World Applied Sciences Journal 13 (7): 1720-1725, 2011
ISSN 1818-4952
© IDOSI Publications, 2011

Improved Performance of LFSR's System with Discrete Chaotic Iterations

Majid Babaei and Mahboobe Ramyar

Computer Engineering, Shahrood University of Technology (SUT), Shahrood, Iran

Abstract: Linear Feedback Shift Registers (LFSRs) are considered powerful methods for generating pseudo-random bits in cryptography algorithm applications. In this paper it is shown that the linear dependencies in the generated random bit sequences can be controlled by adding a chaotic logistic map to the LFSR's systems. The structure of the LFSR's output sequence in combination with a chaotic map is analyzed and proved to have at least as much uniformity than the corresponding set for the linear components individually. In order to understand that using the proposed PRBG is reliable in secure algorithms, the NIST suite test have been taken on the proposed method, finally to compare the proposed PRNG output sequence features with the two types of LFSRs (Fibonacci and Galois).

Key words: Linear Feedback Shift Register · Random Number · Chaotic Map · NIST

INTRODUCTION

In the modern world of computers, network security is the main concern which relies on the use of cryptography algorithms. high quality random number generation is a basic subject of cryptography algorithms and the importance of a secure random number generator design cannot be underestimated. Most common generation techniques about RNGs involve truly random and pseudorandom number generators. For a brief introduction in various types of RNGs:

Truly Random Number Generators (RNGs) is a computer algorithm, which generates a sequence of statistically independent random numbers. Actually these generators require a naturally occurring source of randomness phenomena (i.e. as a non-deterministic system). Most practical implementations design a hardware device or a software program based on RNGs to produce a bit sequence which is statistically independent [1].

Pseudo-Random Bit Generators (PRBGs) are implemented by an algorithm that is actually a finite state machine; reliable RNGs which are implemented by these methods should pass several statistical tests to prove their usefulness [2-4].

With the mention of these points, the security of the entire cryptographic system such as RSA and DES and the other secure algorithms relies on the randomness quality of the generator [5, 6]. PRNGs are based on the algorithmic function, so the outputs of these methods are not truly random.

In the last two decades several works in this area have been implemented based on chaotic systems [7].

Chaotic system is a natural phenomenon that behaves chaotic in the specific system's parameters [8]. Chaotic maps are sensitive to initial conditions; this makes them sensitive to minimal change of information from the input thus heavily varying the output when input sequence changes by the minute. Chaotic maps compute quickly in the regular machine and are able to create sequences with extremely long cycle lengths [9].

Linear feedback shift register (LFSR) is a shift register which is able to generate random bits (with the mention of amount of registers [10]). In the LFSR input bit is a linear function (i.e. it's an exclusive-or function) of its previous state. It's a shift register which input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value.

The initial value of the LFSR is called the *seed*, LFSR's operation is deterministic and so the stream of values produced by the register is completely determined by its algorithm and current (or previous) state.

The theory of the Linear Feedback Shift Registers (LFSRs) is based on the polynomial form, so in the blow equation *p* and *q* are the binary digits:

$$generator(LFSR) = X^p = X^q + 1 \quad (1)$$

In this paper we design a new random number generator by using a LFSR generator with a combination of logistic chaotic maps. [10]. The proposed random bit generator is based on a combination of logistic chaotic

Corresponding Author: Majid Babaei, Computer Engineering, Shahrood University of Technology (SUT), Shahrood, Iran.

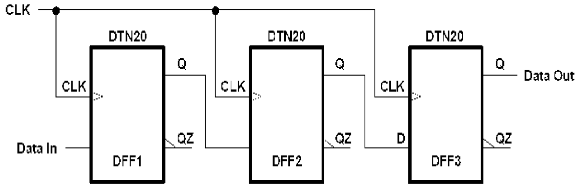

Fig. 1: Three-Bit Shift Register [28]

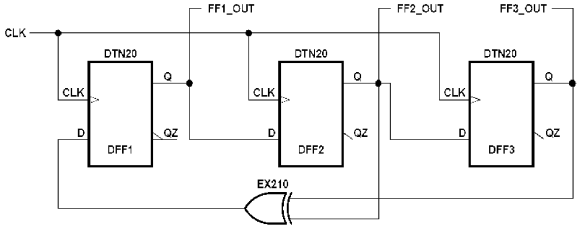

Fig. 2: Liner Feedback Shift Register [28]

maps as a chaotic system in the LFSR algorithm, which of course increases the complexity in output sequence of the LFSR and becomes difficult for an intruder to extract information about the cryptography system. In the next section, we briefly introduce the LFSR algorithm, which is a basic building block of the proposed pseudo random bit generator.

