On the cryptanalysis of an image encryption algorithm with quantum chaotic map and DNA coding

Xin Chen[1], Simin Yu[1†], Qianxue Wang[1*†], Christophe Guyeux[2†] and Mengjie Wang[1†]

[1] Guangdong University of Technology, College of Automation, Guangzhou, 510006, Guangdong, China.
[2] Université de Bourgogne Franche-Comté, FEMTO-ST Institute, BELFORT Cedex, 90016, Franche-Comté Region, France.

*Corresponding author(s). E-mail(s): qianxue_wang@163.com; Contributing authors: chen0717xin@163.com; siminyu@163.com; christophe.guyeux@univ-fcomte.fr; 2112004206@mail2.gdut.edu.cn;
†These authors contributed equally to this work.

Abstract

An Image Encryption Algorithm with the Quantum logistic and lorenz Chaotic map, and DNA Coding (IEA-QCDC) was proposed. Relying on some empirical analyzes and experimental results, the designers of IEA-QCDC claimed that this strategy can led to a significant enhancement in reliability and security. However, we investigate the essential properties of IEA-QCDC, and then propose an efficient chosen-plaintext attack to crack its equivalent permutation and diffusion key. By analyzing the encryption effect of continuous DNA encryption operations, the mathematical properties of some DNA codes were summarized. In addition, we find out that the iterative sequence obtained by the non-chaotic state is not even resistant to only-ciphertext attacks by analyzing the Lyapunov exponent and bifurcation diagrams of the quantum Logistic map. Aiming at the security vulnerabilities existing in IEA-QCDC, suggestions are put forward to improve the security and practicability of encryption algorithms, which will promote the development of cryptographic design to a certain extent.

Keywords: Quantum Logistic map, DNA coding, chosen-plaintext attack, multimedia privacy.

1 Introduction

With the development of chaotic image encryption research, hybrid chaotic image encryption algorithm has become a research hotspot [1–4], more and more hybrid image encryption schemes combined with DNA technology have been proposed [5–9]. For example, a secure and efficient image encryption scheme was proposed based on adaptive permutation-diffusion nuclear DNA random coding [10]. [11] proposed an image encryption scheme in which image pixels were diffused by DNA method and replaced by 2D-HSM. Relying on a Simultaneous intra-inter-Component Permutation Mechanism Dependent on the Plaintext (SCPMDP) on the plaintext, a color image encryption system was proposed based on dynamic DNA and chaos in [12]. Aiming at converting the ordinary images into three DNA matrices by random DNA coding rules, a color image encryption scheme was proposed based on DNA operations and spatiotemporal chaotic systems [13]. Cryptographic designers often claim that their encryption algorithms are secure based on the results of statistical tests. However, according to rigorous security analysis, these data alone cannot prove that the security performance of the encryption algorithm is sufficient [14–16]. In order to improve the security of encryption algorithms, it is necessary to analyze the security of existing encryption algorithms from the perspective of cryptanalysis [17–19].

Compared with cryptographic design, the work of cryptanalysts can expose the vulnerabilities of encryption algorithms, and improve the structure and security of new encryption algorithms [20–24]. Some DNA-based image encryption schemes are not as secure as claimed due to design flaws in the encryption algorithms [25–29]. Akhavan et al. studied the security of the DNA-based image encryption algorithm, and found that the security of the algorithm mainly depends on the static shuffling step, plus a simple permutation operation, and successfully restored the plaintext image by chosen-plaintext attack [25]. Su et al. discovered the role of entropy failure protection permutation index of chaotic image encryption scheme based on DNA coding and information entropy and the cover matrice whose elements in the last column are the same number, through chosen-plaintext attack [26]. In [27], Dou et al. found that the security of the image encryption algorithm based on DNA technology depends on the initial conditions of the one-dimensional and two-dimensional logical chaotic maps, including the sum of the front and rear pixels of the pure image, and based on this, the algorithm was cracked. In [28], Liu et al. found that the improved image encryption algorithm based on the first-order time-delay system did not use efficient nonlinear operations to ensure the sensitivity of the key stream, and on this basis, a specific chosen-plaintext attack was used to prove its effectiveness. Today, more and more DNA encryption algorithms have been proposed and gradually become known to the wider public [29, 30].

Cryptanalysis is used to breach cryptographic security systems and gain access to the contents of encrypted messages, even if the cryptographic key is unknown [31–33]. Cryptanalysis points out the security, practicability and feasibility problems in the encryption scheme, which has certain reference significance for the design of the encryption scheme [34, 35]. In above works, the multi-step encryption is often cracked step by step in order [25–28]. Our cracking algorithm regards the continuous DNA encryption steps as a whole, and uses only one equivalent diffusion matrix to crack the entire continuous encryption, and the crack efficiency is higher. In this paper, we studied the IEA-QCDC algorithm [1], summarize the mathematical properties of the multi-step continuous encryption related to DNA, found that the multi-step continuous encryption can be simplified to a one-time encryption effect without changing the DNA correlation rules, and used an equivalent diffusion matrix to complete the interpretation of this link. Through our analysis, readers can crack the original encryption algorithm even if the initial key is unknown. In addition, we analyzed the Lyapunov exponent and bifurcation diagram of the quantum Logistic map, and pointed out that the system had a non-chaotic state problem within the parameter range in the original paper, which caused the algorithm to be cracked by only-ciphertext attacks when the system was in a non-chaotic state, and put forward some suggestions to improve its security and practicability.

The remainder of this paper is organized as follows. The encryption process of the IEA-QCDC algorithm is briefly introduced in Sec. 2. Sec. 3 conducts a cryptanalysis of the method, and proposes the chosen-plaintext attack method for the main security issues, algorithm design issues and core attack principles of the algorithm. The simulation tests on the feasibility of the proposed the chosen-plaintext attack method is given in Sec. 4. Sec. 5 presents some suggestions to improve the security and practicality of the original method. The last section concludes the paper.

2 A precise and concise description of IEA-QCDC

This section briefly introduces the original encryption algorithm, and the details of the encryption scheme can be found in [1]. During the introduction of the encryption scheme, the original symbols are used wherever possible. The algorithm structure of the original text is shown in Fig. 1.

2.1 The quantum Logistic map and the Lorenz map

In this section, we will introduce the quantum Logistic map and the Lorenz map.

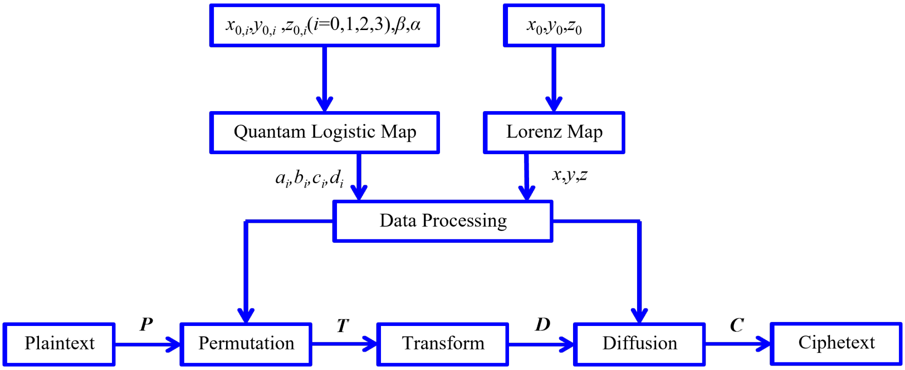


Fig. 1 The overall architecture of IEA-QCDC

The quantum Logistic map used in [1] is described by the following equations:

$$\begin{cases} x_{n+1} = r\left(x_n - \mid x_n \mid^2\right) - r y_n, & (1a) \\ y_{n+1} = -y_n e^{-2\beta} + 2e^{-\beta} r\left[(1 - x_n) y_n - x_n z_n\right], & (1b) \\ z_{n+1} = -z_n e^{-2\beta} + e^{-\beta} r\left[2(1 - x_n) z_n - 2x_n y_n - x_n\right], & (1c) \end{cases} \tag{1}$$

where $n = 1, 2, 3, \cdots$. The state variables are $x_n \in (0, 2)$, $y_n \in (0, 0.1)$ and $z_n \in (0, 0.2)$, while the dissipation parameter $\beta \in [6, \infty)$ and the control parameter $r \in (0, 4)$.

