Multimed Tools Appl
DOI 10.1007/s11042-016-3800-9

Cryptanalysis of a chaos-based image encryption scheme combining DNA coding and entropy

Xin Su[1] · Weihai Li[1] · Honggang Hu[1]

Received: 21 November 2015 / Revised: 13 June 2016 / Accepted: 21 July 2016

Abstract An image encryption scheme based on chaos system combining with DNA coding and information entropy has been proposed recently, in which chaos system and DNA operation are used to perform substitution, and entropy driven chaos system is used to perform permutation. However, two vulnerabilities are found and presented in this paper, which make the encryption fail under chosen-plaintext attack. A complete chosen-plaintext attack algorithm is given to rebuild chaos systems' outputs and recover plain image, and its efficiency is demonstrated by analysis and experiments. Further, some improvements are proposed to make up these vulnerabilities and enhance the security.

Keywords Image encryption · Chosen-plaintext cryptanalysis · Chaotic system · DNA encoding · Information entropy

1 Introduction

In recent years, there has been an increasing interest at multimedia data security because of the wide-spread transmission over all kinds of communication and social networks. Considering the inherent features of image, such as bulk data capacity and strong correlation among adjacent pixels, traditional data encryption schemes cannot meet the requirements of multimedia security. The intrinsic relationship between chaos and cryptography inspired researchers and a variety of encryption schemes have been proposed [1, 2, 5, 6]. For

✉ Weihai Li
whli@ustc.edu.cn

[1] CAS Key Laboratory of Electromagnetic Space Information, School of Information Science and Technology, University of Science and Technology of China, Hefei, 230027, China

example, Chen et al. [1] designed a three dimensional Arnold cat map, and used it in a symmetrical image encryption algorithm. Guan et al. [2] proposed a fast image encryption design according to the character of hyper-chaos. Unfortunately, many cryptanalysis works [4, 9, 11, 12, 14] have revealed that some of those chaos-based cryptosystems have serious security problems due to their improper use of chaos pseudo-random sequences. Rhouma et al. [9] gives a way that just demands three couples of plaintext/ cipher text to break totally the crypt-system.

In 1994, Adleman first introduced DNA computing into the encryption field, which created a new stage of information security. Due to massive parallelism and extraordinary information density exclusive characteristic of DNA molecule, DNA cryptography emerged as a new frontier and is presently at the forefront of international cryptography research [8, 13]. DNA encryption is a subject of study how to utilize DNA bases as an information carrier, and it uses modern biological technology to achieve encryption. Many symmetric encryption schemes have been proposed, in which the DNA-based image encryption is generally categorized into two phases: firstly, using DNA theory to encode plain image pixels into DNA sequence, and each gray pixel value is decomposed into four DNA elements, which can increase the efficiency of image substitution and permutation. Secondly, the encoded DNA sequence is substituted and permuted to generate a key image based on DNA operation rules and form the cipher image [3, 7, 10].

Recently, a novel image encryption algorithm [15] was designed based on chaos system and combined with DNA coding and information entropy. It mainly consists of two stages of process: substitution and permutation process. Encryption is operated on the domain of DNA encoding matrix, and entropy is introduced to generate various permutation matrix which is expected to enhance security against known/chosen plaintext attacks.

However, we notice that two vulnerabilities exist in this algorithm. First, the introduced entropy fails to protect permutation indexes under chosen-plaintext attack, because we can rebuild the entropy from cipher image directly rather than by breaking encrypted entropy in cipher. And second, the substitution of elements in the last column leaks patterns of the encoding rule, and which permit us to recover the encoding rule and cover matrix after breaking the permutation. Then a complete attack algorithm is presented in this paper based on these two vulnerabilities, and our analysis and experiments demonstrate that this algorithm can break the encryption algorithm under chosen-plaintext attack. Further, some improvements are proposed to make up these vulnerabilities and enhance encryption algorithm's security.

The remaining of this paper is organized as follows. The next section gives a brief introduction of the original encryption algorithm. The vulnerabilities and the presented chosen-plaintext attack is discussed detailedly in Section 3. In Section 4, some improvement are proposed to enhance the security. And finally, the last section gives a brief conclusion.

2 The original encryption algorithm

2.1 Process of the encryption

The block diagram of the original image encryption algorithm [15] is shown in Fig. 1.

