Applied Mathematics, **2011, 2, 1531-1534**
doi:10.4236/am.2011.212217 Published Online December 2011 (http://www.SciRP.org/journal/am)

Cryptographic PRNG Based on Combination of LFSR and Chaotic Logistic Map

Hamed Rahimov, Majid Babaei, Mohsen Farhadi
Department of Computer Engineering, Shahrood University of Technology, Shahrood, Iran
E-mail: *hrahimov@shahroodut.ac.ir*, *babaee@comp.tus.ac.ir*, *mfarhadi@shahroodut.ac.ir*
Received September 6, 2011; *revised October* 21, 2011; *accepted October* 30, 2011

Abstract

The random sequence generated by linear feedback shift register can't meet the demand of unpredictability for secure paradigms. A combination logistic chaotic equation improves the linear property of LFSR and constructs a novel random sequence generator with longer period and complex architecture. We present the detailed result of the statistical testing on generated bit sequences, done by very strict tests of randomness: the NIST suite tests, to detect the specific characteristic expected of truly random sequences. The results of NIST's statistical tests show that our proposed method for generating random numbers has more efficient performance.

Keywords: Random Bit Generator, Combination, Chaotic, Logistic Equations, LFSR, NIST Suite Tests

1. Introduction

With perfect method of cryptographic algorithm based on generating high performance of random numbers, the need for random numbers of high quality is growing [1, 2]. Random numbers are also used for key generation in symmetric. For example, in the Smart cards, the main problem is the security which improves by using reliable random number generators (RNGs) [3].

One of the most reliable methods that work as RNG in cryptographic algorithms is Linear Feedback Shift Register (LFSR) [4-6] in stream cipher usage and is appropriate to low power or high speed application [5]. Since LFSRs are linear systems. They lead to responsible easy cryptanalysis tools. The short period of LFSR's outputs is the negative point of using this system as a reliable method in cryptographic algorithms.

LFSR is a shift registered the inputs of which are based on linear function of its previous states. It uses two linear operators Exclusive-OR ($\oplus$) [6]. The LFSR is initializing by the random string that is called the seed and since the operation of register is deterministic, the result's string which is produced by the LFSR is completely determined (*i.e*. the current bit determined by its previous states) [4,7]. If the LFSR has the n bits the biggest number for internal state is 2^n so to gain the longest period of this method (*i.e*. 2^{n-1}), LFSR with length n needs to find n exponent primitive polynomial [4,7]. The architecture is shown in **Figure 1**.

However, the production of n exponent primitive polynomials becomes more difficult with the increment of n. Generally, through 2^{n-1} factorization, we can make sure that an n exponent polynomial is a primitive polynomial. Furthermore, the items of n exponent primitive polynomial are more complex to make a breakthrough. If the exponent of primitive polynomial is bigger, LFSR will be longer.

Reese defined a genetic algorithm for optimization problem. That parent population was generated by random number generator, pseudo random number generator, quasi random number generator [7]. Finally she ranked these generators with comparing precisions generated answers by GA and showed the genetic algorithm is a criterion to realize what initial population is more uniformity [8]. Also chaotic random number generator (CRNGs) was compared to other RNGs with this method [9]. In this Paper we used another famous test method: NIST suite test that is a statistical package comprising of 15 tests.

2. Chaotic Logistic Equation

The simple modified mathematical from of the logistic equation is given as:

$$f(x_n) = x_{n+1} = r \cdot x_n \cdot (1 - x_n) \quad (1)$$

where x_n is the state variable, which lies in the interval

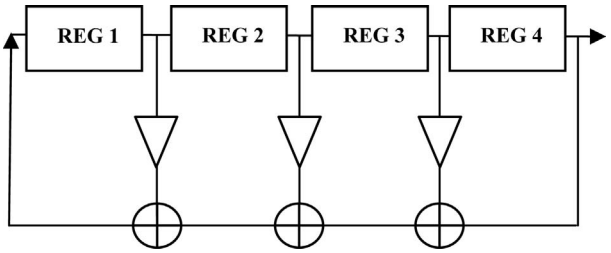

Figure 1. Linear Feedback Shift Register.

$[0\cdots1]$ and r is system parameter which can have any value between $[1\cdots4]$.

In **Figure 2**, we have displayed the Lyapunov exponent which is the quantities measure of chaos as a function of system parameter r. A positive Lyapunov exponent (for example at r = 3.99) indicates chaotic behavior.

This paper proposes a random bit generator, which is based on chaotic LFSR; the chaotic system starting from random independent initial conditions $x_0, y_0 \in (0,1)$ and $x_0 \neq y_0$.

Based on this theorem, in the next section we describe the new random number generator which can be used in the cryptology algorithms.