Secondly, the Lorenz map is defined as follows:

$$\begin{cases} \frac{dx}{dt} = 10\,(y - x)\,, \\ \frac{dy}{dt} = 28x - xz - y, \\ \frac{dz}{dt} = xy - \frac{8}{3}z, \end{cases} \tag{2}$$

where the chaotic sequences can be generated by using Runge-Kutta method.

2.2 DNA coding

The DNA sequence is composed of four different basic deoxynucleotides, namely A (adenine), T (thymine), C (cytosine), and G (guanine), and their complementarity is only allowed between A and T as well as G and C. In addition, each pixel in the image can be represented by an 8-bit binary (4 pairs of 2-bit binary). These 2-bit binary pairs also comprise complementary pairs, i.e. 00 and 11 as well as 01 and 10 are complementary. The 2-bit binary pairs 00, 01, 10, and 11 can be encoded by the DNA bases A, T, C, and G. Because the coding principle must be satisfied: A and T as well as G and C are complementary, there are eight combinations according to different assignments as shown in Table 1. For example, “11100100” can be expressed as “TCGA” by the first rule chosen in Table 1. Each DNA rule has corresponding DNA addition, DNA subtraction, and DNA XOR rules. The essence of these DNA operations is to decode the DNA symbols into 2-bit binary according to the corresponding rules, and then encode the results after addition, subtraction, and XOR operations according to the corresponding rules.

Table 1: Eight DNA encoding and decoding rules

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

2.3 The IEA-QCDC algorithm description

According to [1], the specific flow chart of the IEA-QCDC algorithm is shown in Fig. 2, which is explained as follows.

(1) Select key parameters. It can be seen from Fig. 1 that the IEA-QCDC algorithm has 17 key parameters, namely: $x_{0,i}, y_{0,i}, z_{0,i}$ $(i = 1, 2, 3, 4)$, β, r, x_0, y_0, z_0, where $x_{0,i}, y_{0,i}, z_{0,i}$ $(i = 1, 2, 3, 4)$ are four sets of initial values of the quantum Logistic map, β and r are the control parameters of the quantum Logistic map, x_0, y_0 and z_0 are the initial values of the Lorenz map.

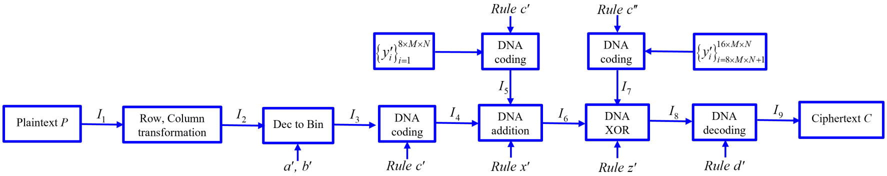


Fig. 2 The flow chart of the IEA-QCDC algorithm

(2) Generate chaotic sequences. The quantum Logistic map is iterated 1000 times with the control parameters β, r and 4 different sets of initial values $x_{0,i}, y_{0,i}, z_{0,i}$ $(i = 1, 2, 3, 4)$ to prevent transient effects on initial iterations. Set $L = M \times N$, iterate $2M$, $2N$, $4L$, and $4L$ times respectively, and only take the result of Eq. 1 (1a) to obtain 4 groups of chaotic sequences:

$$\begin{cases} a = \{x_{i,1}\}_{i=1001}^{1000+2M}, \\ b = \{x_{i,2}\}_{i=1001}^{1000+2M}, \\ c = \{x_{i,3}\}_{i=1001}^{1000+4L}, \\ d = \{x_{i,4}\}_{i=1001}^{1000+4L}, \end{cases} \tag{3}$$

where a consists of the 1001st to the $1000+2M$th element of sequence $x_{i,1}$, b consists of the 1001st to the $1000+2M$th element of sequence $x_{i,2}$, c consists of the 1001st to the $1000+4L$th element of sequence $x_{i,3}$, d consists of the 1001st to the $1000+4L$th element of sequence $x_{i,4}$.

They can be quantified as $a' = \{a'_i\}_{i=1}^{2M}$, $b' = \{b'_i\}_{i=1}^{2N}$, $c' = \{c'_i\}_{i=1}^{4L}$, and $d' = \{d'_i\}_{i=1}^{4L}$ via

$$\begin{cases} a'_i = \mathrm{mod}\left(fix\left(a_i \times 10^6\right), N\right) + 1, \\ b'_i = \mathrm{mod}\left(fix\left(b_i \times 10^6\right), M\right) + 1, \\ c'_i = \mathrm{mod}\left(fix\left(c_i \times 10^6\right), 8\right) + 1, \\ d'_i = \mathrm{mod}\left(fix\left(d_i \times 10^6\right), 8\right) + 1, \end{cases} \tag{4}$$

where $i = 1, 2, \cdots, 2M$, $\mathrm{mod}(\cdot)$ is the operation of rounding to the nearest integer and $fix(\cdot)$ is the remainder operation. Split a' and b' in half as $\{a'_i\}_{i=1}^{M}$ and $\{a'_i\}_{i=M+1}^{2M}$ as well as $\{b'_i\}_{i=1}^{N}$ and $\{b'_i\}_{i=N+1}^{2N}$.

Iterate the Lorenz map by Eq. 2 with the initial values x_0, y_0, and z_0, then select the first $4L$ iterations of x_i, the first $16L$ iterations of y_i and the first $4L$ iterations of z_i, respectively, to form $x = \{x_i\}_{i=1}^{4L}$, $y = \{x_i\}_{i=1}^{16L}$ and $z = \{z_i\}_{i=1}^{4L}$. The random sequences $x' = \{x'_i\}_{i=1}^{4L}$, $y' = \{y'_i\}_{i=1}^{16L}$, and $z' = \{z'_i\}_{i=1}^{4L}$ are obtained by

$$\begin{cases} x'_i = \mathrm{mod}\left\{fix\left\{\left[abs\left(x_i\right) - fix\left(abs\left(x_i\right)\right)\right] \times 10^{10}\right\}, 8\right\} + 1, \\ y'_i = \mathrm{mod}\left\{fix\left\{\left[abs\left(y_i\right) - fix\left(abs\left(y_i\right)\right)\right] \times 10^{10}\right\}, 2\right\} + 1, \\ z'_i = \mathrm{mod}\left\{fix\left\{\left[abs\left(z_i\right) - fix\left(abs\left(z_i\right)\right)\right] \times 10^{10}\right\}, 8\right\} + 1, \end{cases} \tag{5}$$

where $abs(\cdot)$ is used to obtain the absolute value, $fix(\cdot)$ is used to obtain the integer outcome, $y' = \{y'_i\}_{i=1}^{16L}$ is split into the first half $y' = \{y'_i\}_{i=1}^{8L}$ and the latter half $y' = \{y'_i\}_{i=8L+1}^{16L}$.

(3) Convert the grayscale image into a 2-dimensional matrix $I_1 = P = \{p_{i,j}\}_{i=1,j=1}^{M,N}$, where $p_{i,j}$ indicates the pixel value of the i-th row and the j-th column in the input image. According to the Eq. 6, the row transformation is performed on I_1 according to $\{a'_i\}_{i=1}^{M}$, and R' is the image matrix after the first round of row transformation of P. By using the M numbers in the sequence $\{a'_i\}_{i=1}^{M}$, I_1 is transformed to R' by row permutation via

$$R'_{i,j} = P_{i,\, \mathrm{mod}\,(j+a'_i, N)+1}, \tag{6}$$

where $i = 1, 2, \cdots, M$, $j = 1, 2, \cdots, N$.

Then, the column transformation is performed on the row-transformed image R' according to $\{b'_i\}_{i=1}^{N}$, via

$$A'_{i,j} = R'_{\mathrm{mod}\,(i+b'_i, M)+1, j}, \tag{7}$$

where $i = 1, 2, \cdots, M$, $j = 1, 2, \cdots, N$, and A' is the image matrix after the first round of column transformation.

