{17}$$

where $E(x)$ is the base pair of x, $i = 1, 2, \ldots, 4MN$.

Step 8: Decode F' to binary sequence G, and convert G to decimal sequence H.

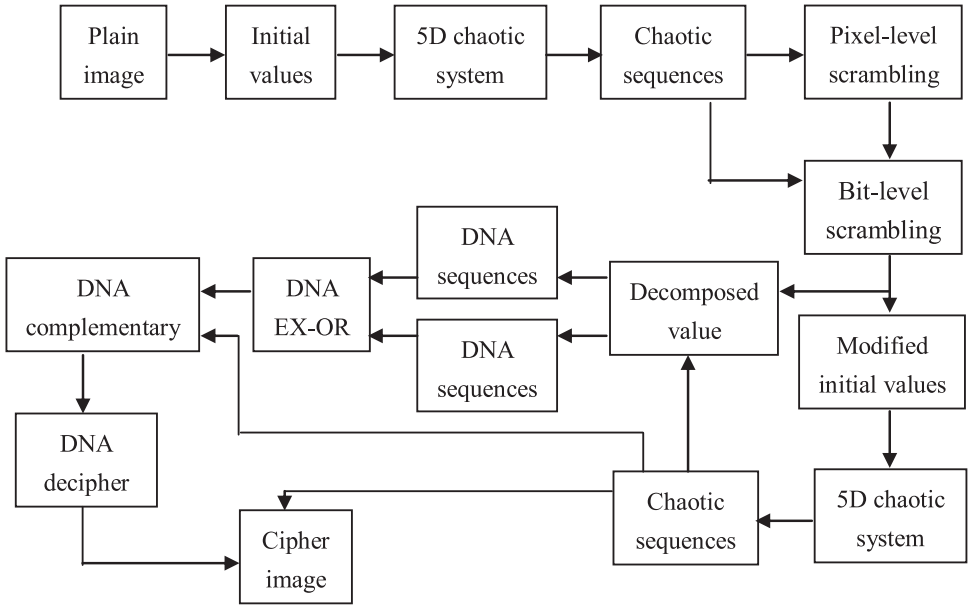


Fig. 2. Flowchart of image encryption procedure.

Step 9: Cipher image R is obtained finally as (18) [22].

$$R(1) = a_4(1) \oplus \mathrm{mod}\left(a_4(1) + H(1), 256\right) \oplus \mathrm{mod}\left(\sum_{j=1}^{6} x_j^0 \times 10^{15}, 256\right), \tag{18}$$

$$R(i) = a_4(i) \oplus \mathrm{mod}(a_4(i) + H(i), 256) \oplus R(i-1), \tag{19}$$

where $H(i)$, $a_4(i)$, $R(i)$ and $R(i-1)$ respectively mean decimal sequence value, chaotic sequence value, output cipher pixel and the previous cipher pixel, $i = 2, 3, \ldots, MN$.

Flowchart of image encryption procedure is shown in Fig. 2.

3.4 The Decryption Process

The decryption procedure is reversion of the encryption procedure and will be described briefly as follows:

Step 1: The chaotic sequences will be generated by chaotic system (1).
Step 2: Decimal sequence H will be obtained by formula (20), (21):

$$H(1) = \mathrm{mod}(a_4(1) \oplus R(1) \oplus \mathrm{mod}\left(\sum_{j=1}^{6} x_j^0 \times 10^{15}, 256\right) - a_4(1), 256), \tag{20}$$

$$H(i) = \mathrm{mod}(a_4(i) \oplus R(i) \oplus R(i-1) - a_4(i), 256), \tag{21}$$

Step 3: Perform DNA inverse replacement operation to get DNA sequence $F(i)$ based on chaotic sequence and selected complementary rule.
Step 4: Convert DNA sequence to bit-level scrambling sequence C.
Step 5: Convert sequence C to pixel-level scrambled image P'.
Step 6: Transform scrambled image P' to original image P.

