Proceeding of the International Conference on Advanced Science, Engineering and Information Technology 2011

Hotel Equatorial Bangi-Putrajaya, Malaysia, 14 - 15 January 2011

ISBN 978-983-42366-4-9

Study on the Effects of Characteristic Polynomial in LFSR for Randomness Quality

Meilana Siswanto, Gunawan Witjaksono, Moesfa Soeheila, Zharfan Hamdan

Information Security Lab, MIMOS Berhad

Technology Park Malaysia, Kuala Lumpur, 57000, Malaysia
Tel.: +60(03)8995 5000 ext. 5119, E-mail: meilana.siswanto@mimos.my

***Abstract* – Randomness quality of keys becomes an essential in secure communications, since the security of modern cryptographic techniques relies on unpredictable and irreproducible digital keys which are generated by random number generator (RNG). This study focuses on the effects of characteristic polynomial in linear feedback shift registers (LFSR) for randomness quality. RNG's output is produced by integrating binary random source based on optic and LFSR. In this observation, randomness of the RNG's output with different characteristic polynomials has been tested using National Institute of Standards & Technology (NIST) test. The result shows that RNG with LFSR which is characterized by a feedback being a primitive polynomial of *n*-1 passes all the NIST-standard statistical tests.**

***Keywords* - LFSR effects in randomness, randomness quality, RNG.**

I. INTRODUCTION

Key with its randomness quality has become an essential in cryptographic systems wherein security level of an encryption system relies on unpredictable and irreproducible keys generated by RNG [1]. Therefore many attempts have been proposed to realize truly random numbers as replacing pseudo RNG (PRNG) which produces un-truly random since having pattern and repetitive occurrence at a certain time period. However, it is difficult to electrically generate high-quality real random digital sequence that can pass all the NIST-standard statistical tests.

Pseudo RNGs such as linear feedback shift register (LFSR) has been used in a number of today's cryptographic LSI systems for mobile applications [1]. Despite PRNGs can suffice for most applications but they suffer from potential attack [2]. Here, we propose a photonic-based random number generator which utilizes an optical component to generate analogue pulses. After the pulses are converted to digital signals or binary random source, the digital signals will be added by LFSR's output which is characterized by a feedback being a primitive polynomial of *n-1*[3].

Since output of the RNG is obtained by adding binary random source which is produced based on optical component into the output of LFSR as shown in Fig 1, the randomness of the RNG's output will be influenced by the LFSR [4]. In this paper, the effects of LFSRs with four different characteristic polynomials i.e. $1+x^7$, $1+x^5+x^7$, $1+x^4+x^5+x^6+x^7$, and $1+x^3+x^4+x^7$ to randomness quality of the RNG's output will be observed by using NIST statistical suite tests.

This article is organised as follows; section 2 presents the concept of randomness, section 3 describes the LFSRs which are used in this observation, section 4 presents the results and discussion and section 5 is conclusions.

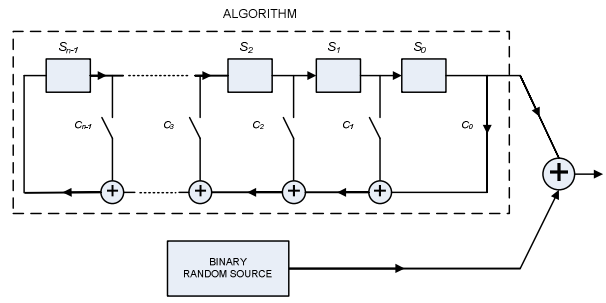

Fig. 1 A stream cipher with a linear shift register as algorithm

II. The Concept of Randomness

The concept and existence of true randomness has been debated for a long time. There has been a debate running throughout the ages of randomness. Some have claimed that some events cannot be fully modelled without disturbing the state (such as observing the spin of photons) and therefore can be random, while others claimed at some level all influential variables controlling an event can be modelled. The debate has pretty much been resolved. Quantum mechanics has shown that randomness does in fact in the real world, and is a critical part of the rules that govern the universe [5].

Definition of randomness cannot be given without introducing more terminology and defining the statistical concept of an autocorrelation function. According to The National Institute of Standards and Technology (NIST), a random bit sequence could be interpreted as the result of the flips of an unbiased fair coin with slides that are labelled 0 and 1, with each flip has a probability of exactly ½ of producing a 0 or 1. Furthermore, the flips are independent of each other; the result of any previous coin flip does not affect future coin flips. The unbiased fair coin is thus the perfect random bit stream generator, since the 0 and 1 values will be randomly distributed. All elements of the sequence are generated independently of each other, and the value of the next element in the sequence cannot be predicted, regardless of how many elements have already been produced [6].