To perform the encryption, the plain image I_{input} of size $m \times n$ is firstly converted into an $m \times 4n$ matrix I_d by decomposing each pixel into four two-bit numbers. Then the matrix I_d is encrypted with the following two main stages:

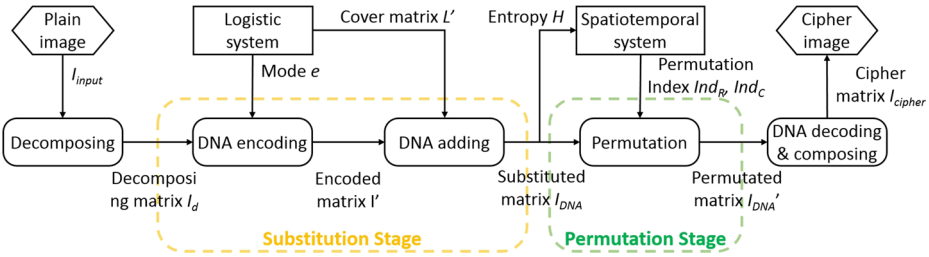


Fig. 1 Block diagram of the image encryption algorithm [15]

(1) Substitution stage

Iterate the logistic chaotic map (1) with the initial state (x_0^l, u_0^l) to get a real number sequence $L_1 = \{x_1, x_2, \cdots, x_{4n\times m}\}$.

$$f(x,u) = ux(1-x) \tag{1}$$

Let $\lfloor x \rfloor$ denote the largest integer less than or equal to x. Compute terms $L(i,j) = \lfloor x_{(i-1)\times 4n+j} \times 2^8 \rfloor \mod 4 (1 \le i \le m, 1 \le j \le 4n)$ to get the integer matrix L, and encode L to cover matrix L' according to rule (2). At the same time, compute encoding modes $e(i) = \lfloor x_{i\times 4n} \times 2^8 \rfloor \mod 8$ and encode I_d to matrix I' (the i-th row is encoded following DNA encoding rule $e(i)$).

$$L'(i,j) = \begin{cases} A \ if \ L(i,j) = 0 \\ C \ if \ L(i,j) = 1 \\ G \ if \ L(i,j) = 2 \\ T \ if \ L(i,j) = 3 \end{cases} \tag{2}$$

Then perform the DNA addition to get the DNA matrix I_{DNA} by $I_{DNA} = I' + L'$. The eight encoding rules for the DNA sequence are listed in Tables 1 and 2 describes DNA addition operation.

(2) Permutation stage

$$x_{n+1}(i) = (1-\varepsilon) f(x_n(i), u) + \frac{\varepsilon}{2}[f(x_n(i+1), u) + f(x_n(i-1), u)] \tag{3}$$

A 3-dimensional spatiotemporal chaotic system (3), with parameter u^s, ε and initial values $x_0^s(1), x_0^s(2), x_0^s(3)$, is adopted to generate two sequences $R = \{x_i(1), 1 \le i \le m\}$ and $C = \{x_j(3), 1 \le j \le 4n\}$. Then the DNA matrix is permutated as $I'_{DNA}(i,j) = I_{DNA}(Ind_R(i), Ind_C(j))$, in which the Ind_R and Ind_C are the sorting indexes of sequences R and C respectively. To avoid chosen-plaintext attack, entropy H of I_{DNA} is introduced to modify u^s thus different plain image will have different permutation indexes. First, the decimal part of entropy H is used as x_0^H (if H is an

Table 1 The DNA encoding and decoding rules

Code	0	1	2	3	4	5	6	7
0	A	A	C	C	G	G	T	T
1	C	G	A	T	A	T	C	G
2	G	C	T	A	T	A	G	C
3	T	T	G	G	C	C	A	A

Table 2 The DNA addition operation

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

integer, then $x_0^H = x_0^l$), and generate x_{20}^H with the logistic map (1). Keep the first 64 bits of x_{20}^H as $H_{decimal}$, and $u^s = 3.75 + 0.25 H_{decimal}$.

Finally, the I'_{DNA} is DNA decoded with the first rule in Table 1 and composed back into an 8-bit cipher image I_{cipher}. The encrypted $H_{decimal}$ and I_{cipher} are sent as cipher text.

2.2 Encryption examples

Here we give a simple example to show the process of encryption. It is also an attack target for our cryptanalysis example in Section 3.