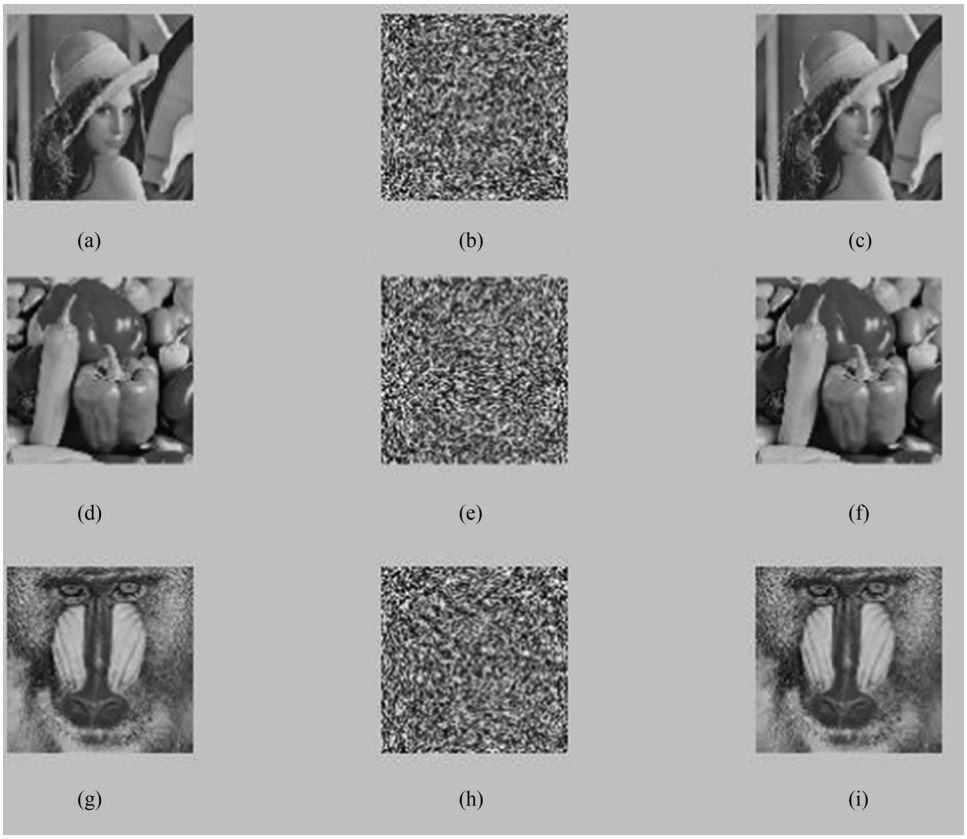


Fig. 3. The plain, encrypted and decrypted images of "Lena"(a)–(c), "Peppers"(d)–(f) and "Baboon" (g)–(i). (a) plain image. (b) cipher image. (c) decrypted image. (d) plain image. (e) cipher image. (f) decrypted image. (g) plain image. (h) cipher image. (i) decrypted image.

4. Simulation Results

In this paper, MATLAB 2010 is applied to execute the algorithm. The initial values of the 5-D chaotic system are $x_1^0 = 1.2356$, $x_2^0 = 2.8905$, $x_3^0 = 0.89648$, $x_4^0 = 3.45797$, $x_5^0 = 0.45723$, $x_6^0 = 3.2579$. The 256×256 grayscale images "Lena", "Peppers" and "Baboon" are used as the plain images. The plain, encrypted and decrypted images are shown in Fig. 3.

5. Security Analysis

5.1 Key Space

Key space of the proposed scheme is decided on the initial values of the hyperchaotic system $\{x_i^0, i = 1, 2, \ldots, 6\}$. The precision of each initial value is 10^{-15}, so the key space is about $(10^{15})^6 = 10^{90} \approx 2^{298}$. If a key space of image cryptosystem is more than 2^{100}, it could withstand an exhaustive attack [23], [24]. So the key space of proposed scheme is large enough to resist brute-force attack.

5.2 Key Sensitivity Analysis

An excellent cryptosystem should be sensitive to the initial key.