III. Linear Feedback Shift Registers

LFSR is a shift register which is frequently used as pseudorandom pattern generators [7]. A general shift register with feedback as algorithm is illustrated in Fig. 2. Each of the squares labelled S_0, S_1,…, S_{n+1}, is a binary storage element, which might be a bistable flip-flop, position on a delay line or some other memory device. These n binary storage elements are called states of the register and, at any given time, their contents are called its state. A shift register with n stages has 2^n possible states.

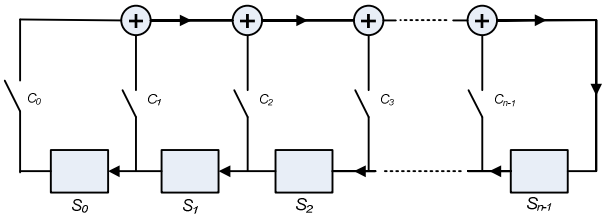

Fig. 2 A shift register with linear feedback as a finite state machine [4]

At time intervals, the content of S_i is transferred into S_{i-1} for all i with $1 \le i \le n-1$ [4]. Since operation of the register inside LFSR is deterministic, the sequence of values produced by the register is completely determined by its current or previous states.

1) Linear Shift Register

The feedback function can be written in the form $f(s_0, s_1, \ldots, s_{n-1}) = c_0s_0 + c_1s_1 + \ldots + c_{n-1}s_{n-1}$, wherein the constants $c_0, c_1, \ldots, c_{n-1}$ are called the feedback coefficient with value 1 will represented by a closed switch while open switch means the corresponding feedback coefficient is 0, and then is called register with linear feedback as shown in Fig. 2 [4]. If $S_i(t)$ denote the content of storage S_i after t^{th} time pulse, for any t, $S_i(t+1) = S_{n-1}(t)$ for $i = 0, 1, 2, \ldots, n-2$, while

$$S_{n-1}(t+1) = \sum_{t=0}^{n-1} c_t s_t(t) \tag{1}$$

2) Polynomials and Periodicity

For any n-stage register with feedback constants $c_0, c_1, \ldots, c_{n-1}$, the characteristic polynomial $f(x)$ *is defined by* $f(x) = c_0 + c_1 + \ldots + c_{n-1}x^{n-1} + x^n$. For a given characteristic polynomial there are 2^n different possible initial settings of the register and, consequently, the polynomial can be used to generate 2^n different sequences of which one will be null. For most choices of $f(x)$ it is possible for two-null sequences to have different periods. If $f(x)$ has the special property of being primitive then every one of its non-null sequences has period 2^n-1, i.e. it is an m-sequence. Furthermore any polynomial which can generate an m-sequence must be primitive [4].

An n-stage linear shift register is determined by the feedback constant $c_0, c_1, \ldots, c_{n-1}$. If S_t is the output sequence generated from an initial state of $s_0, s_1, \ldots, s_{n-1}$, then the following recurrence relation of order n is satisfied:

$$S_{t+n} = \sum_{i=0}^{n-1} C_i S_{t+i} \tag{2}$$

Table I gives the number of primitive polynomial of degree n for $1 \le n \le 24$ which is important to be used to arrange a shift register to generate an m-sequence. As can be seen in the table that if n becomes reasonably large there will a wide choice of polynomials which can be used to generate sequences of maximum period, and that $\lambda(n)$ does not necessarily increase as n increases [4].

TABLE I
The Number of Primitive Polynomials with Degree at Most 24

n	$\lambda(n)$	n	$\lambda(n)$	n	$\lambda(n)$
1	1	9	48	17	7710
2	1	10	60	18	7776
3	2	11	176	19	27594
4	2	12	144	20	24000
5	6	13	630	21	84672
6	6	14	756	22	120032
7	18	15	1800	23	356960
8	16	16	2048	24	276480

IV. DISCUSSION

Testing of the randomness quality is very important issue in cryptography since any practical RNG implementation behaves as a key generator and sometimes generates un-random bits which might be caused by such as circuit saturation, gain errors, temperature and supply voltage variations. There are several test packages and recommendations which are ready to use [8], and the observation and testing used NIST statistical test suite that has sixteen statistical criteria of randomness.