The plain image I is first decomposed into a matrix I_d.

$$I = \begin{array}{|ccc|}\hline 177 & 112 & 47 \\ 81 & 97 & 125 \\ 243 & 195 & 114 \\ 8 & 203 & 165 \\ \hline \end{array}$$

$$I_d = \begin{array}{|cccccccccccc|}\hline 2&3&0&1&1&3&0&0&0&2&3&3 \\ 1&1&0&1&1&2&0&1&1&3&3&1 \\ 3&3&0&3&3&0&0&3&1&3&0&2 \\ 0&0&2&0&3&0&2&3&2&2&1&1 \\ \hline \end{array}$$

For $key = \{x_0^l = 0.437, u^l = 3.785, x_0^s = (0.364, 0.785, 0.293), \varepsilon = 0.2582\}$, the cover matrix L' and the encoding mode e are

$$L' = \begin{array}{|cccccccccccc|}\hline G&G&C&C&C&A&A&C&T&G&C&C \\ G&T&G&T&A&T&T&T&A&C&T&T \\ A&A&C&G&C&T&A&G&G&C&C&G \\ C&T&C&A&G&A&G&C&A&A&T&G \\ \hline \end{array}$$

$$e = (1, 7, 2, 2).$$

and I_d is encoded as

$$I' = \begin{array}{|cccccccccccc|}\hline C&T&A&G&G&T&A&A&A&C&T&T \\ G&G&T&G&G&C&T&G&G&A&A&G \\ G&G&C&G&G&C&C&G&A&G&C&T \\ C&C&T&C&G&C&T&G&T&T&A&A \\ \hline \end{array}$$

Then the substituted matrix

$$I_{DNA} = I' + L' = \begin{array}{|cccccccccccc|}\hline G&A&A&G&G&G&T&A&G&G&T&T \\ T&A&A&A&C&T&C&A&C&A&G&A \\ C&C&C&T&G&T&A&T&C&G&C&A \\ C&T&T&A&T&A&A&G&G&G&G&C \\ \hline \end{array}$$

Fig. 2 Image Lena and its cipher image

Count the numbers of C, A, T, G, they are #(C)=10, #(A)=14, #(T)=11, #(G)=13. Then the entropy $H \approx 1.9874$, and $u^s \approx 3.9321$. Run the spatiotemporal system, and sort the generated sequences R and C, we get

$$Ind_R = (2, 3, 4, 1),\ and\ Ind_C = (1, 2, 10, 7, 3, 11, 8, 5, 4, 12, 6, 9).$$

Then the permutated matrix

$$I'_{DNA} = \begin{array}{cccccccccccc} \mathrm{T} & \mathrm{A} & \mathrm{A} & \mathrm{C} & \mathrm{A} & \mathrm{G} & \mathrm{A} & \mathrm{C} & \mathrm{A} & \mathrm{A} & \mathrm{T} & \mathrm{C} \\ \mathrm{C} & \mathrm{C} & \mathrm{G} & \mathrm{A} & \mathrm{C} & \mathrm{C} & \mathrm{T} & \mathrm{G} & \mathrm{T} & \mathrm{A} & \mathrm{T} & \mathrm{C} \\ \mathrm{C} & \mathrm{T} & \mathrm{G} & \mathrm{A} & \mathrm{T} & \mathrm{G} & \mathrm{G} & \mathrm{T} & \mathrm{A} & \mathrm{C} & \mathrm{A} & \mathrm{G} \\ \mathrm{G} & \mathrm{A} & \mathrm{G} & \mathrm{T} & \mathrm{A} & \mathrm{T} & \mathrm{A} & \mathrm{G} & \mathrm{G} & \mathrm{T} & \mathrm{G} & \mathrm{G} \end{array}$$

Decode I'_{DNA} and compose the matrix, we get the final cipher image I_{cipher}.

$$I_{cipher} = \begin{array}{ccc} 193 & 33 & 13 \\ 88 & 94 & 205 \\ 120 & 235 & 18 \\ 139 & 50 & 186 \end{array}$$

For another example, the gray image Lena is encrypted with the same key, and the result is shown in Fig. 2.

3 Chosen-plaintext cryptanalysis

Since the first step "decomposition" and the last step "DNA decoding & composing" in encryption algorithm are independent with the key, we will bypass them in following discussion for simplicity. The to-be-broken cipher matrix is denoted by I'_{DNA_1}.