5.2.1 Key Sensitivity of Encryption Procedure: Firstly, the image encryption is executed with initial values ($x_1^0 = 1.2356$, $x_2^0 = 2.8905$, $x_3^0 = 0.89648$, $x_4^0 = 3.45797$, $x_5^0 = 0.45723$, $x_6^0 = 3.2579$) to produce a cipher image. Then a tiny alteration (10^{-15}) is brought in one of the initial values while

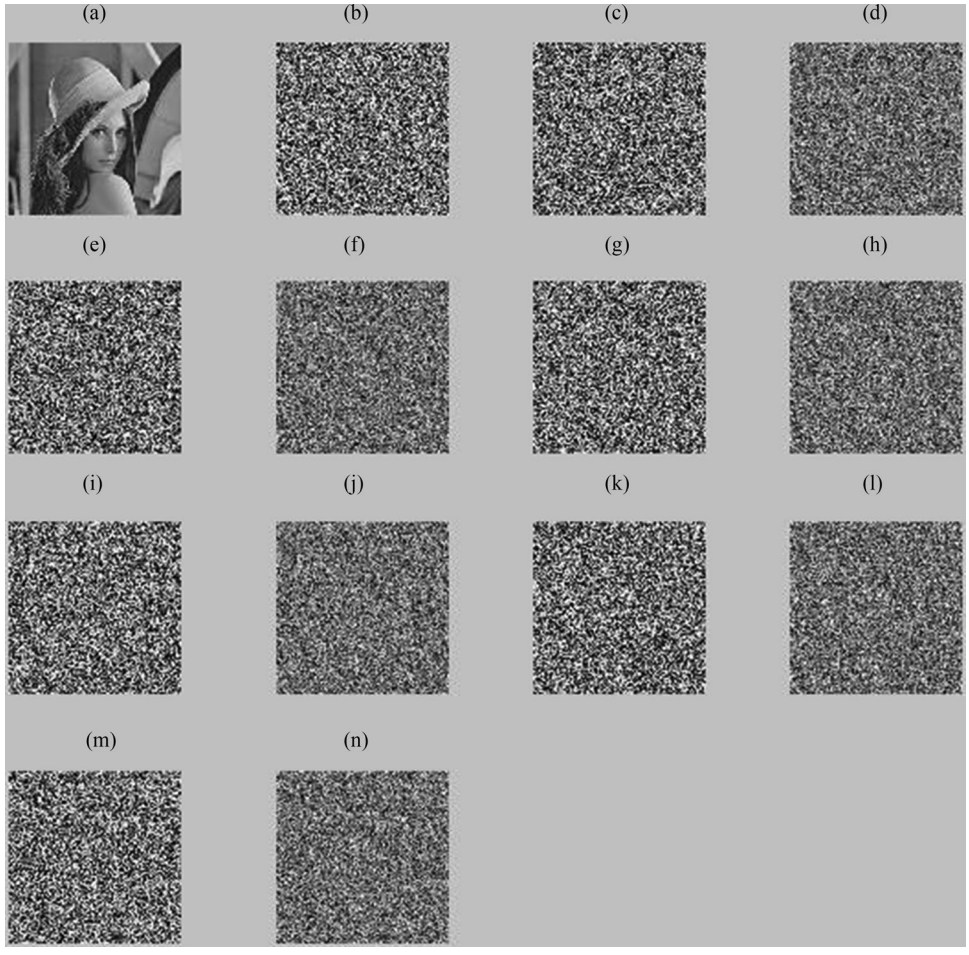


Fig. 4. The decryption results with different keys: (a) plain image; (b) encrypted image using the original key; (c) encrypted image with $x_1^0 + 10^{-15}$; (d) differential image between (c) and (b); (e) encrypted image with $x_2^0 + 10^{-15}$; (f) differential image between (e) and (b); (g) encrypted image with $x_3^0 + 10^{-15}$; (h) differential image between (g) and (b); (i) encrypted image with $x_4^0 + 10^{-15}$; (j) differential image between (i) and (b); (k) encrypted image with $x_5^0 + 10^{-15}$; (l) differential image between (k) and (b); (m) encrypted image with $x_6^0 + 10^{-15}$; (n) differential image between (m) and (b).

others remain the same, and performs the encryption process again. The cipher images and the differential images are depicted in Fig. 4. Table 2 demonstrates the differences between different cipher images. It can be seen that a slight difference in the initial secret key will produce totally different cipher image.

