Optik 145 (2017) 456–464

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Original research article

Cryptanalysis of a DNA and chaos based image encryption algorithm

Yuqiang Dou[a], Xiumin Liu[b], Haiju Fan[a,c], Ming Li[a,d,*]

[a] College of Computer and Information Engineering, Henan Normal University, Xinxiang, 453007, China
[b] Shangqiu Polytechnic, Shangqiu, 476000, China
[c] China National Digital Switching System Engineering and Technological R&D Center, Zhengzhou, 450002, China
[d] School of Automation Science and Electrical Engineering, Beihang University, Beijing, 100191, China

ARTICLE INFO

Article history:
Received 26 April 2017
Accepted 4 August 2017

Keywords:
DNA
Chaos
Image encryption
Cryptanalysis
Chosen-plaintext attack

ABSTRACT

A robust encryption algorithm using DNA and chaotic logistic maps was recently proposed, which can resist exhaustive attack, statistical attack and differential attack. The security of this encryption algorithm depends on the initial conditions of 1D and 2D logistic chaotic maps, which include the front and latter pixel sum of plain image. However, this scheme is not secure enough because the front half and latter half pixel sum can be known to the attacker. According to this security weakness the encryption algorithm is completely cracked by a novel chosen-plaintext attack scheme. Experiments and analysis verify our approach.

1. Introduction

Security of image data transmitted over public network has been a critical issue for researchers. A variety of encryption schemes [1–8] using chaotic maps have been proposed since it has some excellent properties. Recently, an increasing number of image encryption algorithms [9–17] based on both chaos and DNA algorithms have been introduced in literature. DNA computing supports massive parallelism, huge information density and ultra-low power consumption. At the same time, many cryptanalysis works [18–23] have been proposed correspondingly.

In [24], a robust image encryption algorithm using DNA and chaotic logistic maps was introduced. The image pixels were transformed using DNA sequence at first and were added with a DNA matrix generated by one logistic chaotic map. The DNA complement process was realized with the help of another logistic map. The transformed image was divided into equal size blocks and these blocks were shuffled according to two chaos sequences generated by 2D logistic map. However, this is not enough to make the cryptosystem secure. We analyze the security weakness of this scheme and break it by a chosen-plaintext attack scheme.

The rest of this paper is organized as follows. In the next section, the original encryption algorithm is described in Section 2. Section 3 gives the cryptanalysis by chosen plaintext attack in detail. The experimental results and discussions are reported in Section 4. In the last section conclusions are drawn.

[*] Corresponding author at: College of Computer and Information Engineering, Henan Normal University, Xinxiang, 453007, China.
E-mail addresses: douyuqiang@htu.edu.cn (Y. Dou), liuxm2006@163.com (X. Liu), fanhaiiju8706@163.com (H. Fan), liming@htu.edu.cn (M. Li).

http://dx.doi.org/10.1016/j.ijleo.2017.08.050

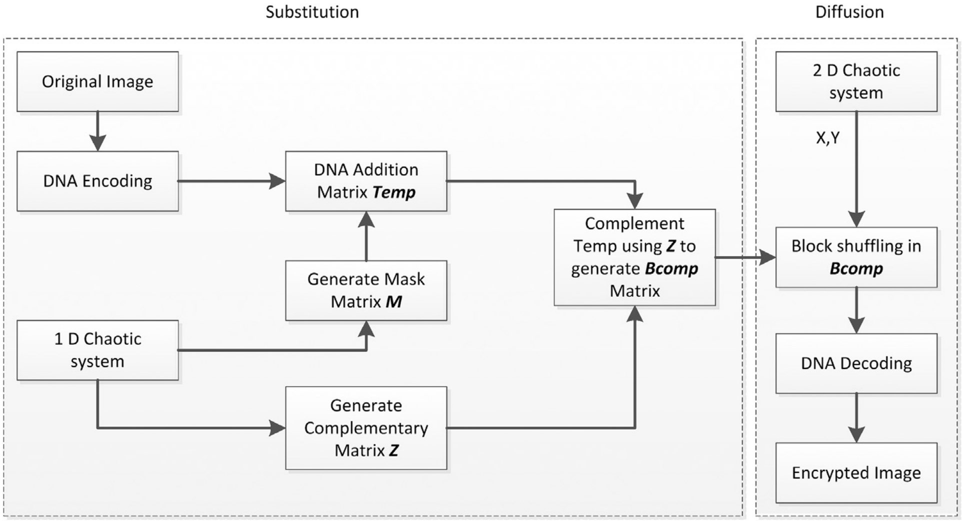
Substitution
Diffusion
Original Image
DNA Encoding
DNA Addition Matrix *Temp*
Generate Mask Matrix *M*
1 D Chaotic system
Generate Complementary Matrix *Z*
Complement Temp using *Z* to generate *Bcomp* Matrix
2 D Chaotic system
X,Y
Block shuffling in *Bcomp*
DNA Decoding
Encrypted Image

Fig. 1. Block diagram of the original image encryption algorithm.

