J. Software Engineering & Applications, 2010, 3, 1141-1147
doi:10.4236/jsea.2010.312133 Published Online December 2010 (http://www.scirp.org/journal/jsea)

An Efficient PRBG Based on Chaotic Map and Engel Continued Fractions

Atef Masmoudi[1,2], William Puech[2], Mohamed Salim Bouhlel[1]

[1]Research Unit: Sciences and Technologies of Image and Telecommunications, Higher Institute of Biotechnology, University of Sfax, Sfax, Tunisia; [2]Laboratory LIRMM, University of Montpellier II, Montpellier, France
E-mail: atef.masmoudi@lirmm.fr, william.puech@lirmm.fr, medsalim.bouhlel@enis.rnu.tn

Received March 3rd, 2010; revised April 12th, 2010; accepted April 16th, 2010.

ABSTRACT

In recent years, a variety of chaos-based cryptosystems have been proposed. Some of these systems are used in designing a pseudo random bit generator (PRBG) for stream cipher applications. Most of the chaotic systems used in cryptography have good chaotic properties like ergodicity, sensitivity to initial values and sensitivity to control parameters. However, some of them are not very suitable for use in cryptography because of their non-uniform density function, and their relatively small key space. To be used in cryptography, a PRBG may need to meet stronger requirements than for other applications. In particular, various statistical tests can be applied to the outputs of such generators to conclude whether the generator produces a truly random sequence or not. In this paper, we propose a PRBG based on the use of the standard chaotic map with large key space and the Engle Continued Fractions (ECF) map. The outputs of the standard map are used as the inputs of ECF-map. The chaotic nature of the standard map and the good statistical properties of the ECF map motivate us to design a new PRBG for stream cipher applications. The numerical simulation analysis indicates that our PRBG produces bit sequences possessing excellent statistical and cryptographic properties.

Keywords: *Cryptography, Continued Fraction, Statistical Tests, PRBG, Chaotic Map*

1. Introduction

Recently, a variety of crypto-systems have been proposed. Many of them are based on chaotic systems [1-5] which possess good cryptographic characteristics. Chaos systems have many important features like ergodicity, sensitivity to initial condition, sensitivity to control parameters and randomness. These features are very important in cryptography as they form the basis of some new and efficient ways to develop encryption algorithms for secure digital image transmission over the Internet and through public networks. In addition, most of the chaos-based image cryptosystems are based on a single chaotic system. And it is possible according to the chaos theory to extract some useful information about the chaotic system from its orbit, which makes chaotic systems insecure [6-8]. To overcome these drawbacks, some techniques such as multiple chaotic systems [9-12], high-dimensional chaotic systems [13-15], multiple iterations of chaotic systems, and many other techniques have been proposed to improve chaos-based ciphers [6,16]. To be used in cryptographic applications, a chaotic system must sat isfy two important characteristics; its large key space, to resist bruit force attacks, and its ability to generate a sequence that has a uniform invariant density function to make it resistant to statistical attacks. The problem is that not all chaotic systems satisfy these characteristics. For example, the chaotic logistic map is widely used to design cryptosystems. The known one-Dimensional logistic map is defined as $x_{n+1} = \lambda x_n (1 - x_n)$ where $\lambda \in [0,4]$, $n = 0,1,2,\cdots$ and $x_n \in [0,1)$. Mi *et al*. [17] proposed a chaotic encryption scheme based on randomized arithmetic coding using the logistic map as the PRBG. In [18], Kanso and Smaoui proposed two Pseudo Random Bit Generator (PRBG) for stream cipher applications. The first is based on a single 1-D logistic map and the second is based on a combination of two logistic maps. The logistic map is weak in security because it neither satisfies the uniform distribution property nor does its key space [19,20]. For this map, the key size is determined by the initial value x_0 and the control parameter λ; 50 bits with a precision of 10^{-14}. In [11], Patidar and Sud proposed a PRBG for stream cipher applications based on two chaotic standard maps running side-by-side and starting from randomly inde pendent initial conditions. The pseudo random bit sequence is generated by comparing the outputs of both the chaotic standard maps. They presented the detailed results of the statistical testing on generated bit sequences, using two statistical test suites: the NIST [21] and the DIEHARD [22]. Thus, the need to find a secure and efficient chaotic-based cryptosystem motivates us to propose a new scheme which consists of using the standard chaotic map and the Engle Continued Fractions (ECF) map. The use of ECF-map increases the complexity of a cryptosystem based only on standard chaotic system and thus makes difficulties in extraction of information about it. In addition, ECF-map conserves the cryptography properties of the chaotic system; like sensitivity to initial values/control parameters, non periodicity and randomness; and adds interesting statistical properties such uniform distribution density function and zero co-correlation.