Although entropy has been introduced to affect the parameter of the spatiotemporal system, and thus the permutation indexes may differ for different images, we found it is still possible to create an image which has the same permutation index as special cipher image used. We name this image a "good-entropy" image. In the following, the method of creating good-entropy images will be described firstly, and it will be used to rebuild the permutation indexes Ind_R and Ind_C; then a further method is given to recover the e and L'. Finally, the target plain image can be decrypted with the e, L', Ind_R and Ind_C.

3.1 Method to generate a good-entropy plain image

Notice that the spatiotemporal system is driven by the key and the entropy of I_{DNA}, so a plain image who generates the same I_{DNA} entropy will have the same spatiotemporal system intput and also have the same permutation indexes. Therefore the focus is how to create a plain image with a specific I_{DNA} entropy.

There is a clue for the I_{DNA} entropy: The numbers of C, A, T, G are the same before and after permutation because permutation changes positions but leaves values unchanged. So the entropy of I_{DNA} is equal to the entropy of I_{cipher}. In fact, we do not need to send the encrypted $H_{decimal}$ in the encryption algorithm, it can be calculated from the cipher image with the key.

Based on this clue, we can design a greedy algorithm which tunes elements one by one in the decomposed plain matrix I_d, while observe the numbers of C, A, T, G in the cipher image, until they are all equal to those in the target cipher image. The input of this algorithm is a random image, and the output image has the same permutation indexes as the target cipher image has. The detail of the greedy algorithm GA() is given in Table 3.

Note elements in the last column in I_{d0} are kept unchanged, because this column will be used to recover the modes e. This is discussed in Section 3.3.

3.2 Rebuild the permutation indexes Ind_R and Ind_C

Once we make a good-entropy image matrix, denoted I_{d0}, it is possible to rebuild the permutation indexes.

Suppose we have another image matrix I_{d1} that is equal to I_{d0}, except two elements (at position a and b) are different: $I_{d0}(a) \neq I_{d1}(a)$ and $I_{d0}(b) \neq I_{d1}(b)$. If the two elements' substitution results are swapped, i.e. $S(I_{d0}(a)) = S(I_{d1}(b))$ and $S(I_{d0}(b)) = S(I_{d1}(a))$, the numbers of C,A,T,G in cipher matrixes for I_{d0} and I_{d1} will be equal, then we know I_{d1} is also a good-entropy image matrix. Here we define function $f(I(a), e)$ represents element $I(a)$ being DNA encoded by rule e, and function $S(I(a)) = f(I(a), e) + L'(a)$ represents the substituted element at position a (Fig. 3).

If we limit the two elements in the same row i ($1 \leq i \leq m$), they will be permuted to a same row j ($1 \leq j \leq m$). By observing the difference between their cipher matrixes Enc(I_{d0}) and Enc(I_{d1}), only two different elements exist in one row, we can find where the

Table 3 Greedy algorithm to create good-entropy images $I_{dt} = GA(I_{ds}, I'_{DNA\_t})$

Input:	Starting plain matrix I_{ds}, target cipher matrix I'_{DNA_t}
Output:	Good-entropy image matrix I_t, which has the same permutation indexes as I_{cipher} used

EC = Count(I'_{DNA_t});
AC = Count(enc(I_{ds}));
for i = 1 to m
for j = 1 to 4n-1 {
if(EC==AC) return($I_{dt} = I_{ds}$);
Try $I_{ds}(i, j) = 0$ to 3, keep the one its AC = Count(Enc(I_{ds})) closest to EC
}	

Note:	Image matrix size is $m \times 4n$ Count(I_c): Return the numbers of C,A,T,G in cipher matrix I_c Enc(I_{d0}): Encrypt plain matrix I_{d0}, and return its cipher matrix I'_{DNA}