2. Review of the original scheme

The sketch of the original scheme is shown in Fig. 1. The encryption process can be divided into two stages: substitution and diffusion. The two stages will be introduced in the following.

In substitution period, the original image is firstly encoded into the DNA sequence matrix Bb. Next, DNA addition is carried out using mask matrix *M* and matrix Bb to generate matrix Temp. Finally Complement operation is performed according to the DNA complementary rule as suggested in [24]. The result is represented as *Bcomp*. Matrix *M* and *Z* are generated respectively by one logistic chaotic sequence as given in the following (1).

$$X_{n+1} = rX_n(1 - X_n) \tag{1}$$

The values of parameter r are in the range 3.057–4. Two initial values x_1 and y_1 are chosen to compute x_0 and y_0 as follows

$$x_0 = \left(\frac{\mathrm{mod}\left(\sum_{i=1}^{m/2} P(i,j), 256 \right)}{256} + x_1 \right) \mathrm{mod}\, 1$$

$$y_0 = \left(\frac{\mathrm{mod}\left(\sum_{i=m/2+1}^{m} P(i,j), 256 \right)}{256} + y_1 \right) \mathrm{mod}\, 1 \tag{2}$$

Two different values of r (r_3 and r_4) are used to generate *M* and *Z*.

In diffusion period *Bcomp* is divided into equal size blocks and these blocks are shuffled based on two chaotic sequences through 2D logistic map as shown in the right part of Fig. 1. 2D logistic map is given in the following (2).

$$\begin{aligned} X_{n+1} &= r_1 X_n (1 - X_n) + s_1 Y_n^2 \\ Y_{n+1} &= r_2 Y_n (1 - Y_n) + s_2 \left(X_n^2 + X_n Y_n \right) \end{aligned} \tag{3}$$

where $2.75 < r_1 < 3.4$, $2.75 < r_2 < 3.45$, $0.15 < s_1 < 0.21$ and $0.13 < s_2 < 0.15$. The parameters $(x_1, x_{2,}\ r_1, r_2, r_3, r_4)$ work as secret key.

3. Chosen-plaintext attack

In this section, we will decrypt the image **I** with 256 × 256 resolution at first and afterwards extend the scheme to the image with other resolution. The corresponding cipher image of **I** is denoted as C_I. Denote the sum of the front half pixels of **I** by μ and that of the latter half pixels of the original by γ, which are expressed as

$$\mu = \bmod\left(\sum_{i=1}^{m/2} P(i,j), 256\right)$$

$$\gamma = \bmod\left(\sum_{i=m/2+1}^{m} P(i,j), 256\right).$$

The parameters $(x_1, y_{1,}\ r_1, r_2, r_3, r_4)$ work as secret key for the image encryption method. When $(x_1, y_{1,}\ r_1, r_2, r_3, r_4)$ is fixed, the encryption process will be different with μ and γ changing. According to Kerckhoffs' principle, the front half and latter half pixel sum is known. We will attack only according to specific μ and γ.

Choose a plain image **P**. The chosen plain image **P** is expected to have the same μ and γ as the original image. We construct the plain image **P** as the following. Divide **P** into 256 equal size blocks of size 16 × 16. Set the first block $P\left\{1,1\right\}$ as

$$P\left\{1,1\right\} = \begin{bmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{16\times16}$$

and the 256th block as

$$P\left\{16,16\right\} = \begin{bmatrix} 255 & 255 & \cdots & 255 \\ 255 & 255 & \cdots & 255 \\ \cdots & \cdots & \cdots & \cdots \\ 255 & 255 & \cdots & \alpha_{256} \end{bmatrix}_{16\times16}.$$

Set the other blocks as

$$P\{m,n\} = \begin{bmatrix} l & l & \cdots & l \\ l & l & \cdots & l \\ \cdots & \cdots & \cdots & \cdots \\ l & l & \cdots & l \end{bmatrix}_{16\times16}$$

where $l=(m-1)*16+n-1$.