The rest of this paper is organized as follows. In Section 2, we discuss the CF theory and we present a brief description of the ECF-map along with some important features. Section 3 details our proposed PRBG for stream cipher applications. In Section 4, we analyze the security of the proposed PRBG and discuss experimental results based on statistical testing. The concluding remarks are given in Section 5.

2. Continued Fraction

Continued fractions (CF) [23-26] refer to all expressions of the form:

$$x = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cfrac{a_4}{\cdots}}}} \qquad (1)$$

Where a_i (i > 0) are the partial numerators, b_i are the partial denominators, b_0 is the integer part of the continued fraction and x is a real number.

Hartono *et al.* [27] introduce a new continued fraction expansion, called Engel continued fraction (ECF) expansion.

The Engel continued fraction (ECF) map $T_E : [0,1) \to [0,1)$ is given by:

$$T_E(x) = \begin{cases} \dfrac{1}{\left\lfloor \dfrac{1}{x} \right\rfloor}\left(\dfrac{1}{x} - \left\lfloor \dfrac{1}{x} \right\rfloor\right) & if \;\; x \neq 0 \\ 0 & if \;\; x = 0 \end{cases} \qquad (2)$$

For any $x \in [0,1)$, the ECF-map generates a new and unique continued fraction expansion of x of the form:

$$x = \cfrac{1}{b_1 + \cfrac{b_1}{b_2 + \cfrac{b_2}{b_3 + \ddots + \cfrac{b_{n-1}}{b_n + \ddots}}}}, \; b_n \in IN, \; b_n \le b_{n+1} \qquad (3)$$

Let $x \in [0,1)$, and define:

$$b_1 = b_1(x) = \left\lfloor \frac{1}{x} \right\rfloor$$

$$b_n = b_n(x) = b_1\left(T_E^{n-1}(x)\right), n \ge 2, \; T_E^{n-1}(x) \neq 0, \qquad (4)$$

Where $T_E^0(x) = x$ and $T_E^n(x) = T_E\left(T_E^{n-1}(x)\right)$ for $n \ge 1$. From definition of T_E it follows that:

$$\begin{aligned} x &= \frac{1}{b_1 + b_1 T_E(x)} \\ &= \cfrac{1}{b_1 + \cfrac{b_1}{b_2 + \cfrac{b_2}{b_3 + \ddots + \cfrac{b_{n-1}}{b_n + b_n T_E^n(x)}}}}, \end{aligned} \qquad (5)$$

We describe the method for generating the ECF-continued fraction expansion of x as follows.

Algorithm 1 ECF expansion

```
Initialize $x_0 \leftarrow x$, $i \leftarrow 0$
while $x_1 \neq 0$ do
    $i \leftarrow i + 1$
    $b_i \leftarrow \left\lfloor \frac{1}{x_{i-1}} \right\rfloor$
    $x_i \leftarrow \frac{1}{\left\lfloor \frac{1}{x_{i-1}} \right\rfloor}\left(\frac{1}{x_{i-1}} - \left\lfloor \frac{1}{x_{i-1}} \right\rfloor\right)$
end while
```