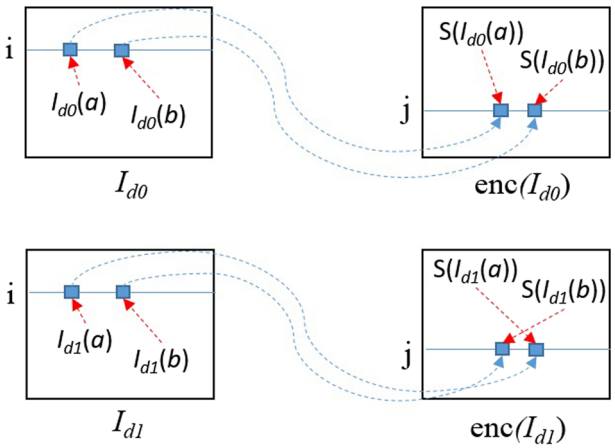


Fig. 3 Construct plain image with equal entropy by changing two elements

j is. Repeat the work for each row, and the permutation index Ind_R is rebuilt. We name this process PI_R() and details are given in Table 4.

Similar algorithm PI_C() is performed to rebuild the column permutation index Ind_C, in which all works on rows are changed to columns.

It is also worth to mention that since there are only four possible values for each elements in plain matrix and cipher matrix, it is easy to find two proper elements and their modifications.

3.3 Recover the encoding modes e and cover matrix L'

The next work is to recover the encoding modes e.

We noticed that the encoding modes and cover matrix for the last column are both from logistic sequence $x_{4n\times i}$. For example, element $I_d(1, 4n)$ is encoded with rule $e(1) = \lfloor x_{4n} \times 2^8 \rfloor \mod 8$ and added with cover $L'(1, 4n)$, which is mapped from $L(1, 4n) = \lfloor x_{4n} \times 2^8 \rfloor \mod 4 = e(1) \mod 4$. The substitution results $f(I(*, 4n), e)+L'(*, 4n)$ for element value $I(*, 4n) =0$ and 3 are listed in Table 5.

It can be seen that the pairs $(f(0, e) + L', f(3, e) + L')$ uniquely decide the encoding rule e, and the mapping is given in Table 6.

Table 4 Algorithm to rebuild row permutation index $Ind_R = PI\_R(I_{ds})$

Input:	Plain matrix I_{ds}
Output:	Row permutation index Ind_R

```
for i = 1 to m{
    repeat
        I_dsi = I_ds;
        randomly select two elements a and b in i-th row in I_ds;
        randomly modify values of elements a and b in I_dsi;
    until Count(Enc(I_ds)) = Count(Enc(I_dsi));
    find the row j that is different between Enc(I_ds) and Enc(I_dsi)
    Ind_R(j) = i;
    }
```

Table 5 Substituted results for element value 0 or 3 in the last column

e	0	1	2	3	4	5	6	7
$L = e \mod 4$	0	1	2	3	0	1	2	3
L'	A	C	G	T	A	C	G	T
$f(0, e)$	A	A	C	C	G	G	T	T
$f(0, e) + L'$	T	A	G	T	C	G	A	C
$f(3, e)$	T	T	G	G	C	C	A	A
$f(3, e) + L'$	G	T	T	A	A	C	C	G

To perform the recovery, we first prepare two plain images I_{df0} (a matrix in which all elements are 0) and I_{df3} (a matrix in which all elements are 3), tune them with algorithm $I_{d0} = GA(I_{df0}, I'_{DNA\_1})$ and $I_{d3} = GA(I_{df3}, I'_{DNA\_1})$, so that they are both good-entropy image matrixes. Note the last column will not be changed in GA(), i.e. elements in the last column of I_{d0} are all 0 and that of I_{d3} are all 3. Encrypt them, and inverse-permute the cipher matrixes, so that we can get the substitution results of the last columns. Then the encoding rule e can be obtained by looking up Table 6 according to the last columns.

To recover the cover matrix L' is easy. A simple DNA subtraction between substitution result and encoded I_{d0} can make it.

3.4 The entire attack process

The entire attack process is given in Table 7, which includes:

Step 1: Generate two good-entropy image matrixes I_{d0} and I_{d3} with the greedy algorithm GA(), in which elements in the last columns are 0 or 3 respectively. Then they are encrypted as I'_{DNA0} and I'_{DNA3}.

Step 2: I_{d0} is used to rebuild the permutation indexes Ind_R and Ind_C with algorithm PI_R() and PI_C().

Step 3: The cipher matrixes of I'_{DNA0} and I'_{DNA3} are inverse permuted. The last columns of them are used to recover the DNA encoding rule e according to Table 6.

Step 4: I_{d0} is DNA encoded with rule e, then DNA subtracted from the inverse permuted cipher matrixes of I_{d0}. The result recovers the cover matrix L'.

