Journal of Information & Computational Science 12:2 (2015) 515–523 January 20, 2015
Available at http://www.joics.com

Improving the Security of Pseudo-random Sequence Generator Based on Chen Chaotic System ⋆

Peiyue Li [a,*], Junxia Shi [b], Pengzhi Li [a], Yongxin Sui [a], Huaijiang Yang [a]

[a] *State Key Laboratory of Applied Optics, Changchun Institute of Optics, Fine Mechanics and Physics Chinese Academy of Sciences, Changchun 130033, China*

[b] *Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences Changchun 130033, China*

Abstract

In this letter, the cause of vulnerability of the original pseudo-random sequence generator based on Chen chaotic system is analyzed, and the corresponding enhancement methods are proposed. Transient data produced by the calculation among fixed-point numbers is used to improve the performance of the original scheme. Statistical tests and security analysis indicate that the modified scheme is more secure than the original one, and the computational complexity of the brute force attack is $O(2^n)$. At the same time, it can still maintain the pseudo-random characteristics and satisfy the other performance requirements of pseudo-random sequence generator.

Keywords: Chaotic System; Pseudo-random Sequence Generator; Transient Data; Brute Force Attack

1 Introduction

As an important part of modern cryptography, pseudo-random binary sequences have been widely investigated in the past decades. Most existing schemes for generating pseudo-random sequences are based on linear feedback shift registers, oscillator, cellular automata rules, etc. [1]-[4]. In recent years, chaotic systems have been used to develop pseudo-random sequence generator because the sensitivity to parameters and initial conditions, ergodicity, and pseudo-random behavior of chaotic systems satisfy the analogous requirements for a good cryptosystem. Li et al. generated multiple pseudo-random-bit sequences from a single spatiotemporal chaotic system (CML-MPRBG) [5]. Kanso et al. generated pseudo-random binary sequences by logistic chaotic map [6], and Persohn et al. explored the consequences finite precision has on the periodicity of a PRNG based on the logistic map [7]. In 2011, Liu indicated the pseudo-randomness and complexity of the binary sequences generated by the Chebyshev map and the Lorenz system [8].

⋆Project supported by the National Basic Research Program of China (No. 2007CB311201) and the Project Development Plan of Science and Technology of Jilin Province (No. 20130522120JH).

*Corresponding author.
Email address: lipy@sklao.ac.cn (Peiyue Li).

DOI: 10.12733/jics20105237

In 2013, Palacios-Luengas et al. shown a digital electronic system that produces uniformly distributed binary sequences using the Inverted Tent Chaotic Map (ITCM) without the scaling and discretization processes [9]. In 2014, Francois et al. proposed a secure pseudo-random number generator three-mixer [10]. The most used chaotic systems for generating pseudo-random sequences above are one-dimensional chaotic maps. It is more suitable to construct pseudo-random sequences generators by using high-dimensional chaotic systems. To overcome this limitation, Hu et al. proposed a binary sequence generator by using the high-dimensional Chen chaotic system [11]. Their pseudo-random bit generator is based on a combination of three coordinates of the chaotic orbit. However, Francois M. et al. pointed out the security weaknesses of this scheme [12]. By applying a brute force attack on a reduced key space, Francois M. et al. shown that 66% of the generated sequences can be revealed.

In this letter, we will analyze the cause of vulnerability of the original pseudo-random sequence generator based on Chen chaotic system in detail, and then propose the corresponding enhancement measures. Statistical tests and security analysis will be performed to evaluate the security of the modified scheme. The rest of this letter is organized as follows: Section 2 discusses the main security problems of the original scheme. In Section 3, a modified scheme is proposed by using the Chen chaotic system which is realized on 32-bits fixed-point calculations. The statistical tests and security analysis of the modified scheme are presented in Section 4. Section 5 concludes the letter.