From the theorem presented in [27], if we let $x \in [0,1)$, then x has a finite ECF-expansion (*i.e.*, $T_E^n(x) = 0$ for some $n \ge 1$) if and only if $x \in Q$. In this paper, we pay most attention to the following sequence:

$$Z_n(x) = b_n(x) T_E^n(x), n \ge 1. \qquad (6)$$

The sequence $\{Z_i(x)\}_{i=1}^n$ is in $[0,1)$ and uniformly distributed for almost all points x (for a proof see [27]). So, the ECF-map generates a random and unpredictable sequence $\{Z_i(x)\}_{i=1}^n$ with a uniform distribution. These properties, which are very useful in cryptography, motivate us to propose a new PRBG for stream cipher applications based on ECF-map.

3. The Proposed Pseudo Random Bit Generator Algorithm

In this section, we describe the process of the proposed PRBG. The first step in designing the proposed PRBG is to choose an n-Dimensional chaotic map [10,28]. Choosing maps for encryption algorithms is the most important task. The use of chaotic maps can make the output very sensitive to the input and in our PRBG, the outputs of the chosen chaotic map are used as the input to the ECF-map for generating sequences with desirable chaotic and statistical properties. In the proposed PRBG, we suggest to use the standard map due to their good chaotic properties like sensitivity to the initial values, sensitivity to the control parameter and its large key space. The 2-D map function known as the standard map is defined by:

$$\begin{cases} x_j = x_{j-1} + p_0 \times \sin\left(y_{j-1}\right) \\ y_j = y_{j-1} + x_j \end{cases}, \tag{7}$$

Where x_j and y_j are taken modulo 2π. The secret key in the proposed PRBG is a set of three floating point numbers and one integer $\left(x_0, y_0, p_0, N_0\right)$, where $\left(x_0, y_0\right) \in [0, 2\pi)$ is the initial values, p_0 is the control parameter value which can have any real value greater than 18.0 and N_0 is the number of initial iterations of the standard chaotic map. The standard map has a good chaotic properties and a large key space of the order 157 bits [10] with an accuracy of 10^{-14}. This key space is sufficient enough to resist the brute-force attack. In the following paragraph, we give the detailed procedure to generate pseudo random binary sequences using the standard and ECF maps.

We define a function $G:[0,1) \to [0,1)$ such that:

$$G\left(x_i\right) = \sum_j Z_j\left(x_i\right) - \left\lfloor \sum_j Z_j\left(x_i\right) \right\rfloor, \tag{8}$$

Where $\left\{Z_j\right\}$ is the set calculated according to (6) using ECF-map. In addition, assuming that we have defined a function $F:[0,1] \to \{0,1\}$ that converts the real number x_i to a discrete bit symbol as follows:

$$F\left(x_i\right) = \begin{cases} 0 & if \quad x_i < 0.5 \\ 1 & otherwise \end{cases}, \tag{9}$$

We propose to use the 2-D standard map, with parameters $\left(x_0, y_0\right) \in [0, 2\pi)$ are the initial values and p_0 is the control parameter of the chaotic map. Firstly, we propose to iterate the chaotic map p_0 times. And the operation procedures of the proposed PRBG are described as follows:

1) **Step 1**: The standard map is iterated continuously. For the jth iteration, the output of the standard map is a new 2-tuplet $\left(x_j, y_j\right) \in [0, 2\pi)$.

2) **Step 2**: Now, we propose to calculate the set $\left\{a_j\right\}_{j=1}^{N}$ using the relation:

$$a_j = \left(x_j + y_j\right) - \left\lfloor \left(x_j + y_j\right) \right\rfloor \tag{10}$$

3) **Step 3**: Finally, the sequence $K^N = \left\{k_j\right\}_{j=1}^{N}$ represents the random binary sequence and it is generated by:

$$k_j = F\left(G\left(a_j\right)\right) \tag{11}$$

The standard and ECF maps are iterated until the generation of a key stream with length N. In order to generate the random binary sequence $\left\{k_j\right\}_{j=1}^{N}$, an initial sequence $\left\{a_j\right\}_{j=1}^{N}$ has to be created using the standard map. From a cryptographic point of view, the sequence $\left\{a_j\right\}_{j=1}^{N}$ is not good enough for designing a PRBG because it is not random. Therefore, we propose to use the ECF-map to convert the generated sequence $\left\{a_j\right\}_{j=1}^{N}$ to a binary sequence $\left\{k_j\right\}_{j=1}^{N}$ of the same length by applying (11).