5.2.2 Key Sensitivity of Decryption Procedure: The encrypted image should be also sensitive to the initial key in decryption phase. The decrypted image in Fig. 4(b) is adopted. The decrypted images are depicted in Fig. 5. The differences between improper decrypted images (Fig. 5(c)–(h)) and the plain image are almost 99.6%. So the proposed scheme is very sensitive to the system key.

5.3 The Histogram Analysis

The histogram of encrypted image should be as flat as possible. In proposed scheme, the histograms of the plain and cipher images of Lena, Peppers and Baboon are displayed in Fig. 6. It is shown that

TABLE 2
Difference between Encrypted Images With Tiny Alteration Keys

Figure	Initial Keys						Difference between 4(b)
	x_1	x_2	x_3	x_4	x_5	x_6	
4(b)	x_1^0	x_2^0	x_3^0	x_4^0	x_5^0	x_6^0	–
4(c)	$x_1^0 + 10^{-15}$	x_2^0	x_3^0	x_4^0	x_5^0	x_6^0	0.9966
4(e)	x_1^0	$x_2^0 + 10^{-15}$	x_3^0	x_4^0	x_5^0	x_6^0	0.9963
4(g)	x_1^0	x_2^0	$x_3^0 + 10^{-15}$	x_4^0	x_5^0	x_6^0	0.9958
4(i)	x_1^0	x_2^0	x_3^0	$x_4^0 + 10^{-15}$	x_5^0	x_6^0	0.9959
4(k)	x_1^0	x_2^0	x_3^0	x_4^0	$x_5^0 + 10^{-15}$	x_6^0	0.9959
4(m)	x_1^0	x_2^0	x_3^0	x_4^0	x_5^0	$x_6^0 + 10^{-15}$	0.9962