After the above steps, repeat the row and column transformation operations according to $\{{a'}_i\}_{i=1+M}^{2M}$ and $\{{b'}_i\}_{i=1+N}^{2N}$ as

$$R''_{i,j} = A'_{i,\ \bmod\ (j+{a'}_{M+i},N)+1} \tag{8}$$

and

$$A''_{i,j} = {R''}_{\bmod\ (i+b'_{N+i},M)+1,j} \tag{9}$$

where $i = 1, 2, \cdots, M$; $j = 1, 2, \cdots, N$, and the replaced image is denoted as $I_2 = A''_{i,j}$.

(4) Each element in I_2 can be expressed as an 8-bit binary number to get I_3, each pair of 2-bit binary in I_3 by scanning it in the raster order (from left to right, and top to bottom) is encoded according to the corresponding DNA coding rule in c', to obtain the symbol matrix I_4 composed of "ATCG"; the sequence $y' = \{{y'}_i\}_{i=1}^{8L}$ forms $M \times N$ matrix in raster scan order through 8 binary digits as an element, and each pair of 2-bit binary digit is encoded according to the corresponding DNA coding rule in c', to obtain the symbol matrix I_5 composed of "ATCG".

(5) According to the values in the sequence x', the DNA addition rule is dynamically selected to realize the DNA addition operation of the matrix I_4 and the matrix I_5, then the addition result I_6 is obtained. Eight rules correspond to eight DNA addition operations. Table 2 only shows the DNA additions under rule 1.

(6) The binary sequence $y' = \{{y'}_i\}_{i=8L+1}^{16L}$ forms $M \times N$ matrix in raster scan order through 8 binary digits as an element, and each pair of 2-bit binary digit is encoded according to the corresponding DNA coding rule in $c' = \{{c'}_i\}_{i=1}^{4L}$, to obtain a symbol matrix I_5 consisting of "ATCG". Then, the DNA XOR rule is dynamically selected to realize the DNA addition operation of I_6 and I_7, to obtain I_8 according to the values in the sequence $z' = \{{z'}_i\}_{i=1}^{4L}$. Table 3 gives the DNA XOR operation under rule 1.

Table 2: The addition operation of DNA sequence based on rule 1

+	A	C	G	T
A	A	C	G	T
C	C	G	T	A
G	G	T	A	C
T	T	A	C	G

Table 3: The XOR operation of DNA sequence based on rule 1

XOR	A	C	G	T
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A

(7) Dynamically select DNA decoding rules according to the values in sequence d' to obtain binary matrix I_9 from I_8, then convert binary to decimal to obtain the final encryption matrix C from I_9.

The decryption process is the reverse process of the IEA-QCDC algorithm encryption.

3 Cryptanalysis of the IEA-QCDC algorithm

In this section, we point out some feasibility, practicality, and security issues found in the IEA-QCDC algorithm. We conduct an overall analysis of the IEA-QCDC algorithm, and propose a specific chosen-plaintext attack based on cryptanalysis.

3.1 The overall analysis of the IEA-QCDC algorithm

In this section, we will conduct an overall analysis of the IEA-QCDC encryption algorithm.

Issue 1: The keystream is independent of the plaintext image and completely depends on the initial parameters of the key.

Specifically, the keystream used in the encryption process of the IEA-QCDC algorithm is obtained from the chaotic sequence. However, the initial parameters and control parameters of the quantum Logistic map and the Lorenz map are fixed. The permutation in the IEA-QCDC algorithm is an invalid encryption from the perspective of cryptanalysis. In the permutation, DNA encoding, DNA addition, DNA XOR, and DNA decoding operations have no direct or indirect relationship with plaintext. That is to say, when the initial parameters and control parameters are fixed, the chaotic sequence involved in the IEA-QCDC encryption algorithm is also fixed.

Issue 2: The use of chaotic sequences is unreasonable.

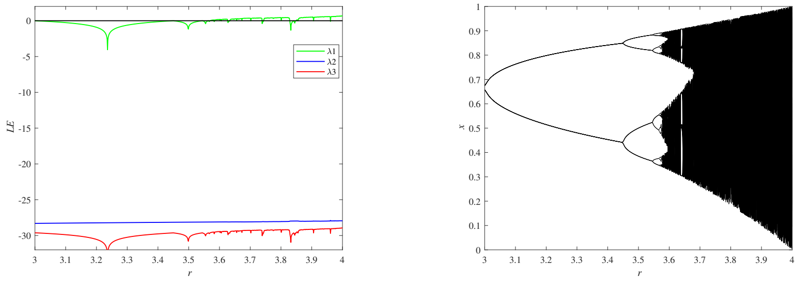
First, in this encryption method, the quantum Logistic system can generate three chaotic sequences during the iterative process, but only the chaotic sequence x is used for encryption. The Lorenz sequence needs to be iterated $16L$ times, but sequences x and z only use the first $4L$ numbers. Secondly, the effective precision of the seven chaotic sequences used in the original text is at least 15 decimal places, and they are quantized into integer sequences with a small range of values. For example, each floating point number with an effective precision of 15 decimal digits in the Lorenz sequence y is only used to generate a single binary number, so the usage rate of the chaotic sequence y is extremely low. Finally, in order to encrypt an image of size $M \times N$, the quantum Logistic map needs to iterate $2 \times (M + N) + 8L$ times with four sets of different initial values, and the Lorenz map needs to iterate $16L$ times, which make the encryption slower.

Issue 3: DNA encoding, DNA addition, DNA XOR, and DNA decoding can be simplified equivalently.

The diffusion process of this encryption method is too complicated, the amount of calculation is too large, and the encryption efficiency is relatively slow. It can be seen from Sec. 2.3 and Fig. 2 that the IEA-QCDC algorithm has four different diffusions for the input pixels, namely DNA coding, DNA addition, DNA XOR, and decoding. However, four diffusions have the same encryption effect as doing one global diffusion. The diffusion process of this encryption method only needs 256 chosen-plaintext images to completely determine the equivalent diffusion key. Theoretically, it takes up to $256 \times M \times N$ searches to eliminate the encryption effect of diffusion.

Issue 4: The selection of some control parameters for the quantum Logistic map in the IEA-QCDC algorithm will lead to non-chaotic phenomenon.

In the description of the quantum Logistic map of the IEA-QCDC algorithm, it is claimed that the dissipation parameter $\beta \in [6, \infty)$ and the control parameter $r \in [3, 4]$. However, the actual test found that there is non-chaotic phenomenon in this range. As shown in Fig. 3a, when taking $\beta = 30$, $r \in [3, 4]$, a maximum Lyapunov exponent smaller than 0 is found, which leads to a quantum Logistic map in a non-chaotic state. The corresponding bifurcation diagram is shown in Fig. 3b, it can be seen that the quantum Logistic map has no wider chaotic interval than the original Logistic map. In this case, there are partial non-chaotic states, and chaotic sequences cannot be guaranteed during iteration. A similar situation also occurs in the interval $\beta = 30$, and $r \in [2, 3]$, as shown in Fig. 4, and its corresponding bifurcation diagram is shown in Fig. 4b. This means that the system is in a non-chaotic state in this case, and the iteration result is not a chaotic sequence. When the selected parameters make the system in a non-chaotic state, the chaotic sequence used for encryption can be easily cracked by statistical attacks and only-ciphertext attacks, and such encryption has no security at all.



(a)

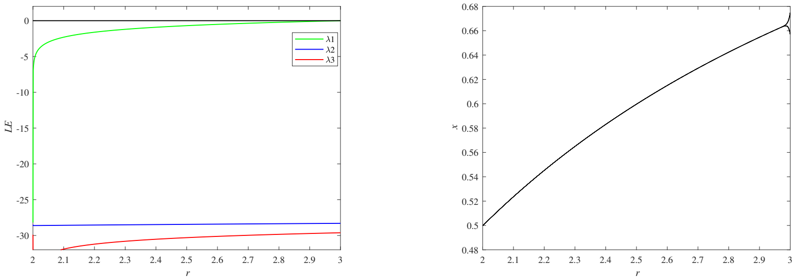
(b)

Fig. 3 Dynamical analysis of quantum Logistic system for $\beta = 30$, $r \in [3, 4]$: (a) Lyapunov exponent spectrum, (b) The bifurcation diagram

3.2 Cryptanalysis and attack methods

3.2.1 Analysis of diffusion process of the IEA-QCDC algorithm

In this section, we analyze the diffusion process of the IEA-QCDC encryption algorithm.