2 Original Scheme and Its Security Problems

2.1 Description of the Pseudo-random Sequence Generator

In the original pseudo-random sequence generator, Chen chaotic system is utilized. Chen chaotic system is defined as:

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = (c - a)x - xz + cy \\ \dot{z} = xy - bz \end{cases} \tag{1}$$

where $a = 35, b = 3, c = 28$, and the initial values $(x_0, y_0, z_0) = (-3, 2, 20)$.

In order to generate sequences of uniform distributed binary random variables by using the Chen chaotic system, effective changes were made to enhance the random statistical properties in [11]. The pseudo-random bit generator is based on a combination of three coordinates of the chaotic orbit. The pseudo-random sequence generator steps as follows:

Step 1. The outputs of the Chen chaotic system are computed for the selected initial conditions and control parameters.

Step 2. Eq. (2) is used to generate the chaotic pseudo-random key stream, where $x(i), y(i), z(i)$

are the samples of the Chen chaotic system.

$$\begin{aligned} v(3i) &= 3000 \times (x(i) + 45) \\ v(3i+1) &= 3000 \times (y(i) + 35) \\ v(3i+2) &= 3000 \times (z(i) + 45) \\ K(j) &= v(j) \bmod 256, i, j = 0, 1, 2, ... \end{aligned} \tag{2}$$

Step 3. Generate pseudo-random key stream by encoding binary representation.

2.2 Security Problems

For a positive integer n, two integers a and b are said to be congruent modulo n, written $a \equiv b(\text{mod} \quad n)$. The properties of the congruence relation can be described as follows:

$$a_1 \equiv b_1(\text{mod}\, n) \tag{3}$$

$$a_2 \equiv b_2(\text{mod}\, n) \tag{4}$$

$$a_1 \mp a_2 \equiv b_1 \mp b_2(\text{mod}\, n) \tag{5}$$

$$a_1 a_2 \equiv b_1 b_2(\text{mod}\, n) \tag{6}$$

By using the properties of modular arithmetic listed above, Eq. (2) can be rearranged:

$$\begin{aligned} v(3i) &= (3000 \bmod 256) \times ((x(i) + 45) \bmod 256) \\ v(3i+1) &= (3000 \bmod 256) \times ((y(i) + 35) \bmod 256) \\ v(3i+2) &= (3000 \bmod 256) \times ((z(i) + 45) \bmod 256) \\ K(j) &= v(j) \bmod 256, i, j = 0, 1, 2, ... \end{aligned} \tag{7}$$

Although the Chen chaotic system is working with continuous values, the range is transformed into [0, 255] with the mod operation. The maximum and minimum values of the digital Chen chaotic system are given in Table 1. It can be seen that there are 44, 53, and 46 different values for $(x(i) + 45), (y(i) + 35), (z(i) + 45)$. In order to decrypt the ciphered image proposed in [11], the attacker is suggested to reveal the $v(j)$ values instead of attacking the initial conditions. Once the $v(j)$ values are known, the ciphered image is decrypted. Any generated pseudo-random number value can be found with a maximum of 53 attempts. In this situation, the computational complexity of the attack is $O(53n) = O(n)$, where n is the sequence length.

Table 1: Probable output values for Eq. (2)

	Minimum value	Maximum Value	Number of values
x(i)+45	20	64	44
y(i)+35	5	59	53
z(i)+45	45	91	46

3 Description of the Modified Scheme

It is obvious that there are $2N-1$ bits output for N-bits multiplication or division. However, while performing N-bits fixed-point calculation, the input and output are both N bits. In other words, the residual $N-1$ bits are ignored. Theoretical analysis and computer simulation indicate that the cryptographic characteristics of this $N-1$ bits are better than the N bits output [13]. We name the ignored $N-1$ bits as transient data. By using the transient data, the main frame of our modified scheme is designed as Fig. 1, and its procedure can be described as follows:

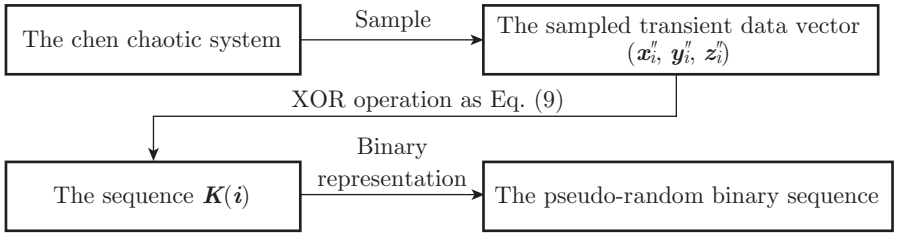


Fig. 1: The main frame of the modified scheme

Step 1. Chen chaotic system is realized under 32-bits fixed-point calculations for the selected initial conditions and control parameters. Assume that (x_0, y_0, z_0) is the initial vector and (x_i', y_i', z_i') is the 31-bits transient data output vector of the ith iteration.

Step 2. We can obtain 32-bits vector (x_i'', y_i'', z_i'') by padding x_i', y_i', z_i' with '0' at the side of the last bit. We write the value of x_i'', y_i'', z_i'' in an integer number with 32 bits. The values x_i'', y_i'', z_i'' are re-divided into 4 integers as Eq. (8), and each integer has 8 bits.

$$\begin{aligned} x_i'' &= \underbrace{\alpha_{31}\alpha_{30}...\alpha_{24}}_{x_{i.4}''}\underbrace{\alpha_{23}\alpha_{22}...\alpha_{16}}_{x_{i.3}''}\underbrace{\alpha_{15}\alpha_{14}...\alpha_{8}}_{x_{i.2}''}\underbrace{\alpha_{7}\alpha_{6}...\alpha_{1}0}_{x_{i.1}''} \\ y_i'' &= \underbrace{\beta_{31}\beta_{30}...\beta_{24}}_{y_{i.4}''}\underbrace{\beta_{23}\beta_{22}...\beta_{16}}_{y_{i.3}''}\underbrace{\beta_{15}\beta_{14}...\beta_{8}}_{y_{i.2}''}\underbrace{\beta_{7}\beta_{6}...\beta_{1}0}_{y_{i.1}''} \\ z_i'' &= \underbrace{\gamma_{31}\gamma_{30}\cdots\gamma_{24}}_{z_{i.4}''}\underbrace{\gamma_{23}\gamma_{22}\cdots\gamma_{16}}_{z_{i.3}''}\underbrace{\gamma_{15}\gamma_{14}\cdots\gamma_{8}}_{z_{i.2}''}\underbrace{\gamma_{7}\gamma_{6}\cdots\gamma_{1}0}_{z_{i.1}''} \end{aligned} \tag{8}$$

where $\alpha_i, \beta_i, \gamma_i \in \{0,1\}$, $i = 1, 2, ..., 31$.

Step 3. The chaotic pseudo-random key stream is generated by Eq. (9), where $\oplus$ is XOR operation.

$$\begin{aligned} K(3i) &= x_{i.4}'' \oplus x_{i.3}'' \oplus x_{i.2}'' \oplus x_{i.1}'' \\ K(3i+1) &= y_{i.4}'' \oplus y_{i.3}'' \oplus y_{i.2}'' \oplus y_{i.1}'' \\ K(3i+2) &= z_{i.4}'' \oplus z_{i.3}'' \oplus z_{i.2}'' \oplus z_{i.1}'' \end{aligned} \tag{9}$$

Step 4. Generate pseudo-random key stream by encoding binary representation.

4 Performance Analysis of the Modified Scheme

The numerical experiments are performed on 32-bits fixed-point calculations. In order to estimate the randomness of a pseudo-random binary sequence, various tests should be available to evaluate

a PRNG for cryptographic purposes. The following measures must be selected: Statistical test, Correlation test, Frequency spectral test, Security test et al..

