A one-time one-key encryption algorithm based on the ergodicity of chaos*

Wang Xing-Yuan(王兴元)† and Teng Lin(滕 琳)‡

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, *China*

(Received 4 July 2011; revised manuscript received 11 August 2011)

In this paper, we propose a new one-time one-key encryption algorithm based on the ergodicity of a skew tent chaotic map. We divide the chaotic trajectory into sub-intervals and map them to integers, and use this scheme to encrypt plaintext and obtain ciphertext. In this algorithm, the plaintext information in the key is used, so different plaintexts or different total numbers of plaintext letters will encrypt different ciphertexts. Simulation results show that the performance and the security of the proposed encryption algorithm can encrypt plaintext effectively and resist various typical attacks.

Keywords: chaos, encryption, ergodicity

PACS: 05.45.–a, 05.45.Gg **DOI:** 10.1088/1674-1056/21/2/020504

1. Introduction

Chaos has become one of the most important research fields in the past decade. It was widely researched in secure communication,[1,2] chaotic synchronization,[3−5] and cryptography.[6,7] Chaos is a random-like process that is present in a deterministic system. A chaotic system has two main characteristics: a high sensitivity to initial value and an unpredictability of system change. These two characteristics are the important characteristics of a cipher random sequence. So, the chaotic theory naturally relates to cryptography and plays an important role in cryptography, that is, the strange characteristics of chaotic maps provide a method to generate random sequences that offer an academic base for an one-time pad mechanism.

In 1998, Baptista proposed a cryptosystem based on the property of ergodicity of chaotic systems.[8] There are two main defects in Baptista's cryptosystem: one is that the ciphertext is generated by performing iteration a few times, so the distribution of ciphertext is not uniform; the other is that the encryption time is too long, so the efficiency of this cryptosystem is low. To overcome these defects, some cryptanalysis and improved methods[9−12] of this algorithm have been proposed in succession, but most of them have been attacked.[13−15] On the basis of this cryptanalysis, in this paper we propose a new one-time one-key chaotic encryption algorithm, and analyse the algorithm performance.

The rest of this paper is organized as follows. In Section 2 the encryption algorithm is described. In Section 3 the experimental results are presented and the security of the algorithm is evaluated. In Section 4 some conclusions are drawn from the present study.

2. Algorithm description

The intrinsic trait of chaotic systems is highly sensitive to the initial value, that is, very similar initial conditions can generate completely different sequences after several iterations, so we cannot predict the next number by the sequence itself. This trait makes the chaotic trajectory go through different regions of the attractor, and then obtains a new sequence value, that is, a random number. The important difference between chaos and cryptography lies in the fact that the systems used in chaos are defined only on a real number, while cryptography deals with systems defined on a finite number of integers.[16] Therefore, when we design an encryption algorithm using chaotic systems,

*Project supported by the National Natural Science Foundation of China (Grant Nos. 61173183, 60973152, and 60573172), the Superior University Doctor Subject Special Scientific Research Foundation of China (Grant No. 20070141014), and the Natural Science Foundation of Liaoning Province, China (Grant No. 20082165).

†Corresponding author. E-mail: wangxy@dlut.edu.cn

‡Corresponding author. E-mail: tenglin@mail.dlut.edu.cn

© 2012 Chinese Physical Society and IOP Publishing Ltd

we need to establish a one-to-one map between the discrete values and the alphabets. So the attractor should be divided into sub-intervals and related to the ASCII values of the alphabet one to one. When the discrete value of the orbits of the chaotic system falls in some interval, there is a corresponding integer.

In Baptista's cryptosystem, the logistic map (Eq. (1)) is used as the chaotic map. When parameter $a \in (3.5699456, 4]$, $x_n \in (0, 1)$, the system is in chaotic state given as

$$x_{n+1} = ax_n(1 - x_n). \tag{1}$$

However, when $a \neq 4$, there always exist sub-intervals which the chaotic trajectory does not go through (Fig. 1). So, Baptista had to choose a part of the attractor, which is interval [0.2, 0.8] used as the encryption interval. When the trajectory does not go into interval [0.2, 0.8], the chaotic system needs to continue iterating, which affects the efficiency.