(a)

(b)

Fig. 4 Dynamical analysis of quantum Logistic system for $\beta = 30$, $r \in [2, 3]$: (a) Lyapunov exponent spectrum, (b) The bifurcation diagram

It can be seen from the analysis in the previous section that the IEA-QCDC algorithm has some defects, and the IEA-QCDC algorithm can be simplified to an encryption algorithm of pixel-level permutation and bit-level diffusion without preconditions. When the key parameters are given, for different input plaintexts, the chaotic sequences a, b, c, d, x, y, and z involved in permutation, DNA encoding, DNA addition, DNA XOR, and DNA decoding remain unchanged, indicating that the IEA-QCDC algorithm has an equivalent permutation key S_{eq} and an equivalent diffusion key E_{eq}, which can be cracked separately by the chosen-plaintext attack.

Let $f_p(x)$ denote the implementation of the DNA encoding defined for x in the p-th column of Table 1, thus, $f_q^{-1}(x)$ denotes the use of the rule q for x to decode DNA, $h_{\underset{s}{\oplus z}}(x)$ means to perform DNA addition operation on x and y according to rule r, $g_{\underset{r}{+y}}(x)$ means to perform DNA addition operation on x and y according to r, $h_{\underset{s}{\oplus z}}(x)$ means to perform DNA XOR operation on x and z according to the rule s, where p, q, r, s represent the rule number and $p, q, r, s \in \{0, 1, 2, 3, 4, 5, 6, 7\}$, x, y, z represent two binary digits and $x, y, z \in \{00, 01, 10, 11\}$. In order to study the intrinsic properties of DNA encoding computations, a composite function is defined as

$$F_{p,\underset{r}{+y},\underset{s}{\oplus z},q} = f_q^{-1} \circ h_{\underset{s}{\oplus z}} \circ g_{\underset{r}{+y}} \circ f_p \tag{10}$$

Eq. 10 can be expressed as the diffusion process of the encryption algorithm.

Property 1 ([33]) If f is a bijective function, then the inverse function f^{-1} of f is also a bijective function.

Proposition 1 *When p and q are determined, DNA encoding f_p and DNA decoding f_q^{-1} are bijective functions.*

Proof When p in $f_p : \{00, 01, 10, 11\} \to \{A, T, C, G\}$ is determined, the rules of Table 1 are determined.

1. For every $y \in \{A, T, C, G\}$, there is an $x \in \{00, 01, 10, 11\}$ such that $f_p(x) = y$, so f_p is a surjective.
2. Evidently, $a \neq b$ implies $f_p(a) \neq f_p(b)$, where $a, b \in \{00, 01, 10, 11\}$, so f_p is injective.

To sum up, f_p is a bijective function, and f_q^{-1} is also a bijective function when q is determined by the Property 1.

□

Proposition 2 *When y, z, r and s are determined, DNA addition function $g_{\underset{r}{+y}}$ and DNA XOR function $h_{\underset{s}{\oplus z}}$ are bijective functions.*

Proof It can be proved by the DNA addition Table 2 and DNA XOR Table 3. □

Property 2 ([33]) If f and g are bijective functions, then the composite function $f \circ g$ is also a bijective function.

In the diffusion process of the IEA-QCDC algorithm, set the k-th pair of 2-bit binary in the matrix I_3 by scanning it in the raster order as A_k. It can be seen from the previous definition that the DNA encoding is $f_{{c'}_k}(A_k)$, and then perform DNA addition operation on $f_{{c'}_k}(A_k)$ and Y'_k (k-th pair of 2-bit binary in I_5) according to the rule x'_k to get $g_{\underset{{x'}_k}{+{Y'}_k}}(f_{{c'}_k}(A_k))$, and perform DNA XOR on $g_{\underset{{x'}_k}{+{Y'}_k}}(f_{{c'}_k}(A_k))$ and Y''_k (k-th pair of 2-bit binary in I_7) according to the rule z'_k to get $h_{\underset{{z'}_k}{\oplus {Y''}_k}}(g_{\underset{{x'}_k}{+{Y'}_k}}(f_{{c'}_k}(A_k)))$, and finally achieve DNA decoding according to the rule d'_k as

$$F_{o,{d'}_k} = f^{-1}_{{d'}_k}(e) = f^{-1}_{{d'}_k} \circ h_{\underset{{z'}_k}{\oplus {Y''}_k}} \circ g_{\underset{{x'}_k}{+{Y'}_k}} \circ f_{{c'}_k} \tag{11}$$

where $k = 1, 2, \ldots, 4L$ and $e = (h_{\underset{{z'}_k}{\oplus {Y''}_k}}(g_{\underset{{x'}_k}{+{Y'}_k}}(f_{{c'}_k}(A_k))))$. Eq. 11 can be expressed as the diffusion process of the dencryption algorithm.

Proposition 3 *When k is determined, the equivalent function $F_{o,{d'}_k}$ for the diffusion of the IEA-QCDC algorithm is a bijective function.*

Proof Because in the IEA-QCDC algorithm, the generated chaotic sequences a, b, c, d, x, y, and z are all unchanged, which leads to DNA encoding rules, DNA addition rules, DNA XOR rules, DNA decoding rules, and I_5 and I_7 used in encryption are also determined. So when k is determined, the values of c'_k, Y'_k, x'_k, Y''_k, z'_k, and d'_k

are also determined. From Propositions 1 and 2, we know that $f^{-1}_{d'_k}, h_{\underset{z'_k}{\oplus}Y''_k}, g_{\underset{x'_k}{+}Y'_k}$ and $f_{c'_k}$ are both bijective functions, then according to Property 2, F_{o,d'_k} is proved to be a bijective function. □

It can be seen from Proposition 3 that as long as we obtain the output values of all possible values corresponding to each position of the matrix I_2 in the matrix C, the original text corresponding to any diffusion ciphertext can be cracked.

In order to eliminate the interference of the permutation step on the cracking process, it can be considered to choose a plaintext or ciphertext image with a single pixel value for cracking. Since there are 256 cases of the size of each pixel, we select 256 images with a single pixel value for cracking. The chosen-plaintext attack method is used to crack the diffusion process in the IEA-QCDC algorithm. The specific process is as follows.

The first step is to select a set $\{P_0, P_1, \ldots, P_{255}\}$ of 256 single-pixel images of size $M \times N$, so that all pixel values in matrix $P_k (k \in \{0, 1, \ldots, 255\})$ is equal to k, those are $P_0 = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$, $P_1 = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$,..., $P_{255} = \begin{pmatrix} 255 & 255 & \cdots & 255 \\ 255 & 255 & \cdots & 255 \\ \vdots & \vdots & & \vdots \\ 255 & 255 & \cdots & 255 \end{pmatrix}$, to ensure that their permuted pixel values remain unchanged. Therefore, the attacker only needs to focus on the intermediate states from the permutation image to the ciphertext image.

The second step is to use the encryption machine to get the corresponding ciphertext image C_k $(k = 0, 1, 2, \ldots, 255)$.

The third step is to scan C_k $(k = 0, 1, 2, \ldots, 255)$ in raster order and write the row matrix C^*_k $(k = 0, 1, 2, \ldots, 255)$.

The fourth step is to "placed" the 256 row matrices C^*_k $(k = 0, 1, 2, \ldots, 255)$ vertically in ascending order of k to obtain the equivalent diffusion key E_{eq}.

The fifth step is to write C as a one-dimensional sequence $C^* = \{c_i{}^*\}_{i=1}^{M \times N}$ for a ciphertext C of a given size $M \times N$. From the i-th column of the equivalent diffusion key E_{eq}, traversing the c_i value can obtain the corresponding pixel value in the matrix I_2 before diffusion. The corresponding pixel value in I_2 before diffusion is equal to the row coordinate of the position where c_i is traversed in the i-th column of the equivalent diffusion key E_{eq}, minus 1. Theoretically, it needs to traverse $M \times N \times 256$ times at most to restore all pixel values of the image before diffusion.