Fig. 3 illustrates proportion values of sixteen statistical criteria of NIST i.e. frequency, frequency within a block, cumulative sum, runs, longest run of ones in a block, random binary matrix rank, discrete Fourier transform (spectral), non-overlapping (a-periodic) template matching, overlapping (periodic) template matching, Maurer's universal statistical, approximate entropy, random excursions, random excursions variant, serial, Lempel-Ziv complexity and linear complexity, wherein the alphabets A to P represent the sixteen statistical criteria that can be referred to TABLE II.

Implementation of LFSR with four varies of characteristic polynomial i.e. $1+x^7$, $1+x^5+x^7$, $1+x^4+x^5+x^6+x^7$, and $1+x^3+x^4+x^7$ in RNG and its randomness testing are shown in Table II. In this testing has used setting of frequency and block frequency is 100, serial and entropy is 5, number bit stream is 643, and length of bit stream is 1000000 [9].

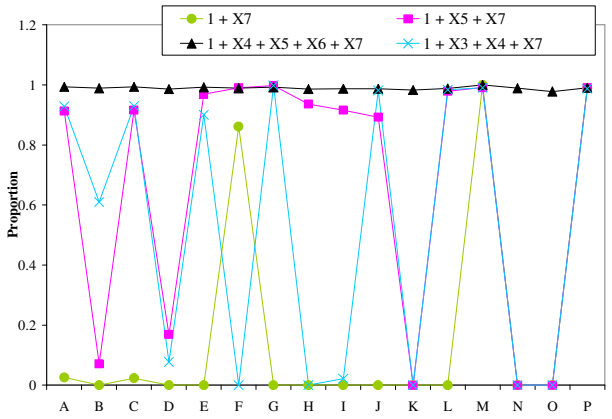

Fig. 3 RNG outputs with LFSR using four different polynomials with their proportion values versus sixteen statistical criteria of NIST.

As can be seen in Fig. 3 that RNG using LFSR which is characterized by a feedback being a primitive polynomial of n-1 degree i.e. $1+x^4+x^5+x^6+x^7$ herein n is the size of register has continuously values close to one without zero values, and different to other LFSRs with different polynomials that have unstable values. LFSR with primitive polynomial $1+x^4+x^5+x^6+x^7$ passes all sixteen statistical criteria of NIST, and gives truly random results. Proportion value that used here was about 3% that it describes a possibility to find un-correct values for all of the sixteen statistical criteria. Hereinafter a graph that illustrates P-values of four different polynomials at sixteen statistical tests of NIST is shown in Fig. 4.

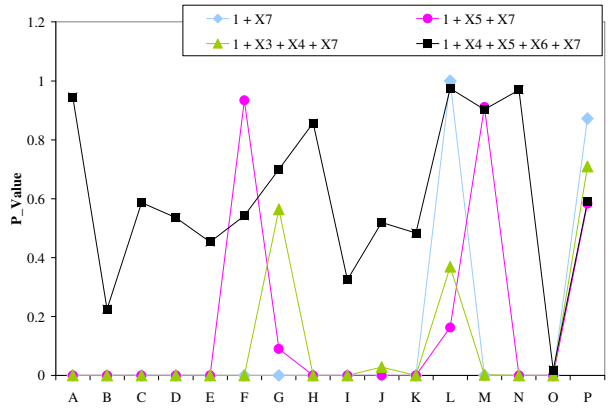

Fig. 4 QRNG outputs with LFSR using four different polynomials with their P-values versus sixteen statistical criteria of NIST.

As shown in Fig. 4, flatness of RNG with LFSR that uses a primitive polynomial of $1+x^4+x^5+x^6+x^7$ does not have null of P-values. Its minimum P-value is around 0.000001 at Lempel-Ziv complexity. Otherwise, LFSRs that use other polynomials have null of its minimum P-values. NIST requires a minimum data of 200 Mbyte for randomness testing and the testing of LFSR with a primitive polynomial of $1+x^4+x^5+x^6+x^7$ that is mentioned above has used data size 334 Mbyte and a baud rate of 57 K character per second (cps). The following figure shows a randomness testing of RNG with LFSR's primitive polynomial of $1+x^4+x^5+x^6+x^7$ using data size of 604 Mbyte that is collected with a data rate of 80 K cps.