4. Statistical Tests

In this section, we apply the statistical test suite designed by NIST [21] to verify the randomness of the binary sequences generated by our PRBG. The NIST test suite includes 16 independent and computationally intensive statistical tests. These tests are useful in detecting deviations of a binary sequence from randomness. The 16 tests can be classified into two categories:

1) **Non-parameterized tests**: Frequency (monobit) test (FT), Runs test (RT), Test for longest run of ones in a block (LROT), Lempel-Ziv compression test (LZT), Binary matrix rank test (MRT), Cumulative sums test (CST), Discrete Fourier transform (spectral) test (SPT), Random excursions test (RET) and Random excursions variant test (REVT).

2) **Parameterized tests**: Frequency test within a block (BFT), Approximate entropy test (AET), Linear complexity test (LCT), Maurer's universal statistical test (MUST), Serial test (ST), Overlapping template matching test (OTMT) and Non-overlapping template matching test (NOTMT). The detailed description of all 16 tests of NIST suite can be found in [21].

4.1. Statistical Analysis of the Proposed PRBG

First, the generator should produce a binary sequence of 0s and 1s of a given length N. Next, we invoke the NIST

Statistical Test Suite using the generated sequence. Finally, a set of p_values for each statistical test will be generated by the test suite. Based on these p_values, a conclusion regarding the quality of the sequences can be drawn. The significance level α for all tests in NIST Suite is set to 0.01. A p_value less than α would mean that the sequence is non-random. If a p_value is greater than α, we accept the sequence as random. In **Table 1**, we list the results of the statistical tests (the p_values) applied on the sequences $\{k_j\}_{j=1}^{N}$ and $\{b_j\}_{j=1}^{N}$ of length $N = 1.000.000$ using random initial values and parameters produced by our PRBG and by a pseudo random bit generator based only on the standard map, respectively.

The sequence $\{b_j\}_{j=1}^{N}$ is generated as follows: $b_j = F(a_j)$ for $1 \le j \le N$. The exact parameters values used in these examples have been included in parentheses, besides the name of the statistical test. **Table 1** shows the p_values of each test of the sequences with and without applying the ECF-map. It is clear that some tests failed if the sequence is simply generated from the standard map. However, a noticeable improvement is observed if we use standard map with the ECF-map, as all the tests are passed.

4.2. The Interpretation of Empirical Results

The interpretation of empirical results can be conducted in a number of ways. Two approaches have been adopted:

1) **The examination of the proportion of passing sequences**: in the final analysis report file generated by the NITS statistical suite, a value called the proportion was listed for each test. The proportion is the number of sequences having a p_value greater than the significance level α, divided by the total number of bit sequences tested, i.e the percentage of passed tests. NIST specifies a, range of acceptable proportions. The range is determined by using the confidence interval defined as $\hat{p} \pm 3\sqrt{\hat{p}(1-\hat{p})/m}$, where $\hat{p} = 1-\alpha$, and m is the sample size. If the proportion falls outside this interval then it is sufficient to deduce that the data is non-random. For this experiment, we generate $m = 100$ binary sequences, each containing $1.000.000$ random bits. The

Table 1. Statistical tests on the sequences $\{k_j\}_{j=1}^{N}$ and $\{b_j\}_{j=1}^{N}$ with different initial states.