3. Proposed RNG

In this section, we describe the proposed method for generating random numbers, so based on previous sections in our method; in the first iteration it generated two random numbers (e.g. 0.9917 and 0.2375) with logistic chaotic equations. Then it defines a function with Equation (2):

$$G(n) = \sum_{n=0}^{M} (x_n + y_n) \mod 1 \qquad (2)$$

where $G:[0\cdots1]\to[0\cdots1]$ so $G(n)$ is the decimal parts of these numbers and $M \in \mathbb{N}$.

L.Y. Deng *et al*. have proved that combination generator should improve upon the uniformity as well as the independence over individual generators [10]. Knowing that for any real number x, $x \bmod 1 = x - [x]$, where $[x]$ is the generated integer $\leq x$.

In the next level of the proposed method, we define a function $f(x)$ which can compare x value with a threshold, the best threshold for interval $[0\cdots1]$ is 0.5. So according to **Figure 3**, if x is the larger than 0.5 number 1 *XOR* with LFSR output, else number 0 *XOR* with LFSR output. Linear correlation in LFSR outputs decreases with this technique. According to the NIST suite test the number of binary sequences at least should be 2000 which the length of each sequence is 10^6 bits so there is no experimental way to generate a valid sequence with the efficient period based on LFSR method. Our proposed RNG method prepares the external bit of

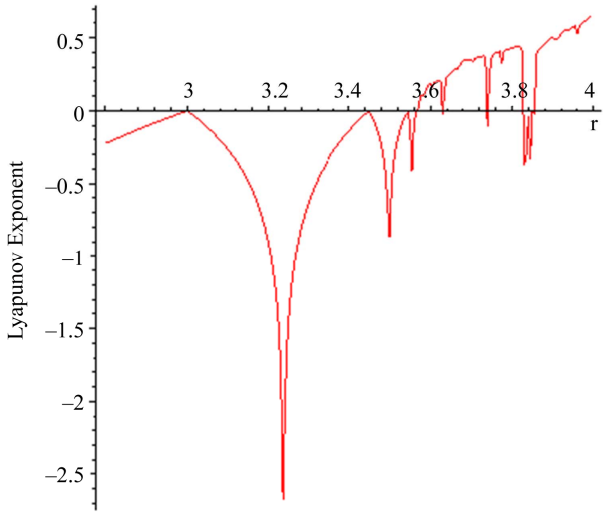

Figure 2. Plot of the lyapunov exponent versus *r* for logistic equation.

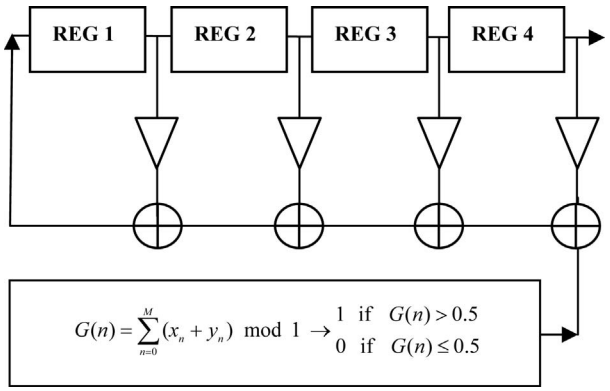

Figure 3. Chaotic liner feedback shift register diagram.

the chaotic logistic map that changes the system condition and generates the sequence of bits with no period.

4. Test Results

In the last part of our paper before conclusion, we will refer to the NIST suite test in the **Tables 1-4**; the first column in these tables is the test's names; the second column is the P-value that is in the interval $[0\cdots1]$, the test has the better condition (*i.e.* it's more reliable) when it's P-value tend to 1; and the third column shows the pass rate of these tests (*i.e.* how proportion of the sequences that tested by this method, pass the test). Considering these facts, we have 15 tests in **Table 1** that describe the general condition of the sequences which is produced by our proposed method and the other Tables show the proposed method condition with more detail. That is, non-overlapping templates in **Table 2**, the purpose of this test is to detect the number of occurrences of specific string. Random excursions test is shown in

Table 1. NIST suite tests table.

Test	P-Value	Pass Rate
Frequency	0.757790	0.9900
Block-Frequency	0.742917	0.9915
CuSums-forward	0.953553	0.9895
CuSums-backward	0.912069	0.9885
Rans	0.302657	0.9950
Long run	0.471146	0.9860
Rank	0.363593	0.9920
FFT	0.000159	0.9950
Overlapping templates	0.422638	0.9885
Universal	0.349676	0.9905
Approximate entropy	0.669359	0.9915
Serial 1	0.301194	0.9910
Serial 2	0.406499	0.9945
Linear complexity	0.125200	0.9920

Table 2. Non-overlapping templates.