Step 5: Use the Ind_R, Ind_C, e and L' to decrypt the cipher image with normal decryption algorithm.

Table 6 Encoding rule e uniquely decided by substituted elements of two cases

Encoding		$f(0, e) + L'$			
rule e		A	C	G	T
$f(3, e) + L'$	A	–	6	–	1
	C	4	–	7	–
	T	–	5	–	2
	G	3	–	0	–

Table 7 The attack algorithm

Input:	Cipher image I_{cipher}
Output:	Plain image I_{input}

I'_{DNA_t} = Decompose and decode I_{cipher} with DNA encode rule 0;
$[m, n]$ = sizeof I_{cipher};
I_{d0} = GA(matrix of 0 of size $m \times 4n$, I'_{DNA_t}); //generate a good-entropy image matrix //with element 0s in the last column
I_{d3} = GA(matrix of 3 of size $m \times 4n$, I'_{DNA_t}); //generate a good-entropy image matrix //with element 3s in the last column
I'_{DNA0} = Enc(I_{d0}); I'_{DNA3} = Enc(I_{d3});
Ind_R = PI_R(I_{d0}); Ind_C = PI_C(I_{d0}); //rebuild the permutation indexes
I_{DNA0} = inverse permute I'_{DNA0}; I_{DNA3} = inverse permute I'_{DNA3};
for $i = 1$ to m
  $e(i)$ = look up Table 6 by ($I_{DNA0}(i, 4n)$, $I_{DNA3}(i, 4n)$); //recover the encoding rule e
I'_0 = DNA encode I_{d0} with rule e
$L' = I_{DNA0} - I'_0$; //recover the cover matrix L'
I_{input} = decrypt I'_{DNA_t} with Ind_R, Ind_C, L', e. //decrypt image

3.5 Attacks on examples in Section 2.2

As an example, we will attack the matrix I_{cipher} in Section 2.2 first.

In step 1, two good-entropy image matrixes I_0 and I_3 are generated according to the matrix I'_{DNA_t}.

$$I_{d0} = \begin{array}{|cccccccccccc|}\hline 1&1&1&0&0&0&0&0&0&1&0&0\\ 0&2&0&2&0&3&3&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ \hline\end{array}$$

$$I_{d3} = \begin{array}{|cccccccccccc|}\hline 3&3&0&0&0&2&2&0&3&3&3&3\\ 3&1&3&3&3&3&3&3&3&3&3&3\\ 3&3&3&3&3&3&3&3&3&3&3&3\\ 3&3&3&3&3&3&3&3&3&3&3&3\\ \hline\end{array}$$

The cipher matrixes are:

$$I'_{DNA0} = \begin{array}{|cccccccccccc|}\hline A&T&T&G&A&C&C&G&T&C&G&G\\ A&A&C&A&C&C&G&C&G&G&T&G\\ C&T&A&G&C&T&C&G&A&G&A&A\\ T&T&T&T&G&A&A&A&A&A&T&G\\ \hline\end{array}$$

$$I'_{DNA3} = \begin{array}{|cccccccccccc|}\hline C&A&A&G&C&G&G&T&G&G&G&T\\ C&C&G&C&G&G&T&G&T&T&A&T\\ G&A&C&T&G&A&G&T&C&T&C&C\\ A&A&A&A&A&T&A&A&A&T&A&C\\ \hline\end{array}$$

Count the numbers of C, A, T, G in both matrixes, they are #(C)=10, #(A)=14, #(T)=11, #(G)=13, equal to the result of I'_{DNA}.

In step 2, permutation indexes Ind_R and Ind_C are rebuilt and equal to those shown in Section 2.2.

Table 8 Recover of rule e for matrix example in Section 2.2

Last column of I_{DNA0}	Last column of I_{DNA3}	Encoding rule e
A	T	1
C	G	7
G	T	2
G	T	2

In step 3, I_{e0} and I_{e3} are inverse permuted, and we get:

$$I_{DNA0} = \begin{array}{cccccccccccc} T & T & G & A & A & T & T & A & G & T & A & A \\ A & T & A & T & G & G & G & C & G & T & C & C \\ A & A & C & G & C & T & A & G & G & C & C & G \\ C & T & C & A & G & A & G & C & A & A & T & G \end{array}$$

$$I_{DNA3} = \begin{array}{cccccccccccc} A & A & A & A & A & A & A & A & C & A & T & T \\ C & A & C & G & T & G & G & G & T & A & G & G \\ C & C & G & T & G & A & C & T & T & G & G & T \\ G & A & G & C & T & C & T & G & C & C & A & T \end{array}$$

Take the last column to look up Table 6, we get the encode rule e. This is shown in Table 8.