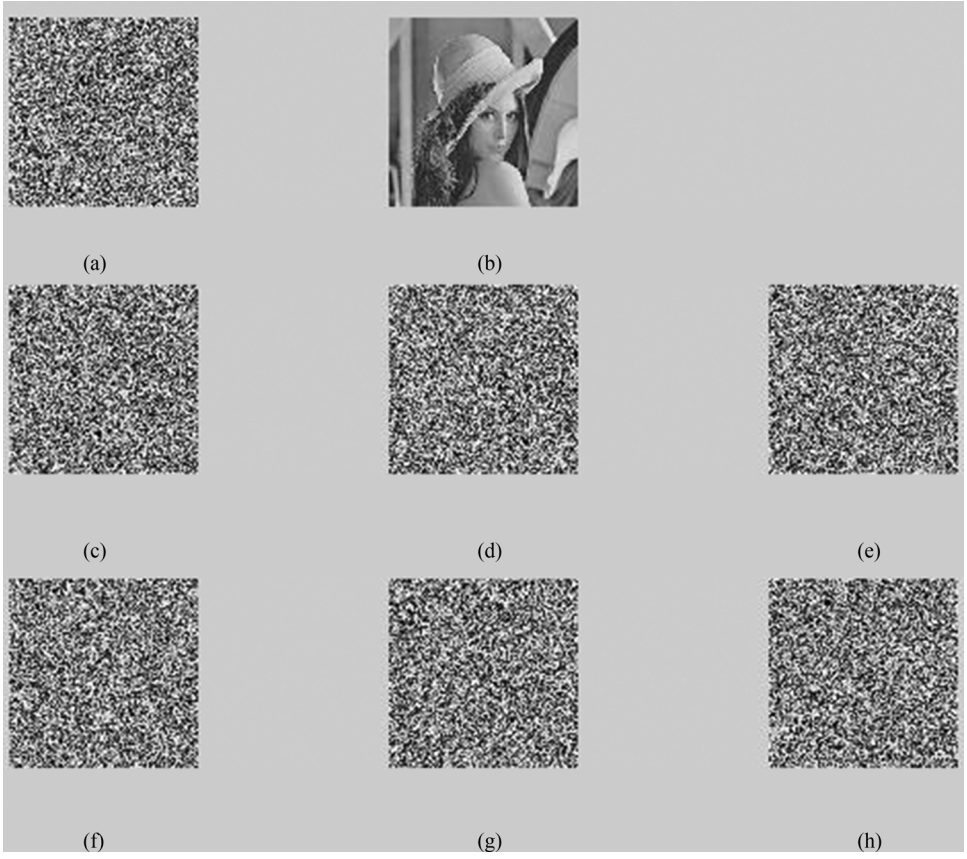


Fig. 5. The decryption results with different keys: (a) encrypted image using the original key; (b) decrypted image using the right key; (c) decrypted image with $x_1^0 + 10^{-15}$; (d) decrypted image with $x_2^0 + 10^{-15}$; (e) decrypted image with $x_3^0 + 10^{-15}$; (f) decrypted image with $x_4^0 + 10^{-15}$; (g) decrypted image with $x_5^0 + 10^{-15}$; (h) encrypted image with $x_6^0 + 10^{-15}$.

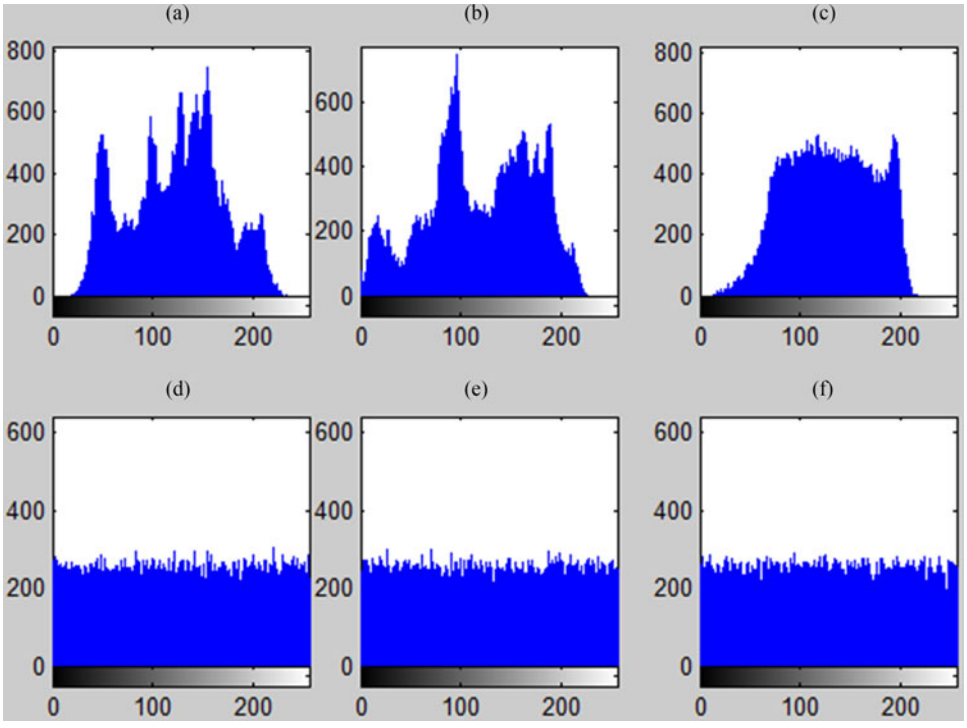


Fig. 6. The histograms of the plain and encrypted images: (a) plain image "Lena"; (b) plain image "Peppers"; (c) plain image "Baboon"; (d) encrypted image "Lena"; (e) encrypted image "Peppers"; (f) encrypted image "Baboon".

the histogram of the plain image pixel values is centralized some values, however the histogram of corresponding cipher image pixel values is very flat. So it could withstand statistical attacks.

5.4 Correlation Analysis

The correlation coefficient r_{xy} between two adjacent pixels x and y is computed as

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)D(y)}} \tag{22}$$

where $\mathrm{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$, $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$, $D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$ 7225 pairs of adjacent pixels from plain image and encrypted image are selected in the horizontal, vertical and diagonal directions. Fig. 7 shows the correlation of two adjacent pixels in the original Lena image and its encrypted image. It can be shown that pixels are highly correlated in plain image whereas correlation is greatly reduced in the encrypted image.

Table 3 shows the result of correlation coefficients of two adjacent pixels in Fig. 4(a), (b), which is compared with the results in [19], [20]. The results reveal that the proposed algorithm is much better than two other methods.