A simple example is used to illustrate this process. Assuming that the image size is 3×3, and each pixel is represented by a 2-bit binary, the permutation matrix is $I_2 = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 1 & 1 \\ 3 & 0 & 3 \end{pmatrix}$. The DNA rule sequences are $c' = \{1,3,5,3,2,8,4,6,7\}$, $d' = \{7,2,4,3,1,8,6,7,5\}$, $x'_k = \{5,4,6,4,7,3,5,5,2\}$ and $z'_k = \{7,4,7,7,7,1,5,7,8\}$, then $I_5{}' = \begin{pmatrix} A & G & C \\ T & A & G \\ C & T & A \end{pmatrix}$ and $I_7{}' = \begin{pmatrix} G & T & A \\ C & G & T \\ T & A & C \end{pmatrix}$.

The diffusion operation process of I_2 is as follows.

$$I_2 = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 1 & 1 \\ 3 & 0 & 3 \end{pmatrix} \xrightarrow[\text{conversion}]{\text{Bit}} I_3 = \begin{pmatrix} 00 & 01 & 10 \\ 10 & 01 & 01 \\ 11 & 00 & 11 \end{pmatrix} \xrightarrow{f_{c'}} I_4 = \begin{pmatrix} A & A & T \\ T & A & C \\ G & G & A \end{pmatrix}$$

$$\xrightarrow{g_{x'+y'}} I_6 = \begin{pmatrix} A & C & A \\ C & G & C \\ G & A & T \end{pmatrix} \xrightarrow{h_{z'\oplus y''}} I_8 = \begin{pmatrix} G & T & T \\ A & C & C \\ T & C & A \end{pmatrix}$$

$$\xrightarrow{f_{d'}^{-1}} I_9 = \begin{pmatrix} 01 & 00 & 10 \\ 01 & 00 & 10 \\ 11 & 01 & 01 \end{pmatrix} \xrightarrow[\text{conversion}]{\text{Bit}} C = \begin{pmatrix} 1 & 0 & 3 \\ 1 & 0 & 2 \\ 3 & 1 & 1 \end{pmatrix}$$

Now consider the given $C = \begin{pmatrix} 1 & 0 & 3 \\ 1 & 0 & 2 \\ 3 & 1 & 1 \end{pmatrix}$, the process of cracking the diffusion to get I_2 is as follows.

Select 4 single-pixel images and put them into the encryption machine to get the corresponding ciphertext.

$$P_0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \xrightarrow{\text{IEA-QCDC}} C_0 = \begin{pmatrix} 1 & 2 & 2 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$P_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \xrightarrow{\text{IEA-QCDC}} C_3 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 2 \\ 2 & 2 & 2 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{pmatrix} \xrightarrow{\text{IEA-QCDC}} C_2 = \begin{pmatrix} 0 & 1 & 3 \\ 1 & 1 & 3 \\ 0 & 3 & 0 \end{pmatrix}$$

$$P_3 = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \end{pmatrix} \xrightarrow{\text{IEA-QCDC}} C_3 = \begin{pmatrix} 3 & 3 & 1 \\ 3 & 3 & 0 \\ 3 & 0 & 1 \end{pmatrix}$$

Write it as a row matrix by scanning it in raster order, and "place" vertically to obtain the equivalent diffusion key.

$$E_{eq} = \begin{pmatrix} [1] & 2 & 2 & 2 & 2 & 1 & 1 & [1] & 1 \\ 2 & [0] & 0 & 0 & [0] & [2] & 2 & 2 & 2 \\ 0 & 1 & [3] & [1] & 1 & 3 & 0 & 3 & 0 \\ 3 & 3 & 1 & 3 & 3 & 0 & [3] & 0 & [1] \end{pmatrix} \tag{12}$$

The framed part $[\cdot]$ is the pixel value in C obtained in raster scan order. After subtracting 1 from the abscissa of the equivalent diffusion key E_{eq} where the framed part is located, we get $0,1,2,2,1,1,3,0,3$.

It is verified that the cracked matrix $\begin{pmatrix} 0 & 1 & 2 \\ 2 & 1 & 1 \\ 3 & 0 & 3 \end{pmatrix}$ is indeed I_2.

3.2.2 Analysis of permutation process of the IEA-QCDC algorithm

In this section, we analyze the permutation process of the IEA-QCDC encryption algorithm.

According to the interpretation of the permutation process of the IEA-QCDC algorithm, it can be seen from Eq. 6 and Eq. 7 that two rounds of row and column transformation are performed on each element of the plaintext image of size $M \times N$ according to the chaotic sequences. In fact, the encryption effect of the permutation process is equivalent to the pixel permutation process of the input plaintext image, and the equivalent permutation key S_{eq} can be obtained by cracking a special plaintext.

Proposition 4 *The final effect after two rounds of permutation is equivalent to a pixel permutation process.*

Proof Let's suppose the ciphertext corresponding to the plaintext image P after two rounds of permutation of the IEA-QCDC algorithm is A'', according to the encryption order of "first row transformation → first column transformation → second row transformation → second column transformation", Eq. 6, Eq. 7, Eq. 8 and Eq. 9, one can get

$$\begin{aligned} A''_{i,j} &= {R''}_{\bmod\left(i+{b'}_{N+i},M\right)+1,j} \\ &= {A'}_{\bmod\left(i+{b'}_{N+i},M\right)+1,\ \bmod\left(j+{a'}_{M+i},N\right)+1} \\ &= {R'}_{\bmod((\bmod(i+{b'}_i),M)+1+{b'}_{N+i},M)+1,\ \bmod(j+{a'}_{M+i},N)+1} \\ &= p_{\bmod((\bmod(i+{b'}_i),M)+1+{b'}_{N+i},M)+1,\ \bmod(\bmod({a'}_i+j,N)+1+{a'}_{M+i},N)+1} \\ &= p_{k,l}. \end{aligned} \tag{13}$$

where $i = 1, 2, \cdots, M$ and $j = 1, 2, \cdots, N$. It can be found that the positional relationship between the pixels of the permutation matrix A'' and the plaintext matrix P:

$$\begin{aligned} k &= \bmod(\bmod({b'}_i + i, M) + 1 + {b'}_{N+i}, M) + 1 \\ l &= \bmod((\bmod({a'}_i + j, N) + 1 + {a'}_{M+i}, N) + 1 \end{aligned} \tag{14}$$

It can be seen from Eq. 14 that the two rounds of permutation operations are equivalent to one transformation, because the different plaintext encryption sequences a', b' are fixed, so the above process has an equivalent permutation encryption key. □

Therefore, for a grayscale image of size $M \times N$, the cracking steps based on the chosen-plaintext attack method are as follows.

The first step is to construct a virtual matrix V, which is the same size as the image $P = \{p_{i,j}\}_{i=1,j=1}^{M,N}$, v_k is the value of the k-th element of V in raster scan order. Store $0, 1, \cdots, M \times N - 1$, that is, $v_k = k - 1 (k = 1, 2, 3, \cdots, M \times N)$ in the virtual matrix V in sequence in raster scan order.

The second step is to create an virtual matrix V of the same size as the image $P = \{p_{i,j}\}_{i=1,j=1}^{M,N}$, v_k is the value of the k-th element of V in raster scan order. Store $0, 1, \cdots, M \times N - 1$ in V in the raster scan order, that is, $v_k = k - 1 (k = 1, 2, 3, \cdots, M \times N)$.

The third step is to use V to write data to the values of t plaintext images $P_1, P_2, \cdots, P_t$. The writing rule for the j-th element $P_{i,j}$ obtained from the raster scan order of the i-th plaintext image is as follows:

$$P_{i,j} = \left\lfloor (v_k / 256^{i-1}) \% 256 \right\rfloor \tag{15}$$

where $i = 1, 2, 3, \cdots, t$ and $j = 1, 2, 3, \cdots, M \times N$.

After constructing the above plaintext, the t plaintext images $P_1, P_2, \cdots, P_t$ are sequentially encrypted by the encryption machine, and the corresponding ciphertext images C_1,$C_2, \cdots, C_t$, and then inversely diffuse through the key E_{eq} to counteract the diffusion effect, and obtain t intermediate ciphertext images, which are $A''_1, A''_2, \cdots, A''_t$. Combine them to obtain the unique permutation matrix W. The merge rule is

$$W = \sum_{i=1}^{t} \left(A''_i \times 256^{i-1} \right) \tag{16}$$

where $i = 1, 2, 3, \cdots, t$. Comparing the position changes of the virtual matrix V and the permutation matrix W with the same pixel value, the equivalent permutation key S_{eq} can be obtained.