Choose α_1 and α_{256} for

$$\bmod\left(\sum_{i=1}^{m/2} P(i,j), 256\right) = \mu$$

$$\bmod\left(\sum_{i=m/2+1}^{m} P(i,j), 256\right) = \gamma.$$

The corresponding cipher image of **P** is denoted as **C**.

The corresponding locations in cipher image of one pixel respectively in the front half pixels and the latter half of the original image need to be revealed at first.

3.1. The locations in cipher of two pixels

Without loss of generality, We choose revealing the corresponding locations of $P(1,1)$ and $P(256,\ 256)$. The process is computed as the following steps.

Step 1: Choose an image **P**′ with the same pixels as **P** except the locations of (1,1) and (1,2). Set $P^{'}(1,1)$ as

$$P^{'}(1,1) = \bmod(P(1,1) + 1256).$$

To keep encryption process of **P’** consistent with that of **P**, i.e., to keep μ unchanged, set $P^{'}(1,2)$ as

$$P^{'}(1,2) = \mathrm{mod}\,(P(1,2) - 1256)$$

The corresponding cipher image of **P**′ is denoted as **C**′ and it will only have two different pixels with **C**.

Step 2: Choose an image **P**″ with the same pixels as **P** except the locations of (1,1) and (1,3). To keep μ unchanged, Set $\ddot{P}(1,1)$ and $\ddot{P}(1,3)$ as

$$\begin{cases} \mathrm{P}^{''e;e;}(1,1) = \mathrm{mod}\,(P(1,1) + 1256) \\ \mathrm{P}^{''e;e;}(1,3) = \mathrm{mod}\,(P(1,3) - 1256) \end{cases}.$$

The corresponding cipher image of **P**″ is denoted as **C**″ and it will only have two different pixels with **C**, too.

Step 3: Perform XOR operation between **C** and **C**′ to obtain Image **CX1** in which there should be two different pixels between **C** and **C**′. There should be two nonzero values in **CX1** The locations of two values are denoted by (a_1, b_1) and (a_2, b_2). Execute XOR operation between **C** and **C**″ to obtain Image **CX2**. Likewise, there would be two nonzeros values in **CX2** whose locations are denoted by (a_3, b_3) and (a_4, b_4).

Step 4: Compare the four locations and there would be sole location in (a_1, b_1) and (a_2, b_2) which is equal to (a_3, b_3) or (a_4, b_4). Suppose the sole location is (a_1, b_1) without loss of generality and the location (a_1, b_1) is the corresponding position in cipher image of the location $(1,\ 1)$ pixel in Image **P**. We can get the corresponding location in cipher image of the location $(256,\ 256)$ just in the same way and suppose it is the location (a_{256}, b_{256}).

3.2. Position and value decryption

The corresponding location and value of each pixel in the original image of the cipher image are computed as follows.

Step 1: Choose 256 images $\{\mathbf{P}_k\}_{k=0}^{255}$ and set them

$$\mathbf{P}_k = \begin{bmatrix} g_k & k & k & k \\ k & k & k & k \\ \vdots & \vdots & \vdots & \vdots \\ k & k & k & k \\ k & k & k & h_k \end{bmatrix},$$

where to keep the same μ and γ with that of **P**,

$$g_k = \mathrm{mod}\left(\mathrm{mod}\left(\sum_{i=1,j=2}^{256} P_k(i,j) + \sum_{i=2}^{m/2} P_k(i,j), 256\right) - \mu, 256\right)$$

$$h_k = \mathrm{mod}\left(\mathrm{mod}\left(\sum_{i=m,j=1}^{255} P_k(i,j) + \sum_{i=m/2+1}^{m-1} P_k(i,j), 256\right) - \gamma, 256\right).$$

The corresponding cipher images $\{\mathbf{P}_k\}_{k=0}^{255}$ are denoted by $\{\mathbf{C}_k\}_{k=0}^{255}$.