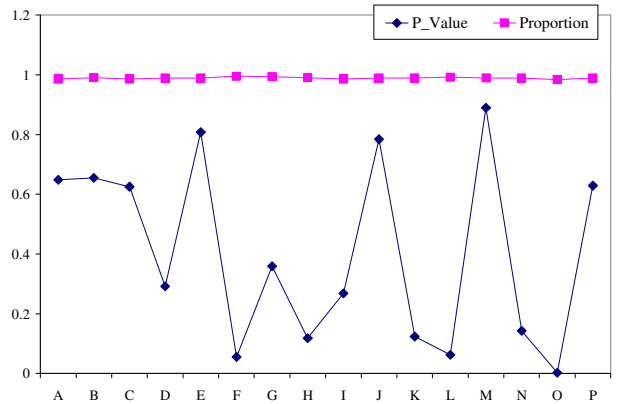

Fig. 5 RNG outputs with LFSR using a primitive polynomial $1+x^4+x^5+x^6+x^7$, data rate 80 K cps and 604 Mbyte data size with its proportion and P values versus sixteen statistical criteria of NIST.

Randomness testing of the RNG using a primitive polynomial $1+x^4+x^5+x^6+x^7$ with bigger data sizes 604 Mbyte and 829 Mbyte and baud rate 80 K cps shows that its outputs are still random without null values appear in the P-values and its proportion values close to one. This output is equivalent to 640 K bps. As shown in Fig. 5 and Fig. 6, that for 604 Mbyte and 829 Mbyte at a baud rate 80 K cps have better minimum P-values of 0.002691 and 0.033231 than using a baud rate 57 K cps that gave a P-value 0.000001.

TABLE II
RESULTS OF NIST TESTS APPLIED ON FOUR SETS OF 250 M-BIT KEY STREAMS WITH DIFFERENT PRIMITIVE POLYNOMIALS

Statistical Tests	Primitive Polynomial $(1+x^7)$		Primitive Polynomial $(1+x^5+x^7)$		Primitive Polynomial $(1+x^4+x^5+x^6+x^7)$		Primitive Polynomial $(1+x^3+x^4+x^7)$	
	P-Value	**Proportion**	**P-Value**	**Proportion**	**P-Value**	**Proportion**	**P-Value**	**Proportion**
[A] Frequency	0.000000	0.0264	0.000000	0.9129	0.945201	0.9938	0.000000	0.9285
[B] Block-frequency	0.000000	0.0000	0.000000	0.0715	0.225004	0.9891	0.000000	0.6096
[C] Cumulative-sums	0.000000	0.0233	0.000000	0.9160	0.586822	0.9938	0.000000	0.9300
[D] Runs	0.000000	0.0000	0.000000	0.1695	0.535722	0.9860	0.000000	0.0778
[E] Longest-runs of Ones	0.000000	0.0000	0.000000	0.9689	0.453289	0.9922	0.000000	0.9005
[F] Rank	0.000000	0.8616	0.933436	0.9907	0.542038	0.9891	0.000000	0.0000
[G] FFT	0.000000	0.0000	0.090508	0.9984	0.700275	0.9922	0.564319	0.9984
[H] Non-periodic-templates	0.000000	0.0000	0.000000	0.9362	0.856822	0.9860	0.000000	0.0000
[I] Overlapping-templates	0.000000	0.0000	0.000000	0.916	0.324580	0.9876	0.000000	0.0218
[J] Universal	0.000000	0.0000	0.000000	0.8927	0.520039	0.9876	0.028428	0.9813
[K] Approximate entropy	0.000000	0.0000	0.000000	0.0000	0.483131	0.9829	0.000000	0.0000
[L] Random-excursions	1.000000	-1.#IND	0.162606	0.9794	0.974208	0.9878	0.369073	0.9841
[M] Random-excursion Variant	0.000648	1.0000	0.911413	0.9912	0.902994	1.0000	0.002634	0.9905
[N] Serial	0.000000	0.0000	0.000000	0.0000	0.969610	0.9891	0.000000	0.0000
[O] Lempel-Ziv Complexity	0.000000	0.0000	0.000000	0.0000	0.000001	0.9782	0.000000	0.0000
[P] Linear Complexity	0.871887	0.9891	0.583596	0.9907	0.590052	0.9907	0.709877	0.9876