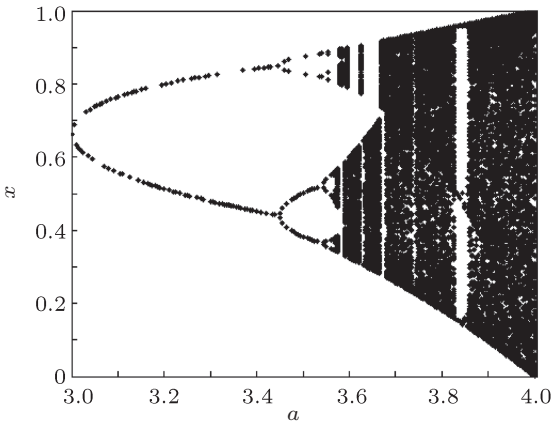


Fig. 1. Bifurcation behaviour of a logistic system.

In this paper, we use the following skew tent map, instead of a logistic map, in our scheme:

$$x_{i+1} = \begin{cases} x_i/b, & x_i \leq b, \\ (1 - x_i)/(1 - b), & x_i > b, \end{cases} \quad x_i \in [0, 1], \tag{2}$$

where $x_i \in [0, 1]$ is the state of the chaotic system, and $b \in (0, 1)$ is the control parameter. For any $b \in (0, 1)$, the piecewise linear map (2) has a positive Lyapunov exponent and thus it is always chaotic.[17] Figures 2 and 3 show the chaotic and bifurcation behaviours of system (2) for $x_0 = 0.34$ and $b = 0.3$.

The trajectory of the skew tent map can reach the entire interval [0, 1], so we choose the interval [0, 1] from the chaotic attractor, and divide it into 256 sub-intervals. According to the order from small to big, a sub-interval corresponds to one of the numbers 0, 1, 2, ..., 255. The corresponding relation is shown in Fig. 4.

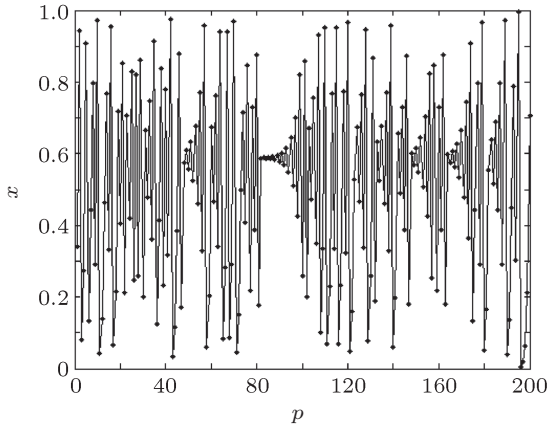


Fig. 2. Chaotic behaviour of skew tent system ($x_0 = 0.34$ and $b = 0.3$).

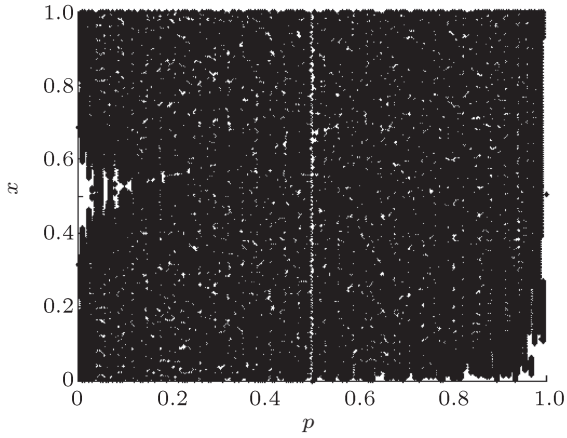


Fig. 3. Bifurcation behaviour of the skew tent system.

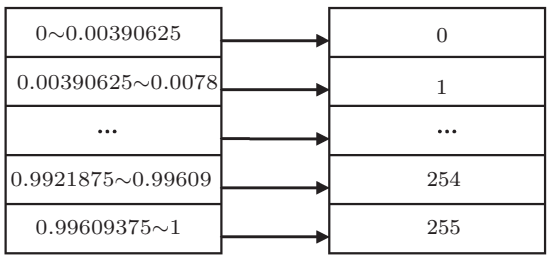


Fig. 4. Each sub-interval corresponds to an integer.

The method of encryption is detailedly described as follows.

(i) First, we divide the plaintext into blocks of same size, 8 bits (a byte) being a block in this paper, according to Baptista's plaintext dividing method, the plaintext ASCII value ($0 \sim 255$), denotes the values of plaintext letters p_i. For example, the ASCII value of '0' is 00110000, so its responding value is 48; 'a' is 01100001, the responding value is 97.