Step 2: Choose Pixel $C(i,j)$, $(i,j = 0,\ 1,\ \cdots,\ 255)$ in **C**, except $C(a_1, b_1)$ and $C(a_{256}, b_{256})$. Compare it with $C_k(i,j)$, k=0, 1, . . ., 255 respectively. There would be one pixel in $\left\{C_k(i,j),\ \ k = 0,\ \ 1,\ \ \cdots,\ \ 255\right\}$ that equals to $C(i,j)$. Suppose the pixel is $C_k(i,j)$ and the corresponding pixel of $C(i,j)$ in plain image is in Block $P\{r,s\}$ where

$$\begin{aligned} r &= \mathrm{floor}(k/16 + 1) \\ s &= \mathrm{mod}\,(k, 16) \end{aligned}.$$

Add the location label (i,j) into Set $INDEX\{r,s\}$. (a_1, b_1) is registered in Set$INDEX\{r_1, s_1\}$ and (a_{256}, b_{256}) in $INDEX\{r_2, s_2\}$ where

$$r_1 = \mathrm{floor}(a_1/16 + 1)$$

$$s_1 = \mathrm{floor}(b_1/16 + 1)$$

$$r_2 = \mathrm{floor}(a_{256}/16 + 1)$$

$$s_2 = \mathrm{floor}(b_{256}/16 + 1)$$

$INDEX\{r,s\}$ has 256 location labels.

Step 3: Choose plain image **Q** where

$$Q\left\{1,1\right\}=\begin{bmatrix}\beta_1 & 1 & \cdots & 15\\ 16 & 18 & \cdots & 31\\ \cdots & \cdots & \cdots & \cdots\\ 240 & 242 & \cdots & 255\end{bmatrix}_{16\times16}$$

$$Q\left\{16,16\right\}=\begin{bmatrix}0 & 1 & \cdots & 15\\ 16 & 18 & \cdots & 31\\ \cdots & \cdots & \cdots & \cdots\\ 240 & 242 & \cdots & \beta_{256}\end{bmatrix}_{16\times16}$$

$$Q\left\{r,s\right\}==\begin{bmatrix}0 & 1 & \cdots & 15\\ 16 & 18 & \cdots & 31\\ \cdots & \cdots & \cdots & \cdots\\ 240 & 242 & \cdots & 255\end{bmatrix}_{16\times16}.$$

The corresponding cipher image of **Q** is denoted by $\mathbf{C}_q$. Choose $C_q(i,j)$ whose corresponding pixel in **Q** is in Set *INDEX* $\{r,s\}$. Compare $C_q(i,j)$, except $C(a_1,b_1)$ and $C(a_{256},b_{256})$, with $C_k(i,j)$, $k=0,1,\ldots,255$ respectively. There would be one pixel in $\left\{C_k(i,j),\quad k=0,\quad 1,\quad \cdots,\quad 255\right\}$ that equals to $C_q(i,j)$. Suppose the pixel is $C_k(i,j)$ and the corresponding pixel of $C_q(i,j)$ in plain image **Q** is the one that is equal to k. Likewise, we can get the corresponding location in plain image of any pixel in $\mathbf{C}_q$. The original image **I** has the same encryption process with **P** and **Q**. So we have known the corresponding location in **I** of each pixel in $\mathbf{C}_I$.

Step 4: Compare $C_I(i,j)$, except $C_I(a_1,b_1)$ and $C_I(a_{256},b_{256})$, with $C_k(i,j)$, $k=0,1,\ldots,255$. There would be sole pixel which is equal to $C_I(i,j)$ and suppose the pixel is $C_k(i,j)$, i.e., the corresponding pixel value in **I** is k. So we can know the corresponding value in **I** of each pixel in $\mathbf{C}_I$, except $C_I(a_1,b_1)$ and $C_I(a_{256},b_{256})$.

Step 5: Choose 256 images $\{\mathbf{W}_k\}_{k=0}^{255}$ and set them

$$\mathbf{W}_k=\begin{bmatrix}k & k & k & k\\ g_k^{'} & k & k & k\\ \vdots & \vdots & \vdots & \vdots\\ k & k & k & h_k^{'}\\ k & k & k & k\end{bmatrix},$$

where

$$g_k^{'}=\mathrm{mod}\left(\mathrm{mod}\left(\sum_{i=1,j=2}^{256}W_k(i,j)+\sum_{i=2}^{m/2}W_k(i,j),256\right)-\mu,256\right)$$

$$h_k^{'}=\mathrm{mod}\left(\mathrm{mod}\left(\sum_{i=m,j=1}^{255}W_k(i,j)+\sum_{i=m/2+1}^{m-1}W_k(i,j),256\right)-\gamma,256\right).$$

Compare $C_I(a_1,b_1)$ and with $C_k(a_1,b_1)$, $k=0,1,\ldots,255$ and there would be one pixel value that is equal to $C_I(a_1,b_1)$. Likewise, we can get the corresponding value of $C_I(a_{256},b_{256})$ in **I**. Now we can recover the original image according to the above steps.