Linear Feedback Shift Register: The LFSR is a shift register that when the signal generator is clocked, each register generates a signal based on the previous registers (see Figure 1). In this system some of the outputs are combined in an exclusive-or arrangement of elements to form a feedback mechanism. A LFSR can be designed by performing a XOR's function on the output of the registers. Figure 2 shows a LFSR based on three D-flip flops which are clocked synchronically.

Based on these theoretical points, the condition of the best performance in the LFSRs occurs when the outputs of the D-flip flops are loaded with a random seed value so the linear feedback shift registers make very good pseudorandom bit generators, it will be able to generate pseudorandom bit sequence of 1s and 0s.

The LFSR generators are defined by the mathematical model:

$$x_n = (a_1x_{n-1} +...+a_kx_{n-k}) \bmod 2 \tag{2}$$

$$u_n = \sum_{i=1}^{w} x_{ns+i-1}2^{-i} \tag{3}$$

For an example of positive integer's s and w, See Refs. [11–13] for more details. The result is that the maximal period length in the specific LFSR's system with the n registers is $\rho=2^n-1$. The longest period in the same LFSR's systems should be found an n as exponent primitive polynomial.

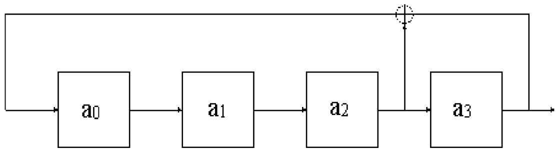

Fig. 3: Galois LFSR setup

In the security field, the period length is very important because it makes the sequence unpredictable. However, the basic solution for this problem is to produce an n exponent primitive polynomial (i.e. n is the large number and reliable sequence length in secure algorithm) with the increased number of D-flip flops. This idea may not work very well, because the number of the registers is limited.

The LFSRs are split into two family devices called the *Fibonacci* and the *Galois* representations. In the two next subsections we introduce the Fibonacci LFSR and the Galois LFSR [12-15].

Galois LFSRs: The *Galois* is presented in the blow equations:

$$a = a_{k-1}x^{k-1}+...+a_1x^1 + a_0 \tag{4}$$

So the value of a_{k-1} results:

$$(a-a_{k-2}x^{k-1}...-a_1x^1-a_0)/x^{k-1} = a_{k-1} \tag{5}$$

And the product xa is given that simply have been resulted by multiplication of x in the equation (5):

$$xa = a_{k-1}x^k + a_{k-2}x^{k-1}+...+a_1x^2 + a_0x \tag{6}$$

Thus as x is the root of the last equation, we obtain:

$$xa = (a_{k-1}s_{k-1} + a_{k-2})x^{k-1}+...+a_{k-1} \tag{7}$$

The above equation is the main description of the Galois device feedback computation. In Figure 3, a Galois device is represented as:

$$P(X) = X^4 + X + 1 \tag{8}$$

Fibonacci LFSRs: The *Fibonacci* implementation is the simple shift register which is given by the Fibonacci representation. Let the value of $a^{'}$ expressed in the following equation:

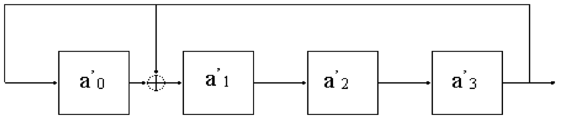

Fig. 4: Fibonacci LFSR setup

$$a = a'_{k-1}y_{k-1} + ... + a'_1 y_1 + a_0 y_0 \tag{9}$$

With the mention of the transposed function the value of a' would be:

$$a'_j = Tr(ax^j) \qquad j = 0,1,...,k-1 \tag{10}$$

So the value of the xa is obtained in the fallowing equation by the replace of a as xa function which are given by:

$$(xa)'_j = Tr(ax^{j+1}) \tag{11}$$

Finally:

$$\begin{aligned} &(xa)'_j = a'_{j+1} \qquad j = 0,1,...,k-2 \\ &(xa)'_{k-1} = Tr(ax^k) = s_{k-1}a'_{k-1} + ... + a'_0 \end{aligned} \tag{12}$$

The two last equations are the best descriptions of the main part of a Fibonacci LFSR's systems. In Figure 4, it's represented as:

$$P(X) = X^4 + X + 1 \tag{13}$$

The properties of LFSR have been deeply studied in Ref. [16-18]. In this paper we will improve the period lengths in these LFSRs generators by adding a chaotic system in a part of the LFSR's algorithm and create a novel Chaotic Linear Feedback Shift Register (CLFSR).