5.5 Information Entropy

Information entropy is one of the most important features of randomness. If m is the information source and information entropy could be calculated as follows:

$$H(m) = -\sum_{i=1}^{L} p(m_i)\log_2 p(m_i) \tag{23}$$

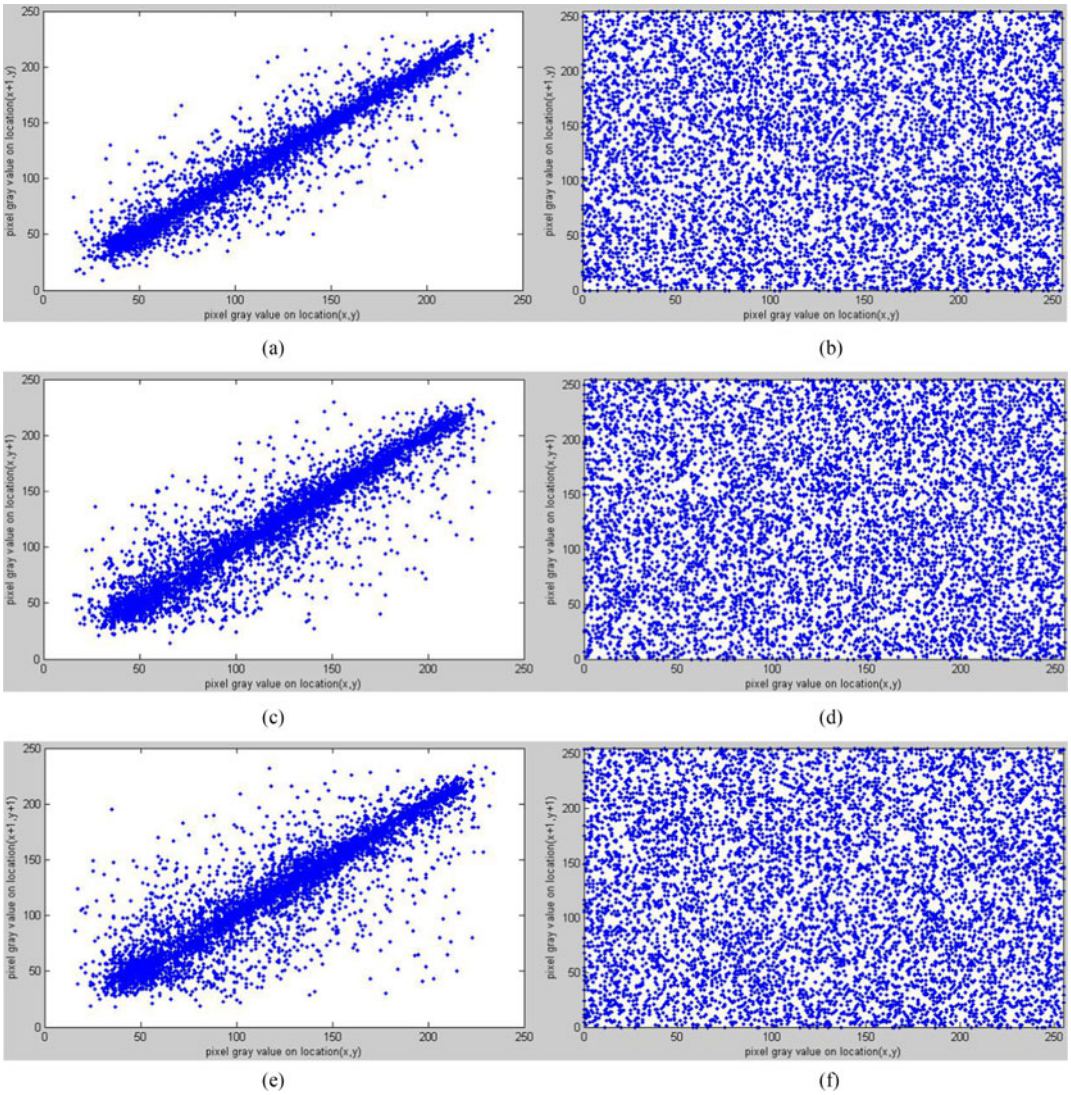


Fig. 7. Correlation distribution in different directions of plain image and its encrypted image. (a) Horizontal direction for original image. (b) Horizontal direction for encrypted image. (c) Vertical direction for original image. (d) Vertical direction for encrypted image. (e) Diagonal direction for original image. (f) Diagonal direction for encrypted image.