Test No.	Test Name	$x_0 = 3.59587469543$ $y_0 = 0.8512974635$ $p_0 = 120.9625487136$ $N_0 = 250$		$x_0 = 5.02548745491$ $y_0 = 2.9654128766$ $p_0 = 100.6$ $N_0 = 250$	
		$\{k_j\}_{j=1}^{N}$	$\{b_j\}_{j=1}^{N}$	$\{k_j\}_{j=1}^{N}$	$\{b_j\}_{j=1}^{N}$
1	FT	0.95056	0.00000	0.57139	0.00000
2	BFT(m = 128)	0.48770	0.00499	0.60654	0.02557
3	RT	0.85244	0.00000	0.58803	0.00000
4	LROT	0.90989	0.21701	0.67662	0.41932
5	MRT	0.93152	0.40617	0.10481	0.76072
6	SPT	0.76038	0.41730	0.06727	0.01983
7	NOTMT (m = 9, B = 000000001)	0.97615	0.00407	0.28535	0.00040
8	OTMT(m = 9, B = 111111111)	0.52804	0.00034	0.50918	0.19895
9	MUST(L = 7,Q=1280)	0.18980	0.02664	0.08763	0.29615
10	LZT	0.53715	0.23431	0.06145	0.00234
11	LCT(M = 500)	0.48293	0.27597	0.68564	0.82922
12	ST(m = 16)	0.44260	0.11511	0.25245	0.95271
13	AET	0.18228	0.00000	0.78445	0.00000

14	CST(Forward)
	CST(Reverse)
		0.83761
		0.80126
			0.00000
			0.00000
				0.60651
				0.22321
					0.00000
					0.00000

15	RET($x = +1$)	0.93862	0.00000	0.40331	0.00000
16	REVT($x = -1$)	0.24142	0.00000	0.76430	0.00000

confidence interval is 0.99 ± 0.029849 and then the range of acceptable proportion is from 0.960150 to 1.019849. **Table 2** and **3** present the proportions of all tests. **Figure 1** shows the proportion for all tests. Since the proportion for each test is within the range, so we are confident to accept the sequence as random bit sequence.

2) **The examination of p value's uniformity:** The distribution of p_values is examined to ensure uniformity. For this, the interval between 0 and 1 is divided into 10 sub-intervals, and the following X^2 value for each test is calculated:

$$X^2 = \sum_{i=1}^{10} \frac{\left(F_i - \frac{m}{10}\right)^2}{\frac{m}{10}},$$

Table 2. Examination of the proportion of sequences that pass the 14 first statistical tests and the distribution of p_values.

S.NO.	Statistical Test	P_value$_T$	Proportion of sequences passing the test
1.	FT	0.739918	0.9700
2.	BFT (m = 128)	0.181557	1.0000
3.	RT	0.897763	0.9900
4.	LROT	0.275709	0.9900
5.	MRT	0.554420	0.9900
6.	SPT	0.759756	1.0000
7.	NOTMT (m = 9)		
	Template = 000000001	0.153763	0.9800
	Template = 000000011	0.319084	0.9900
	Template = 000001011	0.678686	0.9700
	Template = 000001101	0.037566	1.0000
	Template = 000100011	0.275709	0.9900
	Template = 000100101	0.616305	0.9900
	Template = 001000011	0.090936	0.9900
	Template = 001000101	0.437274	0.9900
	Template = 111100000	0.924076	1.0000
	Template = 111111110	0.534146	0.9900
8.	OTMT (Template = 111111111)	0.085587	1.0000
9.	MUST (L = 7, Q = 1280)	0.191687	0.9700
10.	LZT	0.171867	0.9800
11.	LCT (M = 500)	0.554420	0.9900
12.	ST (m = 16)	0.867692	0.9800
13.	AET (m = 10)	0.304126	0.9800
14.	CST		
	Forward	0.719747	0.9700
	Reverse	0.574903	0.9800

Table 3. Examination of the proportion of sequences that pass the 15th and 16th statistical tests and the distribution of p_values.