3.3. Extension

In this part we extend the above cryptanalysis to those images with higher resolution than 256 × 256. Choose an original image **J** with resolution $N\times N$ where $N>256$ and denote the corresponding cipher image $\mathbf{C}_J$. Cryptanalysis process is as follows.

Step 1 Choose a plain image and denote the corresponding cipher image as $\mathbf{C}_{\mathbf{Q_1}}$. Divide into 256 blocks of equal size and call them 1st order sub-blocks. Assume the size of a 1st order sub-block is more than 256. Denote 1st order sub-blocks by *INDEX* $\left\{r^1,s^1\right\}\quad r^1,s^1=1,2,\cdots,16$ and its size by $B_{1.}$ Set them as **P**. Reveal the locations of two pixels by executing the steps of 3.1.

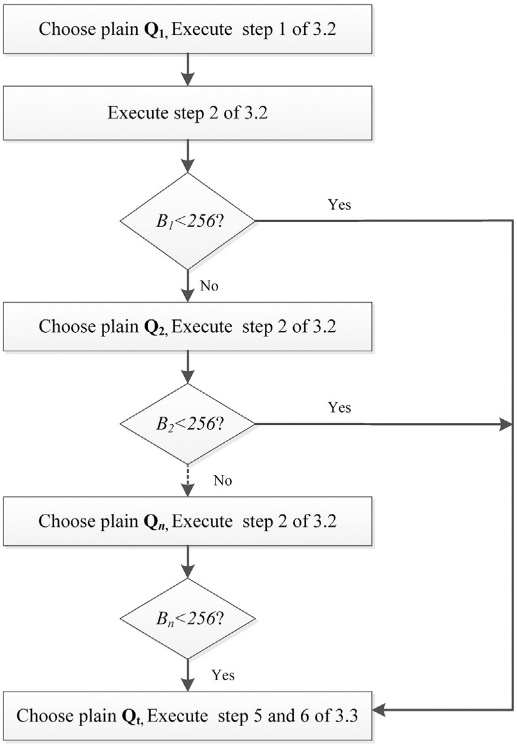
Choose plain $\mathbf{Q_1}$, Execute step 1 of 3.2
Execute step 2 of 3.2
B_1<256?
Yes
No
Choose plain $\mathbf{Q_2}$, Execute step 2 of 3.2
B_2<256?
Yes
No
Choose plain $\mathbf{Q}_n$, Execute step 2 of 3.2
B_n<256?
Yes
Choose plain $\mathbf{Q_t}$, Execute step 5 and 6 of 3.3

Fig. 2. Flowchart of cryptanalysis.

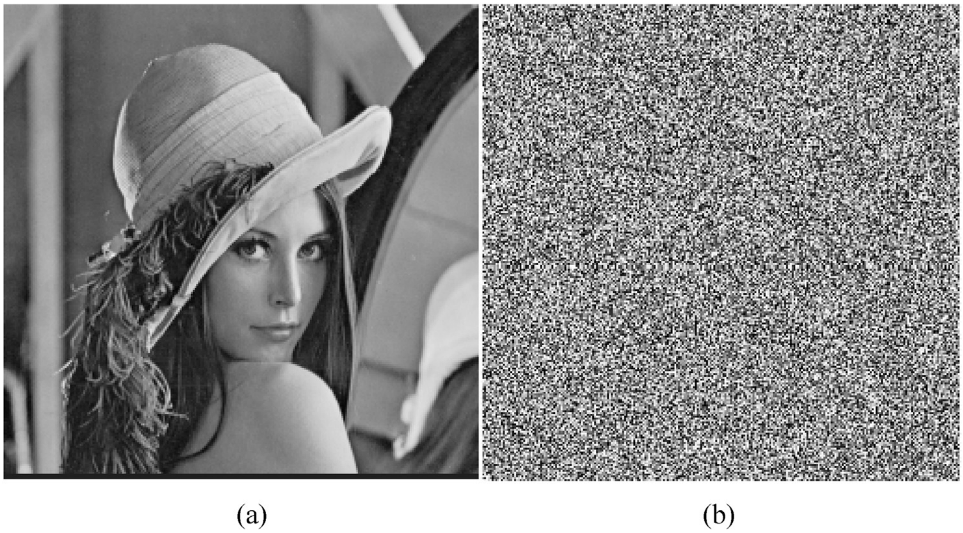

Fig. 3. (a) Image of Lena; (b) Cipher image of Lena.