Therefore, using the chosen-plaintext attack method requires only $256 + \lceil \log_{256} (M \times N) \rceil$ chosen-plaintext images and their corresponding ciphertext images to crack IEA-QCDC, so the data complexity of the chosen-plaintext attack method is $O(\log M \times N)$.

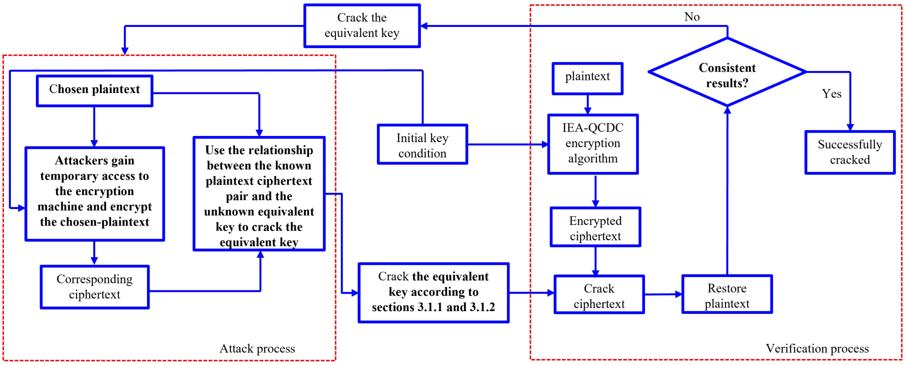

Crack the equivalent key
No
Consistent results?
Chosen plaintext
plaintext
Yes
Attackers gain temporary access to the encryption machine and encrypt the chosen-plaintext
Use the relationship between the known plaintext ciphertext pair and the unknown equivalent key to crack the equivalent key
Initial key condition
IEA-QCDC encryption algorithm
Successfully cracked
Encrypted ciphertext
Corresponding ciphertext
Crack the equivalent key according to sections 3.1.1 and 3.1.2
Crack ciphertext
Restore plaintext
Attack process
Verification process

Fig. 5 Specific attack process against the IEA-QCDC algorithm

The following is a simple example to illustrate the process of permutation. Suppose the row transformation sequence is $a_p = (2,2,2,3,3,1,2,2,4,1,1,2,3,4,2,3,4,4,1,3,4,1,1,2,1,1,2,4,4,1,1,2)$.

Assume that the column transformation sequence is $b_p = (1,1,1,1,2,2,3,1,2,2,4,1,3,3,4,1,1,4,2,4,1,2,3,2,2,4,1,4,4,2,2,2)$.

Suppose a virtual matrix of size 4 × 4 is $p_v = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$

According to the permutation process p_v of the IEA-QCDC algorithm, there will be the following encryption process.

The first round of permutation is as follows:

$$pv = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \xrightarrow[\text{Transformation}]{\text{Rowe}} pv_1 = \left(\begin{array}{|c||c||c||c|} 2 & 3 & 1 & 4 \\ 5 & 7 & 8 & 6 \\ 11 & 9 & 12 & 10 \\ 13 & 15 & 14 & 16 \end{array}\right)$$

$$\xrightarrow[\text{Transformation}]{\text{Column}} pv_2 = \left(\begin{array}{|c||c||c|c|} 5 & 7 & 14 & 10 \\ 11 & 9 & 8 & 16 \\ 2 & 3 & 1 & 4 \\ 13 & 15 & 12 & 6 \end{array}\right)$$

The second round of permutation is as follows:

$$pv_2 = \begin{pmatrix} 5 & 7 & 14 & 10 \\ 11 & 9 & 8 & 16 \\ 2 & 3 & 1 & 4 \\ 13 & 15 & 12 & 6 \end{pmatrix} \xrightarrow[\text{Transformation}]{\text{Row}} pv_3 = \left(\begin{array}{|c||c||c||c|} 14 & 5 & 7 & 10 \\ 8 & 11 & 16 & 9 \\ 4 & 1 & 2 & 3 \\ 12 & 13 & 6 & 15 \end{array}\right)$$

$$\xrightarrow[\text{Transformation}]{\text{Column}} cv = \left(\begin{array}{|c||c||c|c|} 12 & 11 & 2 & 9 \\ 4 & 5 & 16 & 10 \\ 14 & 13 & 6 & 15 \\ 8 & 1 & 7 & 6 \end{array}\right)$$

The equivalent permutation key S_{eq} can be obtained by comparing the position transformation of the same elements of p_v and c_v. As can be seen from the above, we only need $\lceil \log_{256}(M \times N) \rceil$ special plaintexts to obtain the equivalent permutation key S_{eq}.

According to the above analysis, without knowing the key parameters, the two-round permutation process of the IEA-QCDC algorithm can be cracked by obtaining its equivalent permutation key S_{eq}.

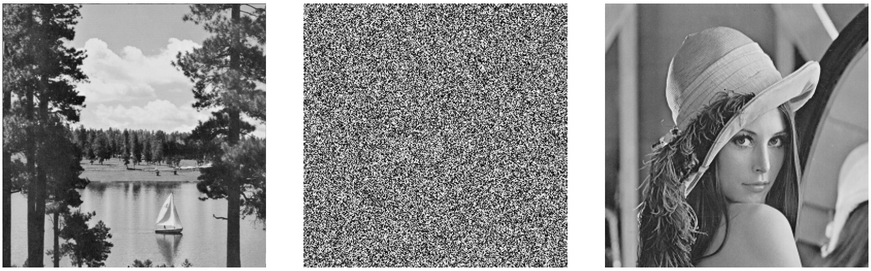
4 Cracking the IEA-QCDC simulations and experiments

The experimental hardware platform is PC. The PC configuration is as follows: the processor is AMD Ryzen 5 5600G, the main frequency is 3.90 GHz, the memory size is 16 G, the hard disks are a 128G solid-state hard disk and a 1 T mechanical hard disk. The software environment is Windows 10 operating system and Matlab R2019a. The images selected in the experiment are the same as the original [1], specifically "Lena" with a grayscale image of size 256×256 and "Lake" with a grayscale image of size 512×512.

Specific attack process against the IEA-QCDC algorithm as Fig. 5.

4.1 The experiment of cracking the IEA-QCDC by the chosen-plaintext attack method

According to the chosen-plaintext attack method proposed in Sec. 3, the IEA-QCDC is cracked. The size of the given image is 256×256. First, according to Sec. 3.2.1, select 256 special plaintexts as their permutation images, and input them into the encryption machine to obtain the corresponding encrypted images. "Place" vertically in raster scan order to obtain the equivalent diffusion key E_{eq}. Then, according to Sec. 3.2.2, establish a data matrix to read the corresponding natural numbers in raster scan order. Second, calculate the minimum effective plaintext quantity to ensure the equivalent permutation key S_{eq}. Then, these valid plaintexts are numerically written according to the writing rules. The valid plaintexts in the previous step are put into the encryption machine for encryption, and the equivalent permutation key is obtained by comparing the positions of the same elements before and after encryption. Finally, the plaintext image is recovered by using the equivalent diffusion key and the equivalent permutation key. For the the grayscale image "Lena" of size 256×256, the experimental results cracked by the chosen-plaintext attack method are also shown in Fig. 6. For the grayscale image "Lake" of size 256×256, the experimental results cracked by the chosen-plaintext attack method are also shown in Fig. 7.



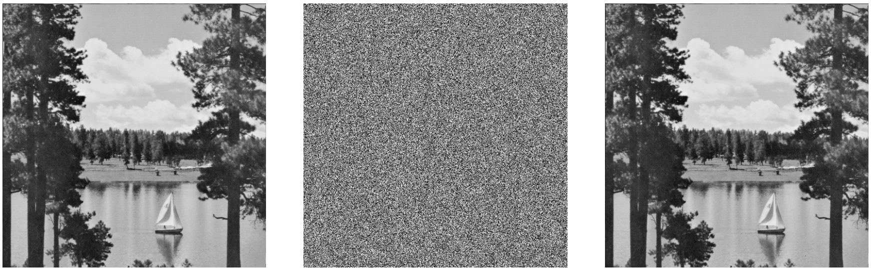
(a) Plaintext-image I (b) Cipher-image C (c) Recovered image P

Fig. 6 Results of the IEA-QCDC decoding experiment for "Lena" image

4.2 Attack Complexity

Attack complexity mainly includes time complexity and data complexity.