S.NO.	Statistical Test	P_value$_T$	Proportion of sequences passing the test
15.	RET		
	x = -4	0.534146	1.0000
	x = -3	0.455937	1.0000
	x = -2	0.739918	0.9828
	x = -1	0.122325	0.9800
	x = 1	0.171867	1.0000
	x = 2	0.911413	1.0000
	x = 3	0.534146	1.0000
	x = 4	0.262249	1.0000
16.	REVT		
	x = -9	0.075719	1.0000
	x = -8	0.534146	1.0000
	x = -7	0.045675	1.0000
	x = -6	0.122325	1.0000
	x = -5	0.383827	1.0000
	x = -4	0.574903	1.0000
	x = -3	0.045675	1.0000
	x = -2	0.122325	1.0000
	x = -1	0.419021	1.0000
	x = 1	0.236810	0.9828
	x = 2	0.455937	1.0000
	x = 3	0.066882	1.0000
	x = 4	0.262249	1.0000
	x = 5	0.816537	1.0000
	x = 6	0.004981	1.0000
	x = 7	0.383827	1.0000
	x = 8	0.534146	1.0000
	x = 9	0.020548	1.0000

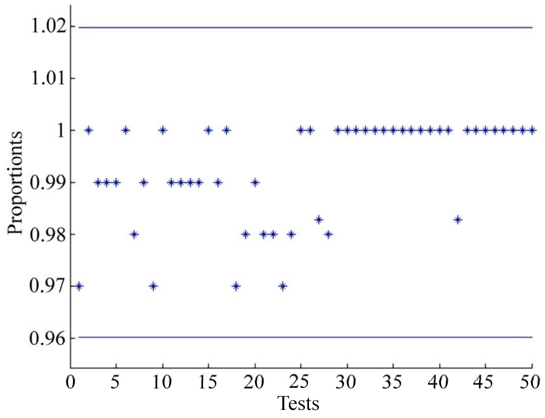

Figure 1. Proportions of the sequences passing the tests for all tests of NIST suite. The region between two horizontal lines is the acceptable range of proportion.

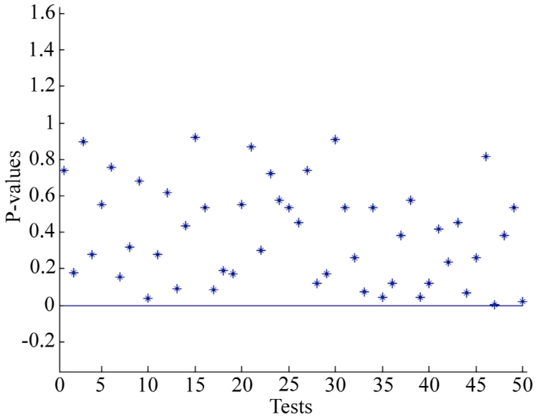

Figure 2. P_values for all tests of NIST suite. The horizontal line represents the threshold value 0.0001.

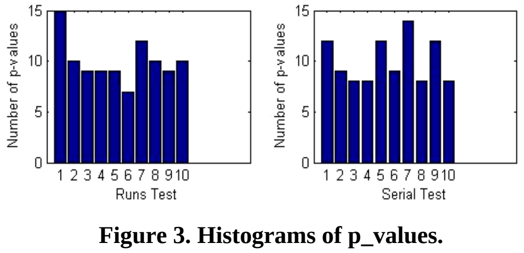

Figure 3. Histograms of p_values.

where F_i is the number of p_value's in the sub-interval i and m is the sample size. Next, a p_value of the p_value's is calculated as:

$$p\_value_T = igamc\left(\frac{X^2}{2}, \frac{9}{2}\right),$$

where $igamc$ is the incomplete gamma function, (for more detail you can refer [29,30]). If $p\_value_T \geq 0.0001$ then those p_value's can be considered uniformly distributed.