4.1 Statistical Test

The randomness of a sequence can be estimated by using the method of statistical test. In fact, the National Institute of Standards & Technology (NIST) proposes a battery of tests that must be performed on the generated binary sequences. These tests assess the presence of a pattern which, if detected, would indicate that the sequence is not random. In each statistical test, a p_value probability is computed. Each value summarizes the strength of the evidence against the perfect randomness assumption. A p_value of zero indicates that the sequence appears to be completely not random. A p_value larger than 0.01 means that the sequence is considered to be random with a confidence level of 99%. The results of NIST tests obtained on 10 groups of 1 000 000 sequences are given in Table 2. We can remark that all the tested sequences pass the NIST tests successfully. These results show a high randomness level of the produced binary sequences.

Table 2: Results of SP 800-22 test

Test name	p_value	Results
Frequency	0.5341	SUCCESS
Block frequency	0.4944	SUCCESS
Cumulative sums (1)	0.7776	SUCCESS
Cumulative sums (2)	0.4642	SUCCESS
Runs	0.1654	SUCCESS
Longest runs of ones	0.3505	SUCCESS
Rank	0.2133	SUCCESS
FFT	0.3505	SUCCESS
Overlapping template matching	0.2364	SUCCESS
Universal statistical	0.3505	SUCCESS
Approximate entropy	0.4731	SUCCESS
Random excursions	0.6421	SUCCESS
Random excursions variant	0.6158	SUCCESS
Serial (1)	0.3201	SUCCESS
Serial (2)	0.4237	SUCCESS
Linear complexity	0.5863	SUCCESS

4.2 Correlation Test

There are two types of correlation tests for pseudo-random binary sequence estimation: the auto-correlation and the cross-correlation. The auto-correlation of binary sequences measures the amount of similarity between the sequences x_n and a shift of x_n by m positions. $\delta-$like auto-correlation is required for a good pseudo-random binary sequence generator. On the other hand, the binary sequences can be generated from pseudo-random sequence generator simultaneously.

If being independent of each other with zero cross-correlation, they can be used to encrypt many plain-texts at one time. The auto-correlation and cross-correlation function, Corr(m), for a binary sequence can be defined by Eq. (10) at the same time.

$$Corr(m) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} (x_{i+m} - \bar{x})(y_i - \bar{y}) \tag{10}$$

where

$$\bar{x} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} (x_i) \tag{11}$$

$$\bar{y} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} (y_i) \tag{12}$$

where, x_i, y_i is the binary symbol of the same or different sequences, N and m are the length and offset of the sequence, respectively. The auto-correlation and the cross-correlation function of binary sequences generated by the modified scheme is shown in Fig. 2. It is clearly shown that the binary sequences have good performance of correlation. It should be noted that the length of the tested sequences is 10^5 bits.