In terms of time complexity, using the chosen-plaintext attack method to crack the equivalent diffusion key E_{eq} of the IEA-QCDC for the grayscale



(a) Plaintext-image I (b) Cipher-image C (c) Recovered image P

Fig. 7 Results of the IEA-QCDC decoding experiment for "Lena" image

images of size 256×256 and size 512×512, the running time is 247.6118s and 1253.4549s, respectively. The running time of cracking the equivalent permutation key S_{eq} of IEA-QCDC is 0.006454s and 0.016487s respectively. Moreover, after obtaining the equivalent diffusion key E_{eq}, the time required to attack the subsequent ciphertext images will also be shorter. According to the above analysis, the time complexity of obtaining the equivalent diffusion key E_{eq} and the equivalent permutation keys S_{eq} are $O\left(M \times N\right)$ and $O\left(\left(M \times N\right)^2\right)$. Then for a ciphertext image of size 512×512, the duration of this attack process should be four times the duration of the ciphertext of size 256×256. The actual test result is 5.07 times.

Table 4: Comparison of time complexity

Time complexity	256×256	512×512	1024×1024
Ref. [23]	$O\left(256 \times 256\right)$	$O\left(\left(512 \times 512\right)^2\right)$	$O\left(\left(1024 \times 1024\right)^3\right)$
Ours	$O\left(256 \times 256\right)$	$O\left(\left(512 \times 512\right)^2\right)$	-
Ref. [8]	$O\left(\left(256 \times 256\right)^{4/3}\right)$	$O\left(\left(512 \times 512\right)^{4/3}\right)$	$O\left(\left(1024 \times 1024\right)^{4/3}\right)$

Table 5: Comparison of data complexity

Data complexity	Ref. [8]	Ours	Ref. [15]
$M \times N$	$O\left(2.5 \times \sqrt[3]{M \times N}\right)$	$O = O(\log_{256} M \times N)$	$\boldsymbol{O}\left(\lceil \log_4(MN) \rceil\right)$

In terms of data complexity, for the grayscale images of size $M \times N$, the chosen-plaintext attack method is used to crack the required $(256 + \lceil \log_{256}\left(M \times N\right) \rceil)$ plaintext images and the corresponding ciphertext images.

Then the corresponding data complexity is $O = O(\log_{256} M \times N)$. Therefore, the experimental results show that the chosen-plaintext attack method proposed in this manuscript can not only successfully crack the IEA-QCDC algorithm, but also completely recover the plaintext image without knowing any key-related information.

The comparison of time complexity and data complexity is shown in Table 4 and Table 5. In particular, the complexity comparison only measures the complexity of the decoding methods. Table 4 shows the comparison of the time complexity of the cracking algorithm. It can be seen that the time complexity of our scheme is relatively good. Table 5 shows the data complexity comparison of the cracking algorithm. It can be seen that the number of plaintext and ciphertext pairs selected by our scheme is the least.

5 Suggestions for improvement

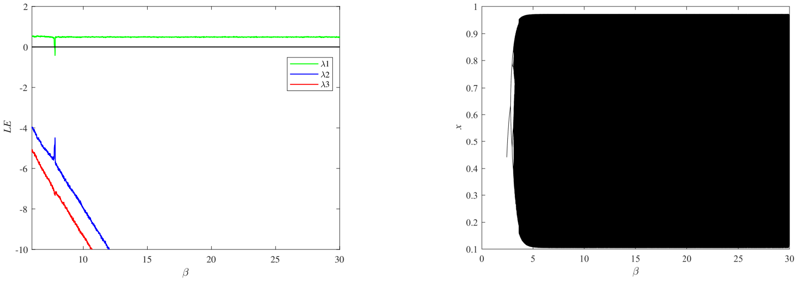
Based on the above analysis, the IEA-QCDC algorithm is insecure. The IEA-QCDC algorithm flexibly uses the DNA coding rules to encrypt the image, uses a variety of DNA operations in the diffusion, and performs multiple diffusions on the image from the bit level, which has a strong diffusion effect, but the permutation part is relatively weak. The IEA-QCDC algorithm introduces 17 initial keys, and its key space is very large, but it is found that its huge key space cannot guarantee its security during our cracking process. In addition, another weakness of the IEA-QCDC algorithm is that the entire encryption process has nothing to do with plaintext images, that is, the key stream remains unchanged when encrypting different plaintext images, making it unable to resist plaintext attacks. Some encryption methods related to DNA technology also have similar security vulnerabilities and defects. In the follow-up in-depth investigation, it is found that, for specific control parameter r described in the original text, the quantum Logistic system will have non-negligible non-chaotic phenomena within the range of parameter selection. At this time, the ciphertext-only attack can be used to crack it. The obtained sequences are poorly encrypted, and the advantage of using quantum Logistic map encryption is greatly reduced.

In order to improve the security of encryption algorithm, improve the efficiency of the use of chaotic sequences, and rationally use the parameters of the chaotic system, we give the following suggestions.

The following suggestions are given for the IEA-QCDC algorithm:

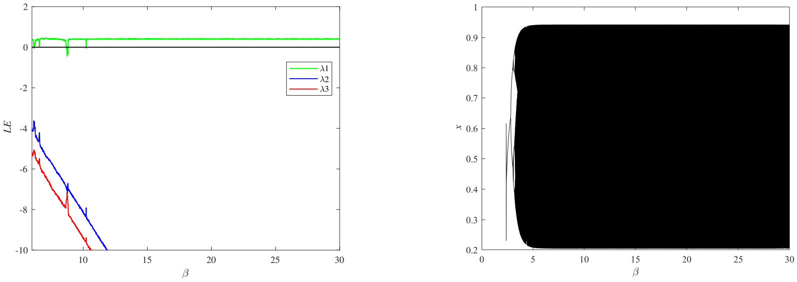
(1) Redefine the interval of control parameter r and dissipation parameter β of quantum Logistic chaotic map. It can be seen from Fig. 3a and Fig. 4a that the quantum Logistic system appears non-chaotic state when the parameters β are fixed and r is within the interval taken by the original text. The corresponding bifurcation graph is highly similar to the Logistic map, and does not show the advantage of uniform traversal in a larger range than the Logistic map. It can be seen from Fig. 8a and Fig. 9a that under the premise of choosing r to satisfy the chaotic characteristics, its corresponding bifurcation graph can

show the advantage that it is easier to traverse the interval than the Logistic map. The selection of the control parameter r will make the system in a non-chaotic state, thus weakening the influence of the generated sequence on the encryption. To obtain the chaotic sequence expected by the cipher designer, it is recommended not to use the control parameter r as the key parameter to ensure that the system is chaotic, and the obtained bifurcation graph also conforms to the original intention of the original cipher designer to use the quantum Logistic map.



(a) (b)

Fig. 8 Dynamical properties analysis of quantum Logistic map $r = 3.89$, $\beta \in [3, 4]$: (a) Lyapunov exponent spectrum (b) The bifurcation diagram



(a) (b)

Fig. 9 Dynamical properties analysis of quantum Logistic map $r = 3.77$, $\beta \in [6, 30]$: (a) Lyapunov exponent spectrum (b) The bifurcation diagram

(2) Improve the efficiency of the use of the chaotic sequence. The IEA-QCDC algorithm introduces too many key initial parameters into the chaotic system, resulting in too many iterations of the chaotic system and low utilization efficiency of the chaotic sequence. For example, when introducing chaotic

sequences a, b, c, and d, there are four different sets of initial parameters, but the lengths of the generated chaotic sequences are only $2M$, $2N$, and $4L$. It can be considered to select the chaotic sequence for each equation of the quantum Logistic map, in order to reduce the number of iterations of the chaotic sequence, reduce the encryption time and improve the efficiency of the use of the chaotic sequence.

(3) It is recommended to study a new permutation process to improve the obfuscation effect. Even if two rounds of encryption are used in the permutation process of the IEA-QCDC algorithm, the structure is relatively simple and can still be equivalent to one round of encryption. Because the permutation is often used to improve the key sensitivity and meet the designer's obfuscation requirements. Therefore, for each pixel permutation, pixel swapping of the entire plaintext image should be achieved. In the case of two rounds of row-by-row and column-by-column position swaps, it is possible to keep the pixel value positions unchanged, resulting in an invalid permutation.