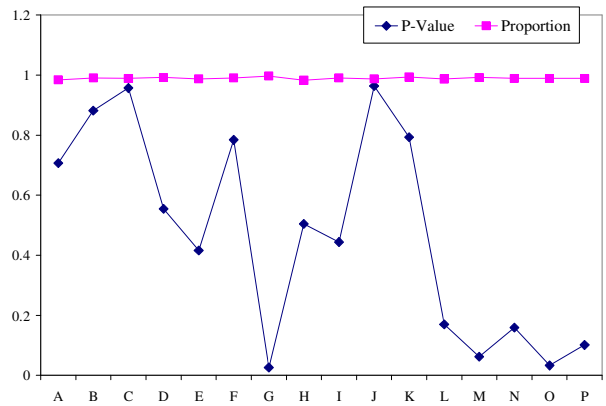

Fig. 6 RNG outputs with LFSR using a primitive polynomial $1+x^4+x^5+x^6+x^7$, data rate 80 K cps and 829 Mbyte data size with its proportion and P values versus sixteen statistical criteria of NIST.

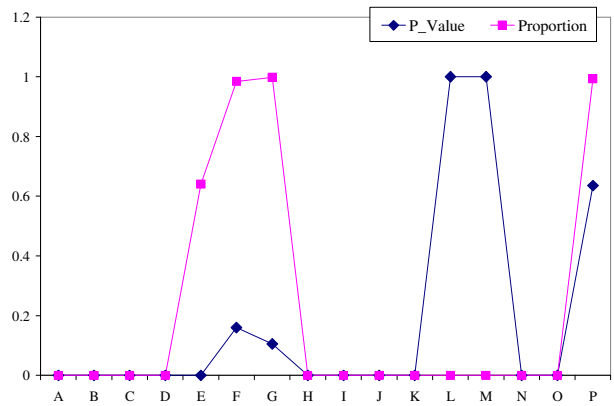

Fig. 7 RNG outputs without LFSR using a primitive polynomial $1+x^4+x^5+x^6+x^7$, data rate 80 K cps and 1.33 Gbyte data size with its proportion and P values versus sixteen statistical criteria of NIST.

The RNG which uses the same primitive polynomial of LFSR has no zero output values as shown in Fig. 6. However without LFSR it produces a lot of zero values as shown in Fig. 7 and passes only three statistical tests of NIST i.e. random binary matrix rank (rank), discrete Fourier transform (fft), and linear complexity. It becomes clearly known that LFSR is an important component in the RNG which gives affects of randomness quality.

V. CONCLUSIONS

In this paper we have proved that a primitive polynomial of LFSR gives effects to randomness quality of the RNG and maintains a high speed output. The LFSR which is characterized by a feedback being a primitive polynomial of n-1 degree i.e. $1+x^4+x^5+x^6+x^7$ has flatness values never reach null and passes all the NIST-standard suite tests, whereas the other LFSR's primitive polynomials have minimum flatness values reach null and cannot pass the test.

REFERENCES

[1] K. Uchida, T. Tanamoto, and S. Fujita, "Single-electron random-number generator (RNG) for highly secure ubiquitous computing applications," *ScienceDirect Solid-State Electronics*, vol. 50, pp. 1552–1557, 2007.

[2] J.–L. Danger, S. Guilley, and P. Hoogvorst, "High speed true random number generator based on open loop structures in FPGAs," *ScienceDirect Microelectronics Journal*, vol. 40, pp. 1650–1656, 2009.

[3] N.M. Thamrin, G. Witjaksono, A. Nuruddin, and M. S. Abdullah, "A Photonic-based random number for cryptographic application," IEEE Computer Society, pp. 356–361, 2008.

[4] H. Beker and F. Piper, *Cipher Systems: The Protection of Communications*, London: Northwood Publications, 1982.

[5] T. S. Denis and S. Johnson, Ed., *Cryptography for Developers*, Rockland, Canada: Syngress, 2007.

[6] A. Rukhin, J. Soto, and at all. , "A Statistical Test Suite for Random Number Generators for Cryptographic Applications," NIST Special Publication 800-22, May 15, 2001.

[7] J. Koeter, *What's an LFSR ?*, Texas Instruments Document, December 1996.

[8] M. Drutarovsky, and P. Galajda, "A robust chaos-based true random number generator embedded in reconfigurable switched-capacitor hardware," *ScienceDirect Solid-State Electronics*, vol. 50, pp. 1552–1557, 2007.

[9] M. Siswanto, G. Witjaksono, and W. F. Hj. Yaakob, "Parallel QRNG with Multi Random Source (MRS)," MIMOS Patent Pending, 2009.