In step 4, I_{d0} is DNA encoded with rule e,

$$I'_0 = \begin{array}{cccccccccccc} G & G & G & A & A & A & A & A & A & G & A & A \\ T & C & T & C & T & A & A & T & T & T & T & T \\ C & C & C & C & C & C & C & C & C & C & C & C \\ C & C & C & C & C & C & C & C & C & C & C & C \end{array}$$

Compute $L' = I_{DNA0} - I'_0$, we can get the same cover matrix as given in Section 2.2.

Finally in step 5, we use the Ind_R, Ind_C, e and L' to decrypt the cipher image with normal decryption algorithm, and the decryption result is equal to original input matrix I.

For the example of encrypted Lena, the attack result is shown in Fig. 4.

4 Improvement on encryption algorithm

There are two vulnerabilities in the original encryption algorithm: one is the entropy fails to play the role in protecting permutation indexes; and the other is the encoding rule and cover matrix for elements in last column are based on a same number, which leaves a pattern in cipher and cause the rule be recovered. Therefore, our improvement aims at these two vulnerabilities.

(1) To protect the permutation indexes

Instead of using entropy of I_{DNA}, a hash function is much better to generate a finger print of I_{DNA}. For example, we can DNA decode I_{DNA} using rule 0, and reform all elements into a 0-1 sequence. Then standard SHA algorithm can be applied to hash the sequence, and the hash value is used to replace the entropy value.

We know hash value can be viewed as a fingerprint of data. No matter what changes in numbers or positions of C,A,T,G, it will lead to a different hash value. So, although

Fig. 4 Revealed image Lena

I_{DNA} and I'_{DNA} have same numbers of C,A,T,G, they don't have same hash value, and attackers can't guess I_{DNA} from I'_{DNA}.

Also, the hash value needs to be encrypted and sent.

(2) To improve the substitution process

To separate the source of encoding rule and cover matrix, we can assign a private value to e by generating longer chaos sequences. Namely,

$$L_1 = \{x_1, x_2, \cdots, x_{4n}\}, \quad e_1 = \lfloor x_{4n+1} \times 2^8 \rfloor \mod 8,$$

$$L_2 = \{x_{4n+2}, x_{4n+3}, \cdots, x_{8n+1}\}, \quad e_2 = \lfloor x_{8n+2} \times 2^8 \rfloor \mod 8.$$

and so on.

Such, the relation between encoding rule and cover matrix will be protected by the property of chaos system.

Further, if we use another logistic sequence driven by appending key (x_0^e, u_0^e) to calculate encoding rule e, and use original logistic sequence driven by (x_0^l, u_0^l) to calculate cover matrix L', the relation between encoding rule and cover matrix can be protected more safely.

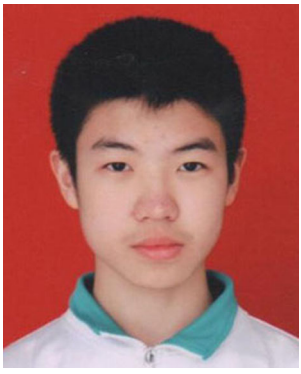
5 Conclusion

In this paper, we present a cryptanalysis on an encryption algorithm which based on chaos system and combined with DNA coding and information entropy. Two vulnerabilities are found, and they make the substitution based on DNA coding and operation fail to cover plain data, and introduced entropy fails to protect permutation indexes under chosen-plaintext attack. A complete attack algorithm is given, and our analysis and experiments demonstrate

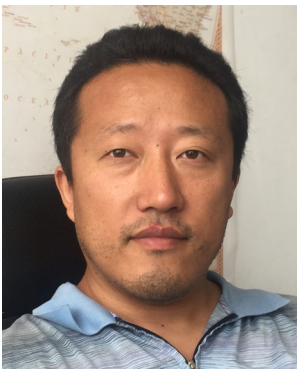
its effectiveness. Finally, some advices are proposed to make up these vulnerabilities and enhance encryption algorithm's security.