(ii) Second, divide interval [0, 1] into 256 sub-intervals as we described.

(iii) Third, The initial point x_0 of the chaotic map and parameter b can be treated as secret keys, s is the initial iteration time. In order to make the ciphertext dependent on the plaintext, we set the secret keys $x_0 = (1/p_{\text{odd}} + \text{key}_1) \bmod 1$, $b = (1/p_{\text{even}} + \text{key}_2) \bmod 1$, where p_{odd} is the total number of the former n odd position plaintext letters, p_{even} is the total number of the former n even position plaintext letters (n is integer user key), key_1, key_2 is the user keys between (0, 1). We set $s = \text{int}(b/x_0 \times 50 + (\text{key}_1 + \text{key}_2) \times 10)$.

(iv) Encryption process: iterate the skew tent map s times to obtain x_1, here x_1 belongs to the sub-interval whose corresponding integer is w_1. Then w_1 plus the ASCII p_1 of the first letter is $m_1 = w_1 + p_1$. If $m_1 < 256$, m_1 is the ciphertext c_1 of the first letter; otherwise $c_1 = m_1 - 256$.

In succession, to encrypt the i-th plaintext letter p_i, continue to iterate the chaotic map p_{i-1} times to obtain x_i. The sub-interval that x_i falls in corresponds to an integer w_i. The w_i plus the ASCII p_i of the i-th letter is $m_i = w_i + p_i$. If $m_i < 256$, m_i is the ciphertext c_i of the i-th letter; otherwise $c_i = m_i - 256$. Until all the letters of plaintext are encrypted. The encryption scheme is shown in Fig. 5.

(v) Decryption process: this is similar to the encryption process. Set the same initial parameters x_0, b, and initial iterative time s as the encryption process. Compare cipher c_i with w_i. If $c_i \geq w_i$, the plaintext ASCII value is $p_i = c_i - w_i$; otherwise, $p_i = c_i + 256 - w_i$, thereby the plaintext is recovered. Iterate chaotic map p_i times to obtain w_{i+1}, continue decryption until all the letters of cipertext are decrypted.

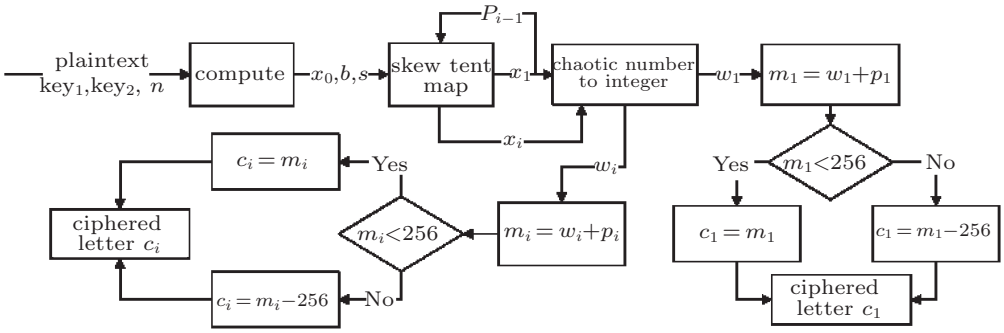


Fig. 5. Block diagram.

3. Experimental results

3.1. Encryption result

In this paper, we encrypt the plaintext "welcome to the world of chaos." as an example. We set the user keys $\text{key}_1 = 0.3$, $\text{key}_2 = 0.6$, and $n = 9$. After encryption, the ciphertext is "198 122 71 203 7 95 130 97 1 88 210 42 250 239 239 172 212 175 2 17 153 13 204 32 2 185 181 203 11 0". Then after decryption, we can obtain the correct plaintext.

3.2. Security analysis

A good encryption algorithm should resist all kinds of known attacks, such as exhaustive attack, statistical attack and chosen-plaintext/ciphertext attack. In this section, we will discuss the security analysis of the proposed encryption scheme.

3.2.1. Key space analysis

A good encryption scheme should be sensitive to the secret keys, and the key space should be large enough to make brute-force attacks infeasible. In this algorithm, the user keys key_1, key_2, and n can be used as keys as well as the plaintext letters. If the accuracy is 10^{-14}, the key space size is bigger than 10^{28}, the key space is large enough to resist the brute-force attacks.