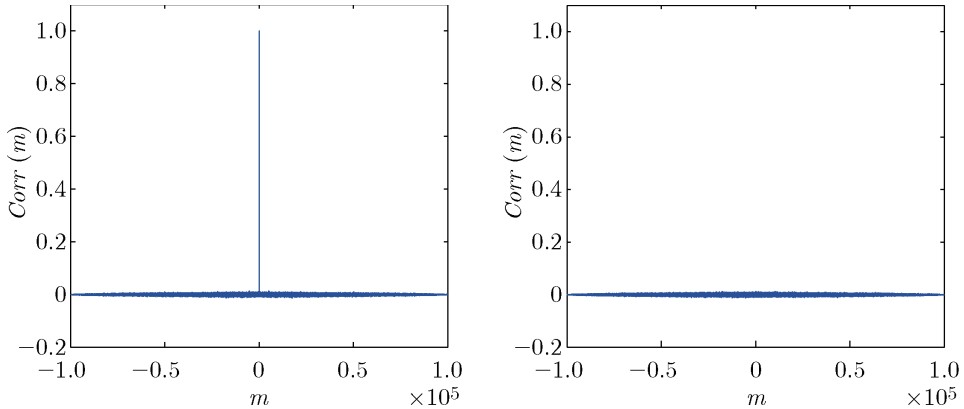


Fig. 2: The auto-correlation and the cross-correlation function

4.3 Frequency Spectral Test

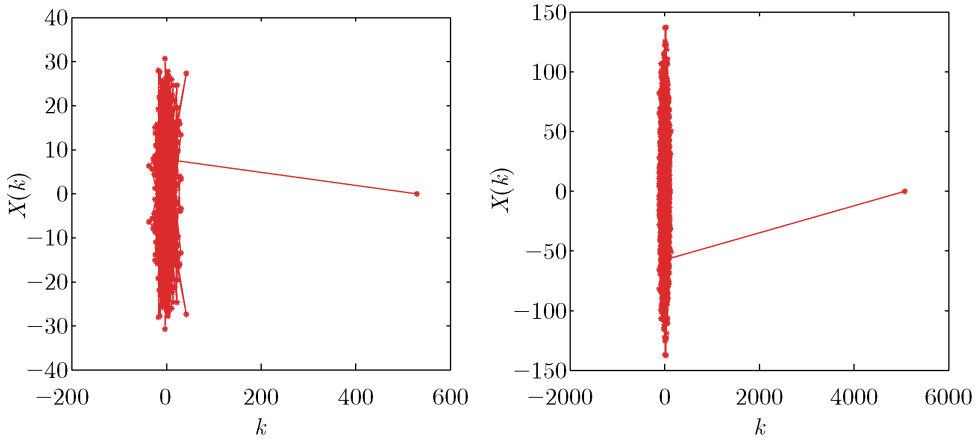
The measure of the frequency spectral analysis tests whether the center frequency exists or not in the binary sequence. If there is a center frequency in a sequence, it has periodicity, and is not an ideal pseudo-random sequence. The frequency spectrum of sequence x_n can be defined as follows:

$$X(k) = \sum_{n=0}^{N-1} x_n e^{-j2\pi nk/N} \tag{13}$$

where, N is the length of a sequence, and k is the rank of a harmonic, $0 \le k \le N$.

For the frequency spectrum of the binary sequences generated by the modified scheme, we randomly select 10^3 and 10^4 sequential bits for each sequence. The results of the frequency

spectral analysis for the selected binary sequences are shown in Fig. 3. It is obvious that there is no center frequency in the spectrum of all binary sequences. Hence, these sequences generated by the modified scheme are all aperiodic.



(a) With 1000 bits length (b) With 10000 bits length

Fig. 3: Frequency spectrum of the selected binary sequences with 10^5 bits length

4.4 Security Test

4.4.1 Key Sensitivity

According to the basic characteristics of chaotic system, we can choose the initial values and control parameters as the secret key of pseudo-random binary sequence generator. In order to evaluate the sensitivity of the modified scheme to the secret key, simulation experiments have been done under the following 5 different conditions:

G1 : $(x_0, y_0, z_0) = (-3, 2, 20),\ a = 35, b = 3, c = 28$

G2 : $(x_0, y_0, z_0) = (-3 + \varepsilon, 2, 20),\ a = 35, b = 3, c = 28$

G3 : $(x_0, y_0, z_0) = (-3, 2 + \varepsilon, 20),\ a = 35, b = 3, c = 28$

G4 : $(x_0, y_0, z_0) = (-3, 2, 20),\ a = 35 + \varepsilon, b = 3, c = 28$

G5 : $(x_0, y_0, z_0) = (-3, 2, 20),\ a = 35, b = 3, c = 28 + \varepsilon$

where $(x_0, y_0, z_0), (a, b, c)$ are the initial values and control parameters of Chen chaotic system, $\varepsilon = 10^{-14}$.

The number of differences among the above conditions is shown in Table 3. For 10^5 bits test binary sequences, it is clear that the variance ration of each bit is approximated 50% even if the change of initial value or parameter is an extremely small value 10^{-14}. The simulation result indicates that the key sensitivity property of the modified scheme is so perfect.

