Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **4**, July **2018**

Security Analysis and Improvement of An Image Encryption Scheme Based on Chaotic Tent Map

Tsu-Yang Wu

Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technolog
33 Xuefunan Road, University Town, Fuzhou 350118, China
National Demonstration Center for Experimental Electronic Information and Electrical Technology Education
Fujian University of Technolog
33 Xuefunan Road, University Town, Fuzhou 350118, China
wutsuyang@gmail.com

Xiaoning Fan

Department of Electronic Engineering
School of Computer Science and Engineering
Harbin Institute of Technology Shenzhen Graduate School
HIT Campus of University Town of Shenzhen, Shenzhen 518055, China
1203119830@qq.com

King-Hang Wang

Department of Computer Science and Engineering
Hong Kong University of Science and Technology
The Hong Kong University of Science and Technology Clear Water Bay, Kowloon, Hong Kong
kevinw@cse.ust.hk

Jeng-Shyang Pan

Fujian Provincial Key Laboratory of Big Data Mining and Applications
Fujian University of Technolog
33 Xuefunan Road, University Town, Fuzhou 350118, China
National Demonstration Center for Experimental Electronic Information and Electrical Technology Education
Fujian University of Technolog
33 Xuefunan Road, University Town, Fuzhou 350118, China
jspan@cc.kuas.edu.tw

Chien-Ming Chen

Department of Electronic Engineering
School of Computer Science and Engineering
Harbin Institute of Technology Shenzhen Graduate School
HIT Campus of University Town of Shenzhen, Shenzhen 518055, China
chienming.taiwan@gmail.com

Jimmy Ming-Tai Wu*

Department of Electronic Engineering
College of Computer Science and Engineering
Shandong University of Science and Technology
579 Qianwangang Road, Huangdao District, Qingdao 266590, China
*Corresponding author: wmt@wmt35.idv.tw

Received May 2018; revised July 2018

Abstract. *In this paper, we make cryptanalysis on an image encryption scheme based on chaotic Tent map proposed by Li et al.* [5]. *We find chosen-plaintext attack can break the scheme. And we successfully carry out the chosen-plaintext attack. Last, we make an improvement method for overcome the drawback.*

Keywords: Image Encryption, Chaotic Map, Security, Chosen Plaintext Attack

1. Introduction.

In 1989, Matthews [7] proposed the application of chaotic systems to image encryption for the first time. He proposed a stream cipher based on deformed Logistic maps. In 1998, Fridrich [1] proposed a symmetric block encryption algorithm. The main idea of the encryption algorithm is to use Baker map to scramble the pixel value location of plaintext image . In 2000, Yen and Guo [3] proposed a new key-based chaotic image encryption algorithm (CKBA), which changed the pixel value of the image. However, in 2002, Li and Zheng [6] pointed out that CKBA can not resist the choice plaintext attack. To enhance the security of CKBA, many scholars began to combine confusion and diffusion encryption. As we all know, plaintext image pixel position after scrambling, the correlation between adjacent pixels will be greatly reduced. After the replacement of the pixel value can successfully resist the statistical attacks. Therefore, a secure image encryption algorithm should include both confusion and diffusion [9, 8]. In 2008, Gao and Chen [2] proposed an image encryption algorithm based on hyper-chaotic map. In 2012, Hussain et al. [4] proposed a color image encryption algorithm that combines a S-box and NCA map. However, Zhang and Xiao [11] proved that the algorithm does not resist the choice plaintext attack. In 2016, Ye and Huang [10] proposed an image encryption algorithm based on both bit-level and pixel-level. The remainder of this paper is organized as follows. In Section 2 we present a review of Li et al.s image encryption algorithm. Then we present security flaws of Li et al.s algorithm under a chosen plaintext attack and our improvement in Section 3 and 4 respectively. Some experiment results and analyses are given in Section 5 and a conclusion is given in Section 6.