Step 2 Execute step 2 of 3.2 and reveal the corresponding 1st order sub-block in $\mathbf{Q}_1$ of each pixel in $\mathbf{C}_{\mathbf{Q}_1}$.

Step 3 Choose a plain image $\mathbf{Q}_2$ and denote the corresponding cipher image as $\mathbf{C}_{\mathbf{Q}_2}$. Divide $\mathbf{Q}_2$ into 256 1st sub-blocks and divide each 1st order sub-block into 256 blocks of equal size which are called 2nd sub-blocks. Denote a 2nd sub-block by $INDEX\left\{r^1, s^1\right\}\left\{r^2, s^2\right\}\quad r^2, s^2 = 1, 2, \cdots, 16$and its size by $B_{2.}$ Assign 256 different integers in [0255] to $INDEX\left\{r^1, s^1\right\}\left\{r^2, s^2\right\}$. Execute step 2 of 3.2 and reveal the corresponding 2nd order sub-block in $\mathbf{Q}_2$ of each pixel in $\mathbf{C}_{\mathbf{Q}_2}$.

Step 4 Repeat choosing plain images sub-block division as Step 3 until the pixel number of each sub-block is less than 256. Denote its size by B_n Assume the pixel number of n$^{\text{th}}$ order $INDEX\left\{r^1, s^1\right\}\left\{r^2, s^2\right\}\cdots\left\{r^n, s^n\right\}$ is less than 256 and

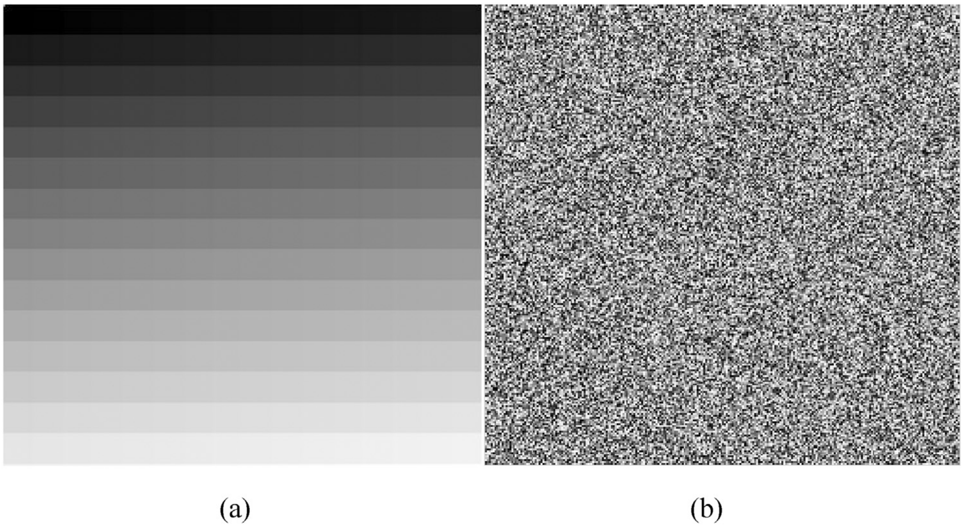

Fig. 4. (a) Chosen plain image **P**; (b) **C**, Cipher image of **P**.

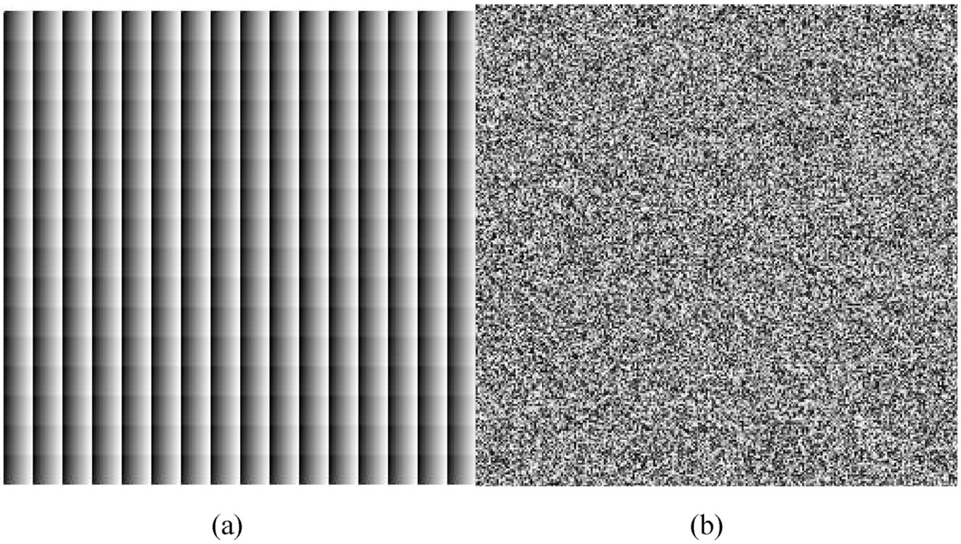

Fig. 5. (a) Chosen plain image **Q**; (b) $\mathbf{C}_q$, Cipher image of **Q**.