3.2.2. Key sensitivity analysis

A good encryption scheme should be sensitive to the secret keys, that is, when secret keys change slightly, the ciphertext will vary greatly. Several key sensitivity tests are performed. Figure 6 shows the effect of encryption using the same plaintext with only a small change of secret key (10^{-14}). Figure 6(a) shows that key_1 is changed by 10^{-14}, figure 6(b) indicates

that key_2 is changed by 10^{-14}, and figure 6(c) exhibits that n is changed into 8. So it can be concluded that the proposed algorithm is sensitive to the key, a small change of the key will generate a completely different decryption result and so one cannot obtain the correct plaintext.

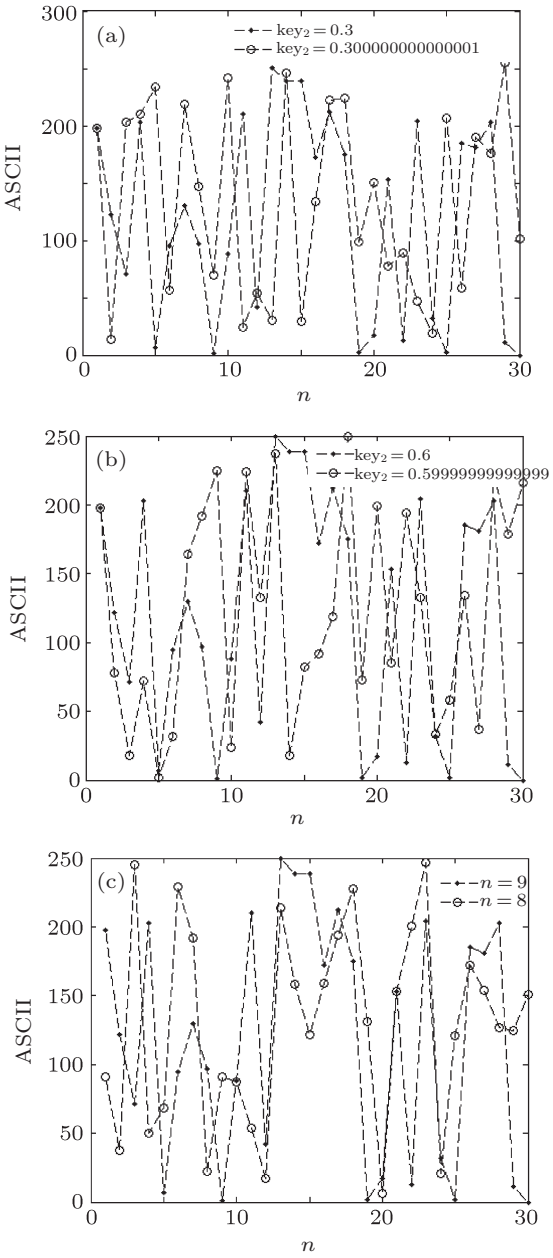


Fig. 6. The results of sensitivity analysis of the secret key.

3.2.3. Statistic characteristic

The ASCII value distribution of a section of an English article (plaintext) and the corresponding ciphertext are shown in Figs. 7(a) and 7(b), respectively. From Fig. 5, we can know that the ASCII value before encryption is between $97 \sim 122$, but the ciphertext ASCII value is distributed uniformly after encryption, which makes an attack very difficult.

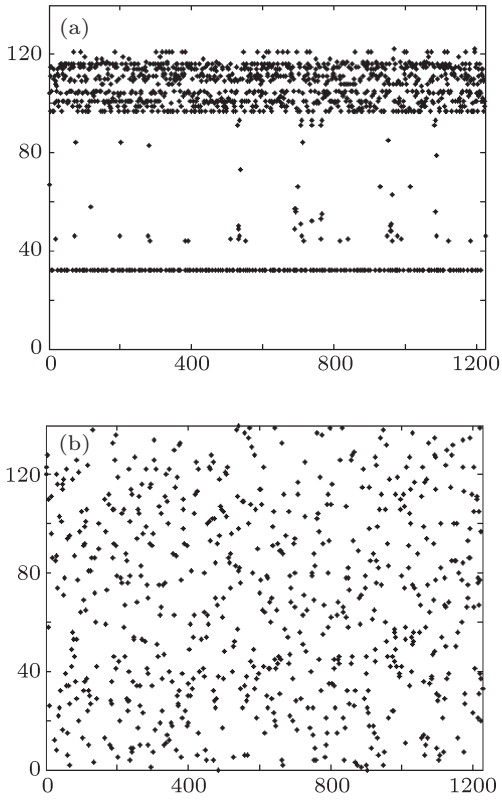


Fig. 7. The plaintext ASCII value and the cipher ASCII value distribution of the same plaintext, (a) the plaintext's ASCII distribution, (b) the ciphertext's ASCII distribution.