2. Review of Li et al.'s image encryption algorithm.

2.1. The chaotic Tent map.

The chaotic Tent map is shown as below:

$$x_{i+1} = f(x_i, \mu) \tag{1}$$

$$f(x_i,\mu) = \begin{cases} f_L(x_i,\mu) = \mu x_i, & \text{if } x_i < 0.5 \\ f_R(x_i,\mu) = \mu(1-x_i), & \text{otherwise;} \end{cases} \tag{2}$$

where $x_i \in [0,1]$, $i = 0,1,2,\cdots$, μ is the control parameter of Tent map. When $\mu \in [0,2]$, the Tent map has chaotic behaviors.

2.2. Details of the Li et al.'s scheme.

The image cryptosystem is shown in FIGURE 1. The encryption process is as follows:

1. Read original image $P(a \times b)$, use $[a,b,c]$ to save the size of P, e.g., $a = 256, b = 256, c = 3,$ let $N = a \times b \times c.$ and initiate control parameter μ of the chaotic Tent map;
2. Use the secret key (x_0, μ) and iterate the chaotic Tent map N times using system (2), and obtain the key array $x(n)$, where N is the size of $x(n)$, we can get the key array $X(n)$ by Eq. (3) and the range of elements is 0-255;

$$X(i) = \mathrm{mod}\left(floor\left(x(i) \times 10^{14}\right), 256\right) \;\; i = 1,2,\cdots N \tag{3}$$

where $floor(x)$ means the nearest integer to x toward minus infinity.

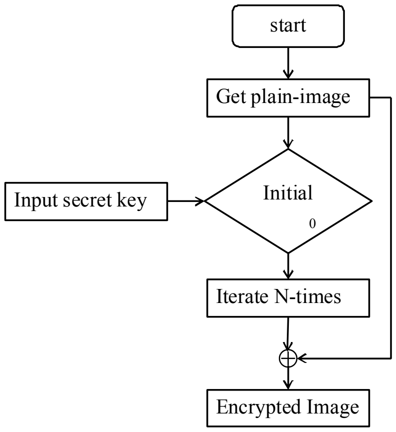


Figure 1. The flowchart of Li et al.'s algorithm.

3. Encrypt each element of matrix P by Eq.(4) and get the cipher image C.

$$C = P \oplus X \tag{4}$$

3. **Cryptanalysis on Li et al.'s algorithm.** Assume $(t \oplus x) \oplus (p \oplus x) = v$, the below algorithm will find the value of p and p is regard as the pixel of plain image, where x is a constant and unknown, x can be regard as the generated random sequence of Tent map. t and v are known. t, x, p and v are all between 0 and 255. It is described in Algorithm 1.

Algorithm 1 The process of recover one pixel

```
 1: set $t = 0$;
 2: $(t \oplus x) \oplus (p \oplus x) = v$;
 3: while do($j \neq 8$)
 4:     if then($v = 0$)
 5:         $t = p$;
 6:         $j = 8$;
 7:     else
 8:         $t = \mathrm{mod}(t + 2^j), 256$;
 9:         go to line 2;
10:     end if
11: end while
```

According to the properties of exclusive OR operation, the low j bits are same between p and t after Step 2 in Algorithm 1. Updating t in line 3 will cause one more bit equal in line 2. When v is zero, t is equal to p. Because all the values are between 0 and 255, we can find the value of p less than 8 times the Algorithm 1.

We can recovered the plain image by Algorithm 2.

Algorithm 2 The process of recover the plain image

```
1: $j = 1$, and pixels of image $PX$ are reset all zeros;
2: encrypt $PX$ by the proposed algorithm, get the cipher image $CX$;
3: calculate $P_j$ by Algorithm 1;
4: go to line 2 until $j = N$;
```

From above analysis, we can see that the number of chosen-images needed is less than $8N$, if the encryption complexity is T, then the attack complexity is $T \times O(N) \approx T \times 4N$.