Test	P-value	Pass Rate
Template = 000000001	0.069863	0.9895
Template = 000100111	0.292519	0.9895
Template = 001010011	0.028529	0.9935
Template = 010001011	0.342451	0.9880
Template = 011101111	0.515118	0.9910
Template = 101101000	0.845490	0.9935
Template = 110100100	0.786830	0.9900
Template = 111100000	0.892036	0.9885

Table 3. Random excursions test.

Test	P-value	Pass rate
X = –4	0.381162	0.9902
X = –3	0.379765	0.9878
X = –2	0.863888	0.9869
X = –1	0.464167	0.9918
X = +1	0.426003	0.9878
X = +2	0.700827	0.9894
X = +3	0.757015	0.9869
X = +4	0.899722	0.9878

Table 4. Random excursions variant test.

Test	P-value	Pass rate
X = –9	0.891536	0.9918
X = –8	0.510380	0.9918
X = –7	0.677158	0.9927
X = –6	0.710898	0.9927
X = –5	0.933972	0.9910
X = –4	0.651625	0.9869
X = –3	0.231937	0.9853
X = –2	0.113023	0.9927
X = –1	0.465727	0.9902
X = +1	0.518531	0.9910
X = +2	0.396742	0.9894
X = +3	0.908694	0.9878
X = +4	0.602174	0.9927
X = +5	0.651625	0.9886
X = +6	0.200238	0.9861
X = +7	0.445638	0.9869
X = +8	0.658445	0.9878
X = +9	0.326496	0.9878

Table 3; this test shows is the number of cycles having exactly K visits in a cumulative sum random walk. **Table 4** describes random excursions variant test, the focus of this test is the total number of times that the particular state is visited [11].

5. Conclusions

We have proposed a design of a pseudo random bit generator (PRBG) based on combination chaotic logistic equations and LFSR method. That chaotic system iterated independently starting from independent initial conditions. The pseudo random bit sequence is obtained by combining the outputs of both the chaotic logistic equations with LFSR method. We have also tested rigorously the generated sequences using the NIST suite tests. The results of statistical testing are encouraging and show that the proposed PRBG has perfect cryptographic properties and hence can be used in the design of new stream ciphers.

6. References

[1] R. C. Fairfield, R. L. Mortenson and K. B. Coulthart, "An LSI Random Number Generator (RNG)," *Advances in Cryptography*: *Proceeding of Cryptography* 84, Vol. 196, 1984, pp. 203-230.

[2] S. Callegari, R. Rovatti and G. Setti, "Embeddable ADC-Based True Random Number Generator for Cryptogra-

phic Applications Exploiting Nonlinear Signal Processing and Chaos," *Signal Processing IEEE Transactions*, Vol. 53, No. 2, 2005, pp. 793-805. doi:10.1109/TSP.2004.839924

[3] K. Tsoi, K. Leung and P. Leong, "Compact FPGA-Based True and Pseudo Random Number Generators," *Proceedings of IEEE Symposium on Field-Programmable Custom Computing Machines* (*FCCM*), 9-11 April 2003, pp. 51-61.

[4] H. Zhang, Y. Wang, B. Wang and X. Wu, "Evolutionary Random Sequence Generators Based on LFSR," *Wuhan University Journal of Natural Sciences*, Vol. 12, No. 1, 2007, pp. 75-78.

[5] T. Stojanovski, J. Pil and L. Kocarev, "Chaos-Based Random Number Generators. Part II: Practical Realization," *IEEE Transactions on Circuits and Systems I*: *Fundamental Theory and Applications*, Vol. 48, No. 3, 2001, pp. 382-385. doi:10.1109/81.915396

[6] M. Sharaf, H. A. K. Mansour and H. H. Zayed, "A Complex Linear Feedback Shift Register Design for the A5 Key Stream Generator," *Proceedings of the Twenty-Second National Radio Science Conference* (*NRSC* 2005), Cairo, 15-17 March 2005, pp. 395-402. doi:10.1109/NRSC.2005.194024

[7] L. C. Reese, W. M. Isenhower and S.-T. Wang. "Analysis and Design of Shallow and Deep Foundations," John Wiley & Sons, New York, 2006.

[8] L. Y. Deng and Y. C. Chu, "Combining Random Number Generators," *Proceedings of the* 23*rd Conference on Winter Simulation*, 1991, pp. 1043-1046.

[9] A. Reese, "Random Number Generators in Genetic Algorithms for Unconstrained and Constrained Optimization," *Nonlinear Analysis*, Vol. 71, 2009, pp. 679-692. doi:10.1016/j.na.2008.11.084

[10] M. Babaei and M. Ramyar, "Improved Performance of LFSR's System with Discrete Chaotic Iterations," *Word Applied Science Journal* (*ISI*), Vol. 13, No. 7, 2011, pp. 1720-1725.

[11] NIST Special Publication 800-22, Statistical Test Suite for Random and Pseudo Random Number Generators for Cryptographic Applications, Available at http://www.nist.gov