Table 3: Results of key sensitivity test

-	G1	G2	G3	G4	G5
G1	-	49975	49778	50094	49928
G2	-	-	49481	50045	50033
G3	-	-	-	49920	50160
G4	-	-	-	-	49662
G5	-	-	-	-	-

4.4.2 Brute Force Attack

For the original scheme, the computational complexity of the brute force attack is $O(53n) = O(n)$, where n is the sequence length. The brute force attack is applied to evaluate the security of the modified scheme as well. Since 32-bits precision fixed-point number is adopted as the representation of the real number on computer in the algorithm, there are 31-bits transient data during the calculation between fixed-point numbers. While the transient data is unavailable for attackers, the probability to guess each bit in the binary sequence is 2 according to the description of the modified scheme. In other words, the computational complexity of the brute force attack is $O(2^n)$ for the modified scheme.

5 Conclusion

In this letter, the cause of vulnerability of the original pseudo-random sequence generator based on Chen chaotic system is analyzed in detail. Enhancement methods are proposed to get rid of the reported cryptanalysis. Transient data produced by the calculation between fixed-point numbers is used to improved the performance of the original scheme. Statistical tests and security analysis indicate that the modified scheme is more secure than the original one. The computational complexity of the brute force attack is $O(2^n)$. Additionally, it still maintains the pseudo-random characteristics and satisfies the other performance requirements of pseudo-random sequence generator.

References

[1] A. Peinado, A. Fuster-Sabater, Generation of psudorandom binary sequences by means of linear feedback shift registers with dynamic feedback [J], Mathematical and Computer Modelling, 57(11-12), 2013, 2596-2604

[2] K. H. Mak, More constructions of pseudorandom sequences of k symbols [J], Finite Fieled and Their Appliacations, 25(1), 2014, 222-233

[3] X. N. Du, A. Klapper, Z. X. Chen, Linear complexity of pseudorandom sequence generated by Fermat quotients and their generalizations [J], Information Processing Letters, 112(6), 2012, 233-237

[4] G. Pirsic, A. Winterhof, On the structure of digital explicit nonlinear and inversive pseudorandom number generators [J], Journal of Complexity, 26(1), 2010, 43-50

[5] P. Li, Z. Li, W. A. Halang, G. R. Chen, A multiple pseudorandom-bit generator based on a spatiotemporal chaotic map [J], Physics Letters A, 349(6), 2006, 467-473

[6] A. Kanso, N. Smaoui, Logistic chaotic maps for binary numbers generations [J], Chaos, Solitons & Fractals, 40(5), 2009, 2557-2568

[7] K. J. Persohn, R. J. Povinelli, Analyzing logistic map pseudorandom number generators for periodicity induced by finite precision floating-point representation [J], Chaos, Solitons & Fractals, 45(3), 2009, 238-245

[8] N. S. Liu, Pesudo-randomness and complexity of binary sequences generated by the chaotic system [J], Communications in Nonlinear Science and Numerical Simulation, 16(2), 2011, 761-768

[9] L. Palacios-Luengas, G. Delgado-Gutierrez, M. Cruz-Irisson, J. L. Del-Rio-Correa, R. Vazquez-Medina, Digital noise produced by a non discretized tent chaotic map [J], Microelectronic Engineering, 112(1), 2013, 264-268

[10] M. Francois, T. Grosges, D. Barchiesi, R. Erra, Pseudo-random number generator based on mixing of three chaotic maps [J], Communications in Nonlinear Science and Numerical Simulation, 19(4), 2014, 887-895

[11] H. Hu, L. Liu, N. Ding, Pseudorandom sequence generator based on Chen chaotic system [J], Computer Physics Communications, 184(3), 2013, 765-768

[12] F. Ozkaynak, S. Yavuz, Security problems for a pseudorandom sequence generator based on the Chen chaotic system [J], Computer Physics Communications, 184(9), 2013, 2178-2181

[13] L. Y. Sheng, Y. Y. Xiao, Z. Sheng, A universal algorithm for transforming chaotic sequences into uniform pseudo-random sequences [J], Acta Physica Sinica, 57(7), 2008, 4007-4013