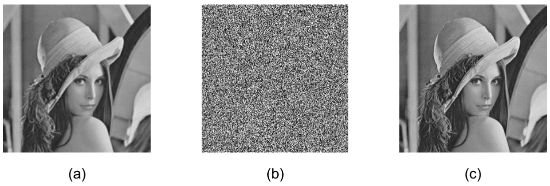


Figure 2. The experimental result of chosen plaintext attack. (*a*) The plain image "Lena"; (*b*) The corresponding cipher image; (*c*) Completely recovered image.

We perform the experiment using C++ programming language and on a PC with Intel(R) Core(TM) i5-3470 CPU 3.20GHz, 4.00 GB memory, and Windows 7 OS. We encrypt an image Lena of size 256 × 256 with the proposed algorithm, and then break it with our chosen plaintext attack described above. The total time of breaking the algorithm is 1. 48389h. Experimental results are shown in FIGURE 2.

4. **Our improvement and simulation results.** Analysis the original algorithm, we can find that it only diffused the pixel value of the plain image and not scramble the positions of pixels. So its security is low. Meanwhile, the secret keys are low sensitivity to the changes of plain image. For these reasons, we can improve the proposed algorithm avoiding these flaws. The specific improvements are described as below:

4.1. **Permutation phase.** Step 1: As an initial value of Tent map, x_0, y_0 and z_0 are updated by Eq. (5).

$$\begin{aligned} y_0^{'} &= \bmod\left(y_0 + \tfrac{\bmod(sum, a\times b\times c)}{a\times b\times c}, 1\right) \\ z_0^{'} &= \bmod\left(z_0 + \tfrac{\bmod(sum, a\times b\times c)}{a\times b\times c}, 1\right) \end{aligned} \tag{5}$$

where sum is the sum of the plain image, a, b and c is the size of the plain image. Step 2: Under the initial values and control parameters $(y_0^{'}, \mu_1)$, iterating the Tent map m times and get the pseudo-random sequence $r(a)$, obtain $R(a)$ by Eq. (6).

$$R(i) = \bmod\left(floor\left(z(i) \times 10^{14}\right), b\right), i = 1, 2, \cdots a \tag{6}$$

Step 3: Under the initial values and control parameters $(z_0^{'}, \mu_2)$, iterating the Tent map n times and get the pseudo-random sequence $z(b)$, obtain $Z(b)$ by Eq. (7).

$$Z(i) = \bmod\left(floor\left(z(i) \times 10^{14}\right), a\right), i = 1, 2, \cdots b \tag{7}$$

Step 4: Scramble the rows and columns of the plain image $P(a, b, c)$.
Row permutation: Perform right units circular shifted on the i-th row sequence $R(i)$ times;
Column permutation: Perform up units circular shifted on the i-th column sequence $Z(i)$ times;
So the shuffled image P^* is generated.

4.2. **Diffusion phase.** Step 5: Use the secret key (x_0, μ) and iterate the chaotic Tent map N times using system (2), and obtain the key array $x(n)$, where N is the size of $x(n)$, we can get the key array $X(n)$ by Eq. (8) and the range of elements is 0-255;

$$X(i) = \bmod\left(floor\left(x(i) \times 10^{14}\right), 256\right), i = 1, 2, \cdots N \tag{8}$$

Step 6: Transform P^* to one dimension from above to below and from left to right. The cipher image C is generated by Eq. (9).

$$C(i) = \bmod\left(P^*(i) + C(i-1), 256\right) \oplus X(i), i = 1, 2, \ldots N \tag{9}$$

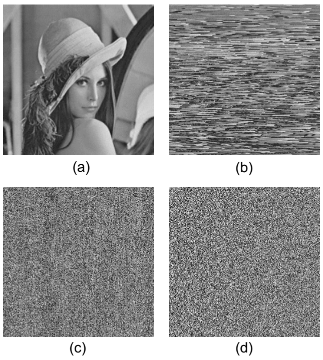


Figure 3. The encryption result of improved algorithm. (*a*) The plain image "Lena"; (*b*) The image after rows permutation; (*c*) The image after rows and columns permutation; (*d*) The cipher image.

where $C(0)$ is a part of initial keys.

The decryption process is the reverse process of encryption algorithm.