In the next section we are going to describe the importance of chaotic maps in cryptography function.

Chaotic Logistic Map: The concept of the chaotic behaviors is related to the positive value of *Lyapunov exponents*. It is described by following assumptions:

Let $s \in S$ and υ be an independent element of tangent space at s and the value of $DF^n(s)(\upsilon)$ mentioned of the n-iteration of F at s in the direction of υ. So the Lyapunov exponent is given by the specific limited in the equation 14:

$$\lambda_{s,\upsilon} \equiv \lim_{n\to\infty} \frac{1}{n} \ln \left\| DF^n(s)(\upsilon) \right\| \tag{14}$$

The dynamic system presented by F when $F{=}S{\to}S$ where S is the state space, so the dynamic system has the chaotic behavior if only the value of the Lyapunov exponent in the specific system parameter is positive [16-19].

These chaotic behavior shown by the simple mathematical model which is used to describe the growth of biological populations are used in the initial population of GA.

The mathematical form of the *chaotic logistic map* is given as:

$$f(x_n) = x_{n+1} = r.x_n.(1{-}x_n) \tag{15}$$

Where x_n is the state variable, which lies in the interval [0..1] and r is called system parameter which can have any value between [1..4].

In the next section a novel combination of the LFSR's system will be described and the chaotic map will be proved by the following presented theorem.

Combining LFSR with Chaotic Map: Now let us express t_1 as a LFSR, t_2 as a chaotic logistic map and the output function as:

$$t_i = t_{1,I} \oplus t_{2,t} \tag{16}$$

The sign of ⊕ represents the operator XOR on the binary sequence of $t_{1,t}$ and $t_{2,t}$. The bit sequence of selection (i.e. the B set with k' bit size), so observed in the expansion of t_0, the string of bit formed by concatenating the bits $b_{0,1}, b_{0,2}..., b_{0,s0}$, the string of bits $b_{1,1}, b_{1,2}, ..., b_{1,s_1}$, in the expansion of t_1 and by this order in the *(i-1)^th^* level (i.e. in the expansion of $t_{i\text{-}1}$) the string of bits $b_{i-1,1}, b_{i-1,2}, ..., b_{i-1,s_{i-1}}$ where:

$$s_0 + ... + s_{i-1} = \sum_{i=1}^{n} k'_{i-1} \tag{17}$$

It is assumed that k'_i is the size of B_i (i^{th} bit selection). Defining P_j as an independent parameter of the j^{th} set, the value of $\theta_{j,B}$ is the corresponding sets of bit strings, it means that the value of the θ_B is the corresponding sets of bit string for P.

By mentioning these assumption the random bit generator is *B-equidistributed* if θ_B is equidistributed [20-23].

For non-liner generators, the uniformity of t_i is often evaluated by discrepancy bounds [24], it's an average over an entire family of generators. Certain types of non-linear generators (like chaotic maps) tend to perform better than the linear ones in statistical tests [25].

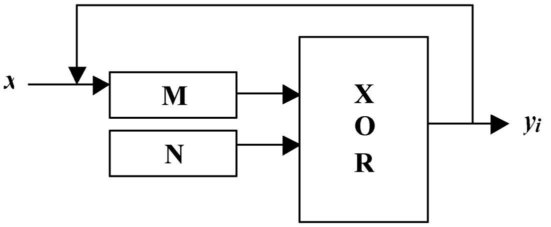

Fig. 5: Chaotic system combined with a LFSR system

Figure 5 shows how a LFSR (M-box) combines with a chaotic system (N-box). So with this complete definition, we now have a theorem:

Theorem 1: *If B is the bit selection of a sequence generated by t_1 (output of a LFSR exclusive-or chaotic map) and P_1 is a B-equidistribution for B of size k', then $\theta_{1,B}\oplus$ ω is equidistributed (where ω is a bit vector of sizek') for any ω and P is also B-equidistributed.*

Proof: The fundamental aspect for proving this theorem is based on the reason that in the sample sequence if $x = x'$ then $x \oplus \omega = x' \oplus \omega$ and if $x \oplus \omega = x' \oplus \omega$ then $x=x'$(i.e.$x \oplus \omega = x' \oplus \omega$ if only if $x=x'$). On the other hand, it is clear that:

$\theta_{1,B} \oplus \omega = \bigcup_{x \in \theta_{1,B}} (x \oplus \omega)$, it means that the distribution of the output sequence that is generated by LFSR *XOR* chaotic logistic map is equal to union of LFSR's distribution and chaotic logistic map's distribution).