denote the number by u. Call the plain image $\mathbf{Q}_n$. Take the image with 1024 × 1024 resolution as example, it need be divided two times and the size of 1nd sub-block is 2 × 2.

Step 5 Choose plain image $\mathbf{Q}_t$ and assign u different integers to each nth order *INDEX* $\left\{r^1, s^1\right\}\left\{r^2, s^2\right\}\cdots\left\{r^n, s^n\right\}$. Denote the corresponding cipher image by $\mathbf{C}_{\mathbf{Q}_2}$. Execute step 3 of 3.2 and reveal the corresponding location in $\mathbf{Q}_t$ of every pixel in $\mathbf{C}_{\mathbf{Q}_t}$.

Step 6 Execute step 4 and step 5 of 3.2 we can reveal the corresponding value in I of each pixel in $\mathbf{C}_{\mathbf{Q}_t}$. We can recover the original image according to the corresponding locations and values. The flowchart of cryptanalysis is shown in Fig. 2.

Take the image with 1024 × 1024 resolution as example; it can be decrypted by 2nd sub-block division.

4. Experiments and analysis

To verify the effectiveness of the aforementioned cryptanalyses, some experiments have been carried out on the grayscale image 'Lena' as shown in Fig. 3(a). First, it is encrypted by the scheme described in [24]. The corresponding cipher image is shown in Fig. 3(b). Fig. 4(a) and (b) is the chosen plain **P** and the corresponding cipher image **C**. Fig. 5(a) and (b) is the chosen plain **Q** and the corresponding cipher image $\mathbf{C}_q$. The recovered image is obtained by applying the cryptanalyses described in Section 3 and is shown in Fig. 6. Obviously it is the same as the original plain images. Therefore the attack results demonstrate the effectiveness of the cryptanalyses.

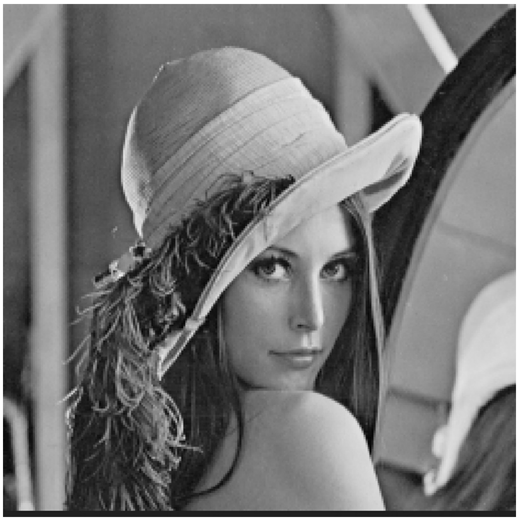

Fig. 6. Recovered image from Fig. 3(b).

5. Conclusions

This paper attacks a robust image encryption algorithm which is recently proposed in [24]. The encryption process depends on DNA algorithm, 1D and 2D logistic chaotic maps. The security weakness of this scheme is analyzed and the algorithm is completely cracked by the chosen plaintext image attack scheme in Section 3. Experimental results verify the proposed chosen plaintext attack scheme.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant no. 61602158), China Postdoctoral Science Foundation (Grant no. 2016M600030), Science Foundation for the Excellent Youth Scholars of Henan Normal University (Grant No. YQ201607), Science Foundation for Young Scholars of Henan Normal University (Grant No. 2014QK20), and PhD Scientific Research Foundation of Henan Normal University (Grant no. qd14134).

References

[1] Y.S. Zhang, D. Xiao, Y.L. Shu, J. Li, A novel image encryption scheme based on a linearhyperbolic chaotic sys-tem of partial differential equations, Signal Process.-Image Commun. 28 (3) (2013) 292–300.