5. **Experimental results.** Experimental analysis of the improved image encryption scheme in this paper has been done. The plain image with size 256 × 256 is shown in FIGURE 3(a) and the image after rows permutation is shown in FIGURE 3(b), FIGURE 3(c) show the image after columns permutation, FIGURE 3(d) is the cipher image after diffusion phase.

5.1. **Key space analysis.** Compared with the original algorithm, the improved scheme has larger key space. The original keys are (x_0, μ), the keys of improved algorithm are $(x_0, \mu, y_0^{'}, \mu_1, z_0^{'}, \mu_2, C(0))$. If the precision is 10^{-14}, the key space is 256×10^{84}. So the key space is large enough to resist brute-force attacks.

5.2. **Key's sensitivity to plaintext.** The improved scheme can overcome the low sensitivity problem. Because the initial keys are related to the plain image. From the algorithm, we can know that the initial keys are different when encrypt different images. Then the improved algorithm can resist the chosen-plaintext attacks.

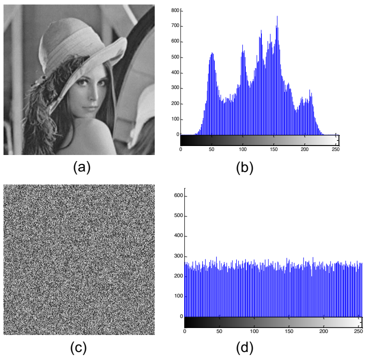
5.3. **Histogram analysis.** The distribution of the ciphered image is a major concern. A histogram shows the distribution of pixel values of an image. If it is flat, some image information can not be find by the statistical attacks. Therefore, a flat distribution is desirable. The histograms are shown in FIGURE 4.

5.4. **Correlation analysis.** In image processing, correlation analysis is often used to test the correlation between two adjacent pixels. We know that the correlation is high between two adjacent pixels in the plain image. A good image encryption algorithm should be able to destroy this high correlation to resist statistic attacks.

2000 pairs of adjacent pixels are selected randomly in vertical, horizontal and diagonal directions from the plain images and their cipher images, respectively. The correlation coefficients $r_{x,y}$ are calculated among pixels as follows:

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{10}$$

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{11}$$



(a) (b)

(c) (d)

Figure 4. The histograms of the plain image and cipher image. (*a*) The plain image “Lena”; (*b*) Distribution of plain image “Lena”; (*c*) The cipher image of “Lena”; (*d*) Distribution of cipher image.

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{12}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \tag{13}$$

where x ,y are the gray values of two adjacent pixels in the image and N is the total number of pixel pairs elected from the image. The lower the correlation of the adjacent pixels of the cipher image, the better the encryption algorithm performs. TABLE 1 shows the correlation coefficient comparison of original scheme and the improved scheme.

Table 1. The correlation coefficient comparison of original scheme and the improved scheme.

Algorithm	**Horizontal**	**Vertical**	**Diagonal**
Original Algorithm	0.0126907	0.01575	-0.029093
Improved Algorithm	-0.27277	-0.065545	-0.023571

5.5. **Information entropy.** Information entropy is an important feature of randomness. Let m be the information source, and the formula for calculating information entropy is:

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i)\log_2\frac{1}{p(m_i)} \tag{14}$$

Suppose information source has 2^8 states:

$$\text{the ideal } H(m) = \sum_{i=0}^{255} \frac{1}{2^8}\log_2 2^8 = 8 \tag{15}$$

where $p(m_i)$ represents the probability of symbol m. For a cipher image with 256 gray levels, the entropy should ideally be 8. TABLE 2 shows the comparison of entropy, which are all very close to ideal value 8, however, our improved algorithm is secure.

Table 2. The comparison of Information entropy of the original algorithm and the improved algorithm.