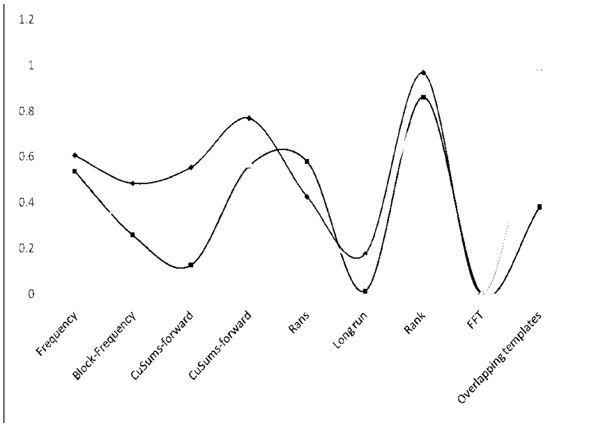

Fig. 6: NIST test result (Red is the Proposed PRNG, Blue represents Galois and Green is Fibonacci)

It is equidistributed if and only if $\theta_{1,B}$ is equidistributed. With the mentioning of these facts the unions of equidistributed sets generate equidistributed set, so $\theta_B = \bigcup_{\omega \in \theta_{2,B}} (\theta_{1,B} \oplus \omega)$ is equidistributed.

Statistical Testing: The new method for generating secure random numbers is evaluated by the NIST test suite which is a theoretical analysis and experiment program.

NIST Statistical Test Suite: The NIST tests suite is a statistical package involving 15 tests which are based on

Table 1: NIST test results

	P-value			Pass rate		
Tests	*proposed PRNG*	*Fibonacci*	*Galois*	*proposed PRNG*	*Fibonacci*	*Galois*
Frequency	0.606499	0.535558	0.269087	0.9930	0.9900	0.9867
Block-Frequency	0.483676	0.256881	0.390767	0.9925	0.9785	0.9550
CuSums-forward	0.553505	0.125567	0.389001	0.9985	0.9815	0.9900
CuSums-backward	0.769260	0.558502	0.568710	0.9900	0.9805	0.9900
Rans	0.425020	0.578382	0.369001	0.9915	0.9910	0.9810
Long run	0.174249	0.012343	0.155672	0.9910	0.9895	0.9915
Rank	0.967341	0.859903	0.790510	0.9915	0.9825	0.9815
FFT	0.000159	0.000159	0.000159	0.9950	0.9950	0.9860
Overlapping templates	0.977566	0.379555	0.977301	0.9895	0.9805	0.9800

Number of binary sequences tested (m) = 2000

Length of each binary sequences = 1000000

Significant level (α) = 0.01

The range of acceptable proportion is [0.9833245, 0,9966745]

Null hypothesis (H_0): The binary sequence is random.

If p-value ≥ α (0.01) then the null hypothesis is accepted.

If p-value ≤ α (0.01) then the null hypothesis is rejected.

If p-value$_T$ ≥ 0.0001 then the p-values can be considered to be uniformly distributed hypothesis testing. Also The NIST tests suite focuses on a variety of different types of non-randomness. These tests focus on a variety of different types of non-randomness that could occur in the sequence [28].

Experiment Results: In our test, m is the sample size and $\hat{p} = 1 - \alpha$ in which m = 2000 and $\hat{p}$ = 1- 0.01 = 0.99 for the present analysis. So the range of an acceptable proportion is [0.9833245, 0.9966745]. The quantitative results of the proportions are given in Table 1 for various statistical tests of the NIST suite.

Accepted p-value in the NIST suite test with the mentioned initial values, should be in interval [0.9833245, 0.9966745]; so the p-values of our purposed method is in this interval and then the 15 tests of the NIST suite have been passed as shown In Fig. 6.

CONCLUSION

In this paper we presented a novel method to generate random bit sequence by combination of LFSR's system and chaotic logistic map and it has been proved in a reliable theorem. At the end, we compared it with the same other methods such as Fibonacci LFSR and Galois LFSR and the result was shown in table 1.