TABLE 3
Comparison of Correlation Coefficients

Algorithm	Horizontal	Vertical	Diagonal
Plain image	0.9391	0.9700	0.9146
Proposed	0.0068	−0.0054	0.0010
[19]	0.0211	0.0412	−0.0016
[20]	0.0082	−0.0107	0.0022

TABLE 4
Information Entropy of Encrypted Images

Image	Lena	Peppers	Baboon
proposed	7.9967	7.9967	7.9976
[25]	7.7893	7.7897	7.9966
[15]	7.9962	7.9961	7.9969

TABLE 5
NPCR and UACI of Proposed Method and Other Schemes

Image	NPCR (%)	UACI (%)
proposed	99.61	33.46
[9]	99.57	33.45
[26]	99.54	33.43

where $p(m_i)$ means the probability of symbol m_i, and L is the total number of m_i. The maximum information entropy is 8 for grayscale image. The information entropy of encrypted images is shown in Table 4.

5.6 Differential Attack

Number of pixels change rate (NPCR) and unified average changing intensity (UACI) are two indicators which are often applied to measure the sensitivity to plaintext. They are used to test ability to resist differential attack. NPCR and UACI are defined in (24)–(26) [16].

$$\mathrm{NPCR} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\% \tag{24}$$

$$\mathrm{UACI} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\% \tag{25}$$

$$D(i,j) = \begin{cases} 0, \text{ if } C_1(i,j) = C_2(i,j) \\ 1, \textit{else} \end{cases} \tag{26}$$

where M and N denote the width and height of the image; C_1 and C_2 represent the ciphered images before and after one pixel of the plain image is changed.

Table 5 shows the information entropy of proposed method and other schemes. It can be concluded that the proposed algorithm could effectively resist plaintext attack and differential attack.

5.7 Time Complexity Analysis

Time speed is another important factor to measure the performance the cryptosystem. The running speed of the proposed scheme is calculated with the Peppers image for different sizes and it is compared with other algorithms. The results are displayed in Table 6.

TABLE 6
Values of Running Speed With Different Schemes

Image	Time (s)
proposed	0.4862
[25]	3.624
[26]	0.5683

From Table 6, it can be concluded that the running time of proposed scheme is shorter than that of others. Therefore, proposed scheme is efficient.

6. Conclusion

In this paper, a novel hyperchaotic image encryption algorithm is proposed based on pixel-level scrambling, bit-level scrambling and DNA encoding. Firstly, the chaotic sequences are generated by 5-D hyperchaotic system. Then pixel-level scrambling and bit-level scrambling are operated to confuse the plain image. In order to enhance the security of the cryptosystem and increase the complexity of information, the permuted image is executed decomposition operation and DNA encoding. DNA XOR operation and DNA complementary rules are also applied to improve the ability of resisting plaintext attacks. Experimental results and theoretical analysis prove that the algorithm could resist differential attack, brute-force attack, statistical attack and plaintext attack. Therefore it has extraordinarily high security and is reliable for practical application.