[2] Y.S. Zhang, D. Xiao, Double optical image encryption using discrete Chirikov standard map and chaos-based fractional random transform, Opt. Lasers Eng. 51 (4) (2013) 472–480.

[3] G.D. Ye, K.W. Wong, An efficient chaotic image encryption algorithm based on a generalized Arnold map, Nonlinear Dyn. 69 (4) (2012) 2079–2087.

[4] O. Mirzaei, M. Yaghoobi, H. Irani, A new image encryption method: parallel sub-image encryption with hyperchaos, Nonlinear Dyn. 67 (1) (2011) 557–566.

[5] Y.B. Liu, S.M. Tian, W.P. Hu, C.C. Xing, Design and statistical analysis of a new chaotic block cipher for wireless sensor networks, Commun. Nonlinear Sci. Numer. Simul. 17 (8) (2012) 3267–3278 (32).

[6] K.W. Wong, B. Kwok, W. Law, A fast image encryption scheme based on chaotic standard map, Phys. Lett. A 372 (15) (2008) 2645–2652.

[7] D. Xiao, X.F. Liao, P.C. Wei, Analysis and improvement of a chaos-based image encryption algorithm, Chaos Solitons Fract. 40 (15) (2009) 2191–2199.

[8] Y.S. Zhang, D. Xiao, W.Y. Wen, Y. Tian, Edge-based lightweight image encryption using chaos-based reversible hidden transform and multiple-order discrete fractional cosine transform, Opt. Laser Technol. 54 (30) (2013) 1–6.

[9] M. Babaei, A novel text and image encryption method based on chaos theory and DNA computing, Nat. Comput. 12 (1) (2013) 101–107.

[10] X.L. Huang, G.D. Ye, An image encryption algorithm based on hyper-chaos and DNA sequence, Multimed. Tools Appl. 72 (1) (2014) 57–70.

[11] L.L. Liu, Q. Zhang, X.P. Wei, A RGB image encryption algorithm based on DNA encoding and chaotic map, Comput. Electr. Eng. 38 (5) (2012) 1240–1248.

[12] H.J. Liu, X.Y. Wang, A. Kadir, Image encryption using DNA complementary rule and chaotic maps, Appl. Soft Comput. 12 (5) (2012) 1457–1466.

[13] X.P. Wei, L. Guo, Q. Zhang, J.X. Zhang, S.G. Lian, A novel color image encryption based on DNA sequence operation and hyper-chaotic system, J. Syst. Softw. 85 (2) (2012) 290–299.

[14] Q. Zhang, L. Guo, X.P. Wei, Image encryption using DNA addition combining with chaotic maps, Math. Comput. Model. 52 (11–12) (2010) 2028–2035.

[15] Q. Zhang, L. Guo, X.P. Wei, A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system, Optik 124 (18) (2013) 3596–3600.

[16] Q. Zhang, L.L. Liu, X.P. Wei, Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps, AEU-Int. J. Electron. Commun. 68 (3) (2014) 186–192.

[17] Q. Zhang, X.P. Wei, A novel couple images encryption algorithm based on DNA subsequence operation and chaotic system, Optik 124 (23) (2013) 6276–6281.

[18] Y.S. Zhang, D. Xiao, Cryptanalysis of S-box-only chaotic image ciphers against chosen plaintext attack, Nonlinear Dyn. 72 (4) (2013) 751–756.

[19] C.Q. Li, L.Y. Zhang, R. Ou, K.W. Wong, S. Shu, Breaking a novel colour image encryption algorithm based on chaos, Nonlinear Dyn. 70 (4) (2012) 2383–2388.

[20] Y. Zhang, C.Q. Li, Q. Li, D. Zhang, S. Shu, Breaking a chaotic image encryption algorithm based on perceptron model, Nonlinear Dyn. 69 (3) (2012) 1091–1096.
[21] S.J. Li, C.Q. Li, G.R. Chen, N.G. Bourbakis, K.T. Lo, A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, Signal Process.-Image Commun. 23 (3) (2008) 212–223.
[22] C.Q. Li, K.T. Lo, Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, Signal Process. 91 (4) (2011) 949–954.
[23] B. Norouzi, S. Mirzakuchaki, S.M. Seyedzadeh, M.R. Mosavi, A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion, Multimed. Tools Appl. 71 (3) (2014) 1469–1497.
[24] Anchal Jain, Navin Rajpal, A robust image encrytion algorithm resistant to attacks using DNA and chaotic logistic maps, Mutitimed. Tools Appl. 75 (2016) 5455–5472.