REFERENCES

1. Menezes, A.J., P.C.V. Oorschot and A.S. Vanstone, 1997. *Hand book of Applied Cryptography.* CRC Press, Boca Raton.
2. NIST Special publication 800-22: A Statistical test suite for the validation of the random number generators and Pseudo random number generators for Cryptographic Applications (2001).
3. Marsaglia, G., 1995. The diehard test suite (1995), http://stat.fsu.edu/geo/diehared.html
4. Knuth, D.E., 1997. *The Art of computer programming,* 3rd edn. Seminumerical Algorithm, vol. 2. Addison-wesley Longman Publishing Co., Massachusetts.
5. Anderson, R., 2001. Security engineering, *A Guide to Build Dependable Distributed Systems.* John Willey and sons Inc., New York.
6. Patidar, V., K.K. sud and N.K. Pareek, 2009. A Pseudo Random Bit Generator Based on Chaotic Logistic map and its Statistical testing, pp: 441-452.
7. Li, S., 2003. *Analysis and new design of Digital Chaotic Ciphers,* PhD thesis, School of Electronic and Information Engineering, Xi'an Jiaotong University.
8. Protopopescu, V.A., R.T. Santoro and J.S. Tollover, 1995. *Fast secure encryption decryption method based on chaotic dynamics.* US Patent, pp: 5479513.
9. Kocarev, L. and G. Jakimoski, 2003. *Pseudo random bits generated by chaotic maps.* IEEE Transactions on Circuits and systems I: Fundamental Theory and Applications, 50(1): 123-126.
10. Li, S., X. Mou and Y. Cai, 2001. *Pseudo random bit generator based on couple chaotic systems and its application in stream ciphers cryptography.* In Progress in cryptography – INDOCRYPT 2001, Lecture Notes in Computer Science, 2247: 316-329.
11. Lidl, R. and H. Niederreiter, 1983. *Finite Fields* (*Encyclopedia of Mathematics vol. 20*), Cambridge University Press, Cambridge.
12. McEliece, R., 1987. *Finite Fields for Computer Scientists and Engineers,* Kluwer Academic Publishers, Boston.
13. Noras, J., 1997. *Fast pseudorandom sequence generators: linear feedback shift registers, cellular automata and carry feedback shift registers.* Univ. of Bradford Electrical Engineering Department Report, pp: 594.
14. Klapper, A. and M. Goresky, 1997. *Feedback shift registers, 2-adic span and combiners with memory,* J. Crypt., 10: 111-147.
15. Golomb, S., 1982. *Shift Register Sequences.* Aegean Park Press, Laguna Hills CA.
16. McEliece, R.J., 1987. Finite _eld for scientists and engineers. Kluwer Academic Publishers.
17. Golomb, S.W.,1981. *Shift Register Sequences.* Aegean Park Press.
18. Lidl, R. and H. Niederreiter, 1997. *Finite Fields*, 2nd ed. Cambridge University Press.
19. Guckenheimer, J. and P. Holmes, 1983. *Nonlinear oscillations, dynamical systems and bifurcations of vector fields,* Springer-Verlag, New York.
20. May, R.M., 1976. *Simple mathematical models with very complicated dynamics.* Nature, 261: 459-467.
21. L'Ecuyer, P., 1996. *Maximally equidistributed combined Tausworthe generator*, Math. Comput., 65(213): 203-213.
22. L'Ecuyer, P. and C. Lemieux, 2000. *Variance reduction via lattice rules, Manage.* Sci., 46(9): 1214-1235.
23. L'Ecuyer, P. and F. Panneton, 2000. *Construction of equidistributed generators based on linear recurrences modulo 2*, in: K.T. Fang, F.J. Hickernell, H. Niederreiter (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*, Springer, Berlin, pp: 318-330.

24. L'Ecuyer, P. and P. Hellekalek, 1998. Random *number generators: selection criteria and testing*, in: P. Hellekalek, G. Larcher (Eds.), Random and Quasi-Random Point Sets, Lecture Notes in Statistics, vol. 138, Springer, New York, pp: 223-265.
25. Niederreiter, H., 1992. *Random number generation and quasi-Monte Carlo methods*, in: Proceedings of the SIAM CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 63, SIAM, Philadelphia.
26. Goresky, M. and A.M. Klapper, 1997. *Arithmetic crosscorrelation of feedback with carry shift regiseter sequences*, *IEEE Trans. Inform. Theory*, 43: 1342-1346.
27. Peterson, W.W., 1960. *Encoding and error-correction procedure for the Bose–Chaudhuri codes*, *IRE Trans. Inform. Theory*, IT-6: 459-470.
28. Noras, J., 1997. *Fast pseudorandom sequence generators: Linear feedback shift registers, cellular automata and carry feedback shift registers*, Univ. Bradford Elec. Eng. Dept., Bradford, U.K., Rep., pp: 94.