References

[1] Z. Zhu, W. Zhang, K. W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Inf. Sci.*, vol. 181, no. 6, pp. 1171–1186, 2011.
[2] D. Ravichandran, P. Praveenkumar, J. B. Balaguru Rayappan, and R. Amirtharaja, "Chaos based crossover and mutation for securing DICOM image," *Comput. Biol. Med.*, vol. 72, pp. 170–184, 2016.
[3] D. Ravichandran, P. Praveenkumar, J. B. B. Rayappan, and R. Amirtharajan, "DNA chaos blend to secure medical privacy," *IEEE Trans. Nanobiosci.*, vol. 16, no. 8, pp. 850–858, Dec. 2017.
[4] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Inf. Sci.*, vol. 222, pp. 323–342, 2013.
[5] X. Wang and H. Zhang, "A color image encryption with heterogeneous bit-permutation and correlated chaos," *Opt. Commun.*, vol. 342, pp. 51–60, 2015.
[6] W. Liu, K. Sun, and C. Zhu, "A fast image encryption algorithm based on chaotic map," *Opt. Lasers Eng.*, vol. 84, pp. 26–36, 2016.
[7] X. Wang, Q. Wang, and Y. Zhang, "A fast image algorithm based on rows and columns switch," *Nonlinear Dyn.*, vol. 79, no. 2, pp. 1141–1149, 2015.
[8] T. Sivakumar and R. Venkatesan, "A new image encryption method based on knight's travel path and true random number," *J. Inf. Sci. Eng.*, vol. 32, no. 1, pp. 133–152, 2016.
[9] H. Niu, C. Zhou, B. Wang, X. Zheng, and S. Zhou, "Splicing model and hyper–chaotic system for image encryption," *J. Elect. Eng.*, vol. 67, no. 2, pp. 78–86, 2016.
[10] A. Khalifa and A. Atito, "High-capacity DNA-based steganography," in *Proc. 8th Int. Conf. Informat. Syst.*, 2012, pp. BIO-76–BIO-80.
[11] Y. Liu, J. Tang, and T. Xie, "Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map," *Opt. Laser Technol.*, vol. 60, no. 2, pp. 111–115, 2014.
[12] X. Wang, Y. Zhang, and Y. Zhao, "A novel image encryption scheme based on 2-D logistic map and DNA sequence operations," *Nonlinear Dyn.*, vol. 82, no. 3, pp. 1269–1280, 2015.
[13] A. Rehman, X. Liao, A. Kulsoom, and S. A. Abbas, "Selective encryption for gray images based on chaos and DNA complementary rules," *Multimedia Tools Appl.*, vol. 74, no. 13, pp. 4655–4677, 2015.
[14] 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, 2016.

[15] X. Wang, Y. Zhang, and X. Bao, "A novel chaotic image encryption scheme using DNA sequence operations," *Opt. Lasers Eng.*, vol. 73, pp. 53–61, 2015.
[16] Q. Zhang, L. Guo, and X. Wei, "Image encryption using DNA addition combining with chaotic maps," *Math. Comput. Model.*, vol. 52, no. 11/12, pp. 2028–2035, 2010.
[17] H. Yuan, Y. Liu, T. Lin, T. Hu, and L. Gong, "A new parallel image cryptosystem based on 5D hyper-chaotic system," *Signal Process., Image Commun.*, vol. 52, no. C, pp. 87–96, 2017.
[18] Y. Li, C. Wang, and H. Chen, "A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Opt. Lasers Eng.*, vol. 90, pp. 238–246, 2017.
[19] S. Sun, "A novel secure image steganography using improved logistic map and DNA techniques," *J. Internet Technol.*, vol. 18, no. 3, pp. 647–652, 2017.
[20] S. Sun, "Chaotic image encryption scheme using Two-by-two deoxyribonucleic acid complementary rules," *Opt. Eng.*, vol. 56, no. 11, 2017, Art. no. 116117.
[21] J. Chen, Z. Zhu, C. Fu, L. Zhang, and Y. Zhang, "An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 23, no. 1–3, pp. 294–310, 2015.
[22] X. Chai, Z. Gan, and M. Zhang, "A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion," *Multimedia Tools Appl.*, vol. 76, no. 14, pp. 15561–15585, 2017.
[23] R. Enayatifar, A. H. Abdullah, I. F. Isnin, A. Altameem, and M. Lee, "Image encryption using a synchronous permutation-diffusion technique," *Opt. Lasers Eng.*, vol. 90, pp. 146–154, 2017.
[24] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber J., Multidisciplinary J. Sci. Technol., J. Sel. Areas Telecommun.*, vol. 2, pp. 31–38, 2011.
[25] 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, no. 21, pp. 17–25, 2016.
[26] M. Brindha and N. A. Gounden, "A chaos based image encryption and lossless compression algorithm using hash table and Chinese remainder theorem," *Appl. Soft Comput.*, vol. 40, pp. 379–390, 2016.