Table 2 and **3** show the p_value$_T$ of the 16 tests obtained by $m = 100$ binary sequences, each containing $1.000.000$ random bits. The sequences were generated from the proposed PRBG by using different keys. Referring to **Table 2** and **3**, it is clear that the p_values are uniformly distributed. In **Figure 2**, we have graphically presented the computed p_value$_T$ for all tests with the threshold value 0.0001. In addition, **Figure 3** shows the histograms of p_values for runs and serial tests. We can see that the p_values are uniformly distributed.

5. Conclusions

In this paper, the ECF-map has been presented and then used to design a new PRBG for stream cipher applications. This new pseudo random generator is based on the standard map with large key space and the ECF-map to generate a key stream with good cryptographic properties. The use of the ECF-map increases the randomness of the proposed PRBG. The detailed analysis done by NIST statistical test Suite demonstrates that the proposed PRGB is suitable for cryptography.

REFERENCES

[1] S. Li and X. Mou, "Improving Security of a Chaotic Encryption Approach," *Physics Letters A*, Vol. 290, No. 3-4, 2001, pp. 127-133. doi:10.1016/S0375-9601(01)00612-0

[2] X. G. Wu, H. P. Hu and B. L. Zhang, "Analyzing and Improving a Chaotic Encryption Method," *Chaos, Solitons and Fractals*, Vol. 22, No. 2, 2004, pp. 367-373. doi:10.1016/j.chaos.2004.02.009

[3] T. Yang, "A Survey of Chaotic Secure Communication Systems," *Journal of Computational Cognition*, Vol. 2, No. 2, 2004, pp. 81-130.

[4] T. Yang, L. B. Yang and C. M. Yang, "Cryptanalyzing Chaotic Secure Communications Using Return Maps," *Physics Letters, Section A*: *General, Atomic and Solid State Physics*, Vol. 245, No. 6, 1998, pp. 495-510.

[5] H. Zhou and X. T. Ling, "Problems with the Chaotic Inverse System Encryption Approach," *IEEE Circuits and Systems*, Vol. 44, No. 3, 1997, pp. 268-271. doi:10.1109/81.557386

[6] P. Li, Z. Li, W. A. Halang and G. Chen, "A Stream Cipher Based on a Spatiotemporal Chaotic System," *Chaos, Solitons and Fractals*, Vol. 32, No. 7, 2007, pp. 1867-1876. doi:10.1016/j.chaos.2005.12.021

[7] K. M. Short, "Signal Extraction from Chaotic Communications". *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, Vol. 7, No. 7, 1997, pp. 1579-1597. doi:10.1142/S0218127497001230

[8] L. Zhang, X. Liao and X. Wang, "An Image Encryption Approach Based on Chaotic Maps," *Chaos, Solitons and*

Fractals, Vol. 24, No.3, 2005, pp. 759-765. doi:10.1016/j.chaos.2004.09.035

[9] S. J. Li, X. Q. Mou and Y. L. Cai, "Pseudo-Random Bit generator Based on Couple Chaotic Systems and its Application in Stream-Ciphers Cryptography," *Progress in Cryptology-INDOCRYPT, Lecture Notes in Computer Science*, Vol. 2247, 2001, pp. 316-329.

[10] V. Patidar, N. K. Parekk and K. K. Sud, "A New Substitution-Diffusion Based Image Cipher Using Chaotic Standard and Logistic Maps," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 14, No. 7, 2009, pp. 3056-3075. doi:10.1016/j.cnsns.2008.11.005

[11] V. Patidar and K. K. Sud, "A Novel Pseudo Random Bit Generator Based on Chaotic Standard Map and its Testing," *Electronic Journal of Theoretical Physics*, Vol. 6, No. 20, 2009, pp. 327-344.