6 Conclusion

This paper analyzes the security of the IEA-QCDC algorithm. First, without changing the structure and cryptographic properties of the original encryption algorithm, we make a supplementary description of its encrypted representation. Secondly, the cryptographic analysis of the IEA-QCDC algorithm is carried out, the encryption of the continuous DNA operation is analyzed, the mathematical properties of a single DNA encryption and multiple DNA encryption are summarized, and a feasible and effective chosen-plaintext attack method is proposed crack it. Again, this paper has carried out experiments and simulations, and the test results show that the security problems pointed out in this paper do exist, and the proposed cracking algorithm is also effective. Finally, this paper presents suggestions to improve the security and practicality of the IEA-QCDC algorithm.

At present, chaotic ciphers have the problems such as single permutation and easy to crack, we intend to consider designing and implementing a new secure and practical chaotic encryption algorithm in real-world communication. On the basis of this cryptanalysis, the design of chaotic image encryption algorithm with iterative redundancy and cumbersome encryption steps is avoided, and the development of chaotic encryption algorithm is promoted.

Declarations

- **Funding** This work was supported by the National Natural Science Foundation of China (No. 62271157), the Natural Science Foundation of Guangdong Province (No. 2022A1515010005).

- **Conflict of interest** The authors declare that there is no conflict of interest.
- **Availability of data and materials** All data generated or analysed during this study are included in this published article.

References

[1] Zhang, J. and Huo, D.: Image encryption algorithm based on quantum chaotic map and dna coding. Multimedia Tools and Applications, 78(11), 15605–15621 (2019)

[2] Talhaoui, M. Z. and X, Wang.: A new fractional one dimensional chaotic map and its application in high-speed image encryption. Information Sciences, 550, 13-269 (2021)

[3] Xian, Y., Wang, X., Yan, X., Li, Q. and Wang, X.: Image Encryption Based on Chaotic Sub-Block Scrambling and Chaotic Digit Selection Diffusion. Optics and Lasers in Engineering, 134, 106202 (2020)

[4] Pareek, N. K., Patidar, V. and Sud, K. K.: Image encryption using chaotic logistic map. Image and vision computing, 24(9), 926–934 (2006)

[5] Luo, Y., Yu, J., Lai, W. and Liu, L.: A novel chaotic image encryption algorithm based on improved baker map and logistic map. Multimedia Tools and Applications, 78(15), 22023-22043 (2019)

[6] Wang, Q., Yu, S., Guyeux, C. and Wang, W.: Constructing Higher-Dimensional Digital Chaotic Systems via Loop-State Contraction Algorithm. IEEE Transactions on Circuits and Systems I: Regular Papers, 68(9), 3794-3807 (2021)

[7] Signing, V. F., Tegue, G. G., Kountchou, M., Njitacke, Z., Tsafack, N., Nkapkop, J., Etoundi, C. L. and Kengne, J.: A cryptosystem based on a chameleon chaotic system and dynamic dna coding. Chaos, Solitons &Fractals, 155, 111777 (2022)

[8] Ma, Y., Li, C. and Ou, B.: Cryptanalysis of an image block encryption algorithm based on chaotic maps. Journal of Information Security and Applications, 54, 102566 (2020)

[9] Wen, H., Yu, S. and Lü, J. : Breaking an image encryption algorithm based on dna encoding and spatiotemporal chaos. Entropy, 21(3), 246 (2019)

[10] Chen, J., Zhu, Z., Zhang, L., Zhang, Y. and Yang, B. Q.: Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption[J]. Signal Processing, 142: 340-353 (2018)

[11] Wu, J., Liao, X. and Yang, B.: Image encryption using 2d h´enon-sine map and dna approach. Signal processing, 153, 11–23 (2018)

[12] Chai, X., Fu, X., Gan, Z., Lu, Y. and Chen, Y.: A color image cryptosystem based on dynamic DNA encryption and chaos. Signal Processing,

155, 44–62 (2019)

[13] Xuejing, K. and Zihui, G.: A new color image encryption scheme based on dna encoding and spatiotemporal chaotic system. Signal Processing: Image Communication, 80, 115670 (2020)

[14] Alawida ,M., Teh ,J. S. and Samsudin, A.: An image encryption scheme based on hybridizing digital chaos and finite state machine. Signal Processing, 164, 249-266 (2019)

[15] Jain, K., Aji, A. and Krishnan, P.:Medical Image Encryption Scheme Using Multiple Chaotic Maps. Pattern Recognition Letters, 152, 356-364 (2021)

[16] Zhao, D., Liu, L., Yu, F., Heidari, A. A., Wang, M., Liang, G. and Chen, H.: Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy. Knowledge-Based Systems, 216, 106510 (2021)

[17] Zhang, Z. and Yu, S.: On the security of a Latin-bit cube-based image chaotic encryption algorithm. Entropy, 21(9), 888 (2019)

[18] Chen, J., Chen, L. and Zhou, Y.: Cryptanalysis of a dna-based image encryption scheme. Information Sciences, 520, 130–141 (2020)

[19] Farah, M. A., A. Farah. and T, Farah.: An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. Nonlinear Dynamics, 99(4), 3041-3064 (2020)

[20] Sun, S.: A novel hyperchaotic image encryption scheme based on dna encoding, pixel-level scrambling and bit-level scrambling. IEEE Photonics Journal, 10(2), 1–14 (2018)

[21] Feng, W. and He, Y. G.: Cryptanalysis and improvement of the hyperchaotic image encryption scheme based on dna encoding and scrambling. IEEE Photonics Journal, 10(6), 1–15 (2018)

[22] Wang, X., Chen, S. and Zhang, Y.: A chaotic image encryption algorithm based on random dynamic mixing. Optics & Laser Technology, 138, 106837 (2021)

[23] Wang, S., Peng, Q. and Du, B.: Chaotic color image encryption based on 4D chaotic maps and DNA sequence. Optics & Laser Technology, 148, 107753 (2022)

[24] Lenstra, A. K. and Verheul, E. R.: Selecting cryptographic key sizes [J]. Journal of cryptology, 14(4), 255-293 (2001)

[25] Akhavan, A., Samsudin, A. and Akhshani, A.: Cryptanalysis of an image encryption algorithm based on dna encoding. Optics & Laser Technology, 95, 94–99 (2017)

[26] Su, X., Li, W. and Hu, H.: Cryptanalysis of a chaos-based image encryption scheme combining DNA coding and entropy[J]. Multimedia Tools and Applications, 76(12), 14021-14033 (2017)

[27] Dou, Y., Liu, X., Fan, H. and Li, M.: Cryptanalysis of a dna and chaos based image encryption algorithm. Optik, 145, 456–464 (2017)

[28] Liu, S., Li, C. and Hu, Q.: Cryptanalyzing two image encryption algorithmsbased on a first-order time-delay system. IEEE MultiMedia, 29(1), 74–84 (2021)

[29] Lin, H., Wang, C., Cui, L., Sun, Y., Zhang, X. and Yao, W.: Hyperchaotic memristive ring neural network and application in medical image encryption[J]. Nonlinear Dynamics, 110(1): 841-855 (2022)

[30] Cheng, G., Wang, C. and Xu, C.: A novel hyper-chaotic image encryption scheme based on quantum genetic algorithm and compressive sensing[J]. Multimedia Tools and Applications, 79(39), 29243-29263 (2020)

[31] O'Connor, L.: On the distribution of characteristics in bijective mappings[C]. Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, 360-370 (1993)

[32] Li, C. and Lo, K. T.: Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. Signal processing, 91(4), 949–954 (2011)

[33] Qader, S. M., Hassan, B. A. and Rashid, T. A.: An improved deep convolutional neural network by using hybrid optimization algorithms to detect and classify brain tumor using augmented MRI images. Multimed Tools Appl, 81, 44059–44086 (2022)

[34] Hassan, B. A.CSCF: a chaotic sine cosine firefly algorithm for practical application problems. Neural Comput & Applic, 33, 7011–7030 (2021)

[35] Khan, M. and Masood, F.: A novel chaotic image encryption technique based on multiple discrete dynamical maps. Multimedia Tools and Applications, 78(18), 26203-26222 (2019)