3.2.4. Classical types of attacks

When cryptanalyzing a cryptosystem, the general assumption made is that the cryptanalyst knows exactly the design and working of the cryptosystem under study; he knows everything about the cryptosystem except for the secret key. This is an evident requirement for today's secure communications networks, usually referred to as Kerckhoff's principle.[18] There are four classical types of attacks:

i) ciphertext only: the opponent possesses a string of ciphertext;

ii) known plaintext: the opponent possesses a string of plaintext and the corresponding ciphertext;

iii) chosen plaintext: the opponent has obtained temporary access to the encryption machinery. Hence he can choose a plaintext string, and construct the corresponding ciphertext string;

iv) chosen ciphertext: the opponent has obtained temporary access to the decryption machinery. Hence

he can choose a ciphertext string, and construct the corresponding plaintext string.

Obviously, the chosen plaintext attack is the most powerful attack. If a cryptosystem can resist this attack, it can resist other types of attacks.

The proposed algorithm is sensitive to the initial parameters x_0, b, and initial iterative time s. They are all related to the plaintext, so different plaintexts or different numbers of former position plaintext letters n can obtain parameters different from x_0, b, and s, which means that a different ciphertext has a different plaintext. So, the proposed algorithm can resist the chosen plaintext/ciphertext attack.

4. Conclusion

In this paper, we proposed a new one-time one-key encryption algorithm based on the ergodicity of a skew tent chaotic map. We divided the chaotic trajectory into sub-intervals and mapped them to integers, and used this scheme to encrypt plaintext and obtained the ciphertext. In this algorithm, the plaintext information in the keys is used, so different plaintexts or different total numbers of plaintext letters would encrypt different ciphertexts. Simulation results show that the performance and the security of the proposed encryption algorithm can encrypt plaintext effectively and resist various typical attacks.

References

[1] Wang X Y and Gao Y F 2010 *Nonlinear Sci. Numer. Simulat.* **15** 99

[2] Ashraf A. Z and Abdulnasser A R 2011 *Nonlinear Sci. Numer. Simulat.* **16** 3721

[3] Liu Z B, Zhang H G and Sun Q Y 2010 *Chin. Phys.* B **19** 090506

[4] Zhang H G, Ma T D, Fu J and Tong S C 2009 *Chin. Phys.* B **18** 3742

[5] Zhang H G, Ma T D, Yu W and Fu J 2008 *Chin. Phys.* B **17** 3616

[6] Long M, Peng F and Chen G R 2008 *Chin. Phys.* B **17** 3588

[7] Wang K, Pei W J, Zou L H and He Z Y 2006 *Acta Phys. Sin.* **55** 6243 (in Chinese)

[8] Baptista M S 1998 *Phys. Lett.* A **240** 50

[9] Wong W K, Lee L P and Wong K W 2001 *Comput. Phys. Commun.* **138** 234

[10] Wong K W 2002 *Phys. Lett.* A **298** 238

[11] Wong K W 2003 *Phys. Lett.* A **307** 292

[12] Wong K W, Ho S W and Yung C K 2003 *Phys. Lett.* A **310** 67

[13] Jakimoski G and Kocarev L 2001 *Phys. Lett.* A **291** 381

[14] Wang Y, Liao X F, Xiang T, Wong K W and Yang D G 2007 *Phys. Lett.* A **363** 277

[15] Rhouma R, Solak E, Arroyo D, Li S J, Alvarez G and Belghith S 2009 *Phys. Lett.* A **373** 3398

[16] Kocarev L and Jakimoski G 2001 *Phys. Lett.* A **289** 200

[17] Alvarez G and Li S 2006 *Int. J. Bifurc. Chaos* **16** 2129

[18] Pareek N K, Patidar V and Sud K K 2005 *Nonlinear Sci. Numer. Simulat.* **10** 715