[12] X. Y. Wanga and Q. Yu, "A Block Encryption Algorithm Based on Dynamic Sequences of Multiple Chaotic Systems," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 14, No. 2, 2009, pp. 574-581. doi:10.1016/j.cnsns.2007.10.011

[13] G. Chen, Y. Mao and C. K. Chui, "A Symmetric Image Encryption Scheme Based on 3D Chaotic Cat Maps," *Chaos, Solitons and Fractals*, Vol. 21, No. 3, 2004, pp. 749-761. doi:10.1016/j.chaos.2003.12.022

[14] P. Garcia, A. Parravano, M. G. Cosenza, J. Jiménez and A. Marcano, "Coupled Map Networks as Communication Schemes," *Physical Review E*, Vol. 65, No. 4, 2002, pp. 045201.1-045201.4.

[15] H. Li and J. Zhang, "A Secure and Efficient Entropy Coding Based on Arithmetic Coding," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 14 No. 12, 2009, pp. 4304-4318. doi:10.1016/j.cnsns.2009.03.003

[16] K. Fallahi, R. Raoufi and H. Khoshbin, "An Application of Chen System for Secure Chaotic Communication Based on Extended Kalman Filter and Multi-Shift Cipher Algorithm," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 13, No. 4, 2008, pp. 763-781. doi:10.1016/j.cnsns.2006.07.006

[17] B. Mi, X. Liao and Y. Chen, "A Novel Chaotic Encryption Scheme Based on Arithmetic Coding," *Chaos, Solitons and Fractals*, Vol. 38, No. 5, 2008, pp. 1523-1531, 2008.

[18] A. Kanso and N. Smaoui, "Logistic Chaotic Maps for Binary Numbers Generations," *Chaos, Solitons and Fractals*, Vol. 40, No. 5, 2009, pp. 2557-2568. doi: 10.1016/j.chaos.2007.10.049

[19] G. Alvarez, F. Montoya, M. Romera and G. Pastor, "Cryptanalysis of a Discrete Chaotic Cryptosystem Using External Key," *Physics Letters A*, Vol. 319, No. 3-4, 2003, pp. 334-339. doi:10.1016/j.physleta.2003.10.044

[20] G. A. Alvarez and L. B. Shujun, "Cryptanalyzing a Nonlinear Chaotic Algorithm (NCA) for Image Encryption," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 14, No. 11, 2009, pp. 3743-3749. doi:10.1016/j.cnsns.2009.02.033

[21] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, "A Statistical Test Suite for Random and Pseudo Random Number Generators for Cryptographic Applications," National Institute of Standards and Technology Special Publication 800-22, 2001.

[22] G. Marsaglia, "DIEHARD: A Battery of Tests of Randomness," 1997. http://stat.fsu.edu/geo/diehard.html.

[23] L. Lorentzen and H. Waadeland, "Continued Fractions with Applications," North Holland, Amsterdam, 1992.

[24] R. B. Seidensticker, "Continued Fractions for High-Speed and High-Accuracy Computer Arithmetic," *Proceedings of the 6th IEEE Symposium on Computer Arithmetic*, 1983, pp. 184-193.

[25] J. Vuillemin, "Exact Real Computer Arithmetic with Continued Fractions," INRIA Report 760, Le Chesnay, 1987.

[26] H. S. Wall. "Analytic Theory of Continued Fractions". Chelsea, 1973.

[27] Y. Hartono, C. Kraaikamp and F. Schweiger, "Algebraic and Ergodic Properties of a New Continued Fraction Algorithm with Non-Decreasing Partial Quotients," *Journal de théorie des nombres de 9 Bordeaux*, Vol. 14, No. 2, 2002, pp. 497-516.

[28] X. Di, X. Liao and P. Wei, "Analysis and Improvement of a Chaos-Based Image Encryption Algorithm," *Chaos, Solitions and Fractals*, Vol. 40, No. 5, 2009, pp. 2191-2199. doi:10.1016/j.chaos.2007.10.009

[29] W. H. Press. "Numerical Recipes in C: The Art of Scientific Computing," Cambridge University Press, Cambridge, 1992, pp. 169-173.

[30] M. Abramowitz and I. Stegun, "Handbook of Mathematical Functions," *Applied Mathematics Series*, Vol. 55, 1968, pp. 2286-2293.