Algorithm	**Information entropy of plain image**
Original Algorithm	7.9972
Improved Algorithm	7.99705

5.6. **Differential attack.** If the cipher images have strong difference with their corresponding plain images having only a little difference, we say that the encryption algorithm can resist the differential attack. The NPCR(Number of Pixels Change Rate) and UACI(Unified Average Changing Intensity) are two parameters to the sensitivity of an encryption algorithm. NPCR represents the percentage of the different pixel values of two cipher images. UACI represents the average difference between pixel values of two cipher images. NPCR and UACI can be calculated as Eq. (16) and Eq. (18).

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\% \tag{16}$$

$$D(i,j) = \begin{cases} 0, & C_1(i,j) = C_2(i,j) \\ 1, & C_1(i,j) \neq C_2(i,j) \end{cases} \tag{17}$$

$$\mathrm{UACI} = \left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255 \times M \times N} \right] \times 100\% \tag{18}$$

where C_1 and C_2 represent pixel values of two cipher images with size $M \times N$. The two plain images corresponding to a C_1 and C_2 have only one pixel value different. TABLE 3 shows the comparison of NPCR and UACI of the original algorithm and the improved algorithm. In the improved algorithm, the NPCR is over 99% and the UACI is over 33%, which indicate that the improved algorithm is very sensitive with a little change in the plain image. Therefore, the proposed algorithm is secure to resist differential attacks. However, the original algorithm is low sensitivity to the change of plain image, which has a low secure to resist differential attacks.

Table 3. The comparison of NPCR and UACI of the original algorithm and the improved algorithm.

Algorithm	**NPCR**	**UACI**
Original Algorithm	0.010634	0.01668099
Improved Algorithm	0.995621	0.33477

6. **Conclusions.** In this paper we present the vulnerability of Li et al.'s algorithm and present our improvement with significant performance. The main reason that allows us to break their algorithm is the fact that it only has a diffusion process and its very simple. Our improvement combines confusion and diffusion. Moreover, the improvement has a good encryption result.

Acknowledgment. The work of Tsu-Yang Wu was supported in part by the Science and Technology Development Center, Ministry of Education, China under Grant no. 2017A13025 and the Natural Science Foundation of Fujian Province under Grant no. 2018J01636.

REFERENCES

[1] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, *International Journal of Bifurcation and Chaos*, vol. 8, no. 06, pp. 1259-1284, 1998.

[2] T. Gao and Z. Chen, A new image encryption algorithm based on hyper-chaos, *Physics Letters A*, vol. 372, no. 4, pp. 394-400, 2008.

[3] J.-I. Guo, A new chaotic key-based design for image encryption and decryption, *In The IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 49-52, 2000.

[4] I. Hussain, T. Shah, and M. A. Gondal, An efficient image encryption algorithm based on S8 S-box transformation and NCA map, *Optics Communications*, vol. 285, no. 24, pp. 4887-4890, 2012.

[5] C. Li, G. Luo, K. Qin, and C. Li, An image encryption scheme based on chaotic tent map, *Nonlinear Dynamics*, vol. 87, no. 1, pp. 127-133, 2017.

[6] SJ Li and X. Zheng, Cryptanalysis of a chaotic image encryption method, *In IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 708-711, 2002.

[7] R. Matthews, On the derivation of a “chaotic” encryption algorithm, *Cryptologia*, vo. 13, no. 1, pp. 29-42, 1989.

[8] X.-Y. Wang, F. Chen, and T. Wang, A new compound mode of confusion and diffusion for block encryption of image based on chaos, *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2479-2485, 2010.

[9] K.-W. Wong, Bernie S.-H. Kwok, and W.-S. Law, A fast image encryption scheme based on chaotic standard map, *Physics Letters A*, vol. 372, no. 15, pp. 2645-2652, 2008.

[10] G. Ye and X. Huang, A feedback chaotic image encryption scheme based on both bit-level and pixel-level, *Journal of Vibration and Control*, vol. 22, no. 5, pp. 1171-1180, 2016.

[11] Y. Zhang and D. Xiao, Cryptanalysis of S-box-only chaotic image ciphers against chosen plaintext attack, *Nonlinear Dynamics*, vol. 72, no. 4, pp. 751-756, 2013.