References

1. Chen GR, Mao YB, Chui CK (2004) A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Soliton Frac 21(3):749–761. doi:10.1016/j.chaos.2003.12.022
2. Guan ZH, Huang FJ, Guan WJ (2005) Chaos-based image encryption algorithm. Phys Lett A 346(1-3):153–157. doi:10.1016/j.physleta.2005.08.006
3. Halvorsen K, Wong WP (2012) Binary DNA nanostructures for data encryption. Plos One 7(9). ARTN e44212 doi:10.1371/journal.pone.0044212
4. Li SJ, Li CQ, Chen GR, Lo KT (2008) Cryptanalysis of the RCES/RSES image encryption scheme. J Syst Softw 81(7):1130–1143. doi:10.1016/j.jss.2007.07.037
5. Masuda N, Jakimoski G, Aihara K, Kocarev L (2006) Chaotic block ciphers: from theory to practical algorithms. IEEE T Circuits-I 53(6):1341–1352. doi:10.1109/Tcsi.2006.874186
6. Mirzaei O, Yaghoobi M, Irani H (2012) A new image encryption method: parallel sub-image encryption with hyper chaos. Nonlinear Dynam 67(1):557–566. doi:10.1007/s11071-011-0006-6
7. Mousa H, Moustafa K, Abdel-Wahed W, Hadhoud M (2011) Data hiding based on contrast mapping using DNA medium. Int Arab J Inf Techn 8(2):147–154
8. O'Driscoll C (2009) DNA encryption On The Origin of Species could be archived in bacteria. Chem Ind-London 6:10–10
9. Rhouma R, Belghith S (2008) Cryptanalysis of a new image encryption algorithm based on hyper-chaos. Phys Lett A 372(38):5973–5978. doi:10.1016/j.physleta.2008.07.057
10. Shoshani S, Piran R, Arava Y, Keinan E (2012) A molecular cryptosystem for images by DNA computing. Angew Chem Int Edit 51(12):2883–2887. doi:10.1002/anie.201107156
11. Solak E (2009) Cryptanalysis of image encryption with compound chaotic sequence. In: 2009 6th international multi-conference on systems, signals and devices, vol 1 and 2, pp 317–321
12. Solak E, Cokal C, Yildiz OT, Biyikoglu T (2010) Cryptanalysis of Fridrich's chaotic image encryption. Int J Bifurcat Chaos 20(5):1405–1413. doi:10.1142/S0218127410026563
13. Xiao GZ, Lu MX, Lei Q, Lai XJ (2006) New field of cryptography: DNA cryptography. Chinese Sci Bull 51(12):1413–1420. doi:10.1007/s11434-006-2012-5
14. Xiao D, Liao XF, Wei PC (2009) Analysis and improvement of a chaos-based image encryption algorithm. Chaos Soliton Frac 40(5):2191–2199. doi:10.1016/j.chaos.2007.10.009
15. Zhen P, Zhao G, Min LQ, Jin X (2015) Chaos-based image encryption scheme combining DNA coding and entropy. Multimed Tools Appl:1–17. doi:10.1007/s11042-015-2573-x



Xin Su was born in Tianjin Province, China. He is an undergraduate in School of Information Science and Technology, University of Science and Technology of China. His research interests include information security, cryptography and cryptanlysis.



Weihai Li received his Ph.D. degree from University of Science and Technology of China in 2003. He is currently an associate professor in Information Security Department at University of Science and Technology of China. His research interests include multimedia content security, video surveillance, and remote sensing processing.

Honggang Hu received the B.S. degree in mathematics in 2000, and the B.E.degree in electrical engineering in 2001, both from the University of Science and Technology of China (USTC), Hefei, China, and the Ph.D. degree in electrical engineering from the Chinese Academy of Sciences (CAS), Beijing, China, in2005. From July 2005 to April 2007, he was a Postdoctoral Fellow at the Institute of Software, CAS. From August 2007 to July 2009, he was a Postdoctoral Fellow at the University of Waterloo, Canada, and from August 2009 to August 2011, he was a Research Associate at the University of Waterloo, Canada. Since September 2011, he has been a Professor with USTC. His research interests include pseudorandom sequences, cryptography, and coding theory.