DESIGN OF A NEW CRYPTOGRAPHIC ALGORITHM USING HIGH-THROUGHPUT RM-PRNG

Shazia Tazyeen[1], Dr G.S.Biradar[2], Meenakshi Patil[3]
[1]PG Student, VLSI Design & Embedded Systems, Department of PG Studies, VTU RC, Gulbarga, India.
[2,3]Professor, Department of E&CE, PDA College Of Engineering, Gulbarga, India.

Abstract—Pseudo Random Number Generator (PRNG) is an algorithm for generating a sequence of numbers. Due to speed in number generation pseudorandom numbers are very important. They are mostly used for cryptography applications. In cryptography there are mainly two mechanisms. They are Encryption and Decryption. In Encryption the sender will convert the original message called plain text into cipher text which is unreadable, using an encryption key. This cipher text will be send through communication channel to receiver. The receiver needs to decrypt the cipher text into plain text by using decryption key. Here key plays main role. The security of the encrypted message depends on key. The key should be unpredictable, random, and nonlinear and hardware cost for generating key should be less. There are many ways to generate random keys. Here, a technique called Reseeding mixing Pseudo Random number generator simply RM-PRNG is used. The reseeding-mixing method is used to extend the system period length and to enhance the statistical properties of a chaos-based logistic map pseudo random number generator (PRNG). The reseeding method removes the short periods of the digitized logistic map and the mixing method extends the system period length to 2^{253} by "XORing" with a DX generator.

Keywords—Encryption, Decryption, Reseeding, Mixing, PRNG, Peroid Extension.

I. INTRODUCTION

PSEUDO random number generator (PRNG) has been widely used in Monte Carlo simulations, test pattern generation, cryptography, and telecommunication systems. A good PRNG should have characteristics of:

1) long-period random number sequence;
2) A fit in statistical properties;
3) A high throughput rate; and
4) An unpredictability.

PRNGs are efficient, meaning they can produce many numbers in a short time, and deterministic, meaning that a given sequence of numbers can be reproduced at a later date if the starting point in the sequence is known. Efficiency is a nice characteristic if your application needs many numbers, and determinism is handy if you need to replay the same sequence of numbers again at a later stage. PRNGs are typically also periodic, which means that the sequence will eventually repeat itself. While periodicity is hardly ever a desirable characteristic, modern PRNGs have a period that is so long that it can be ignored for most practical purposes. These characteristics make PRNGs suitable for applications where many numbers are required and where it is useful that the same sequence can be replayed easily. Popular examples of such applications are simulation and modeling applications. Pseudorandom numbers are important in practice for their speed in number generation and their reproducibility, and they are thus central in applications such as simulations and modeling. Linear PRNGs, such as linear feedback shift registers (LFSRs), linear congruential generators (LCGs), and multiple recursive generators (MRGs) can produce long-period random number sequences. When implemented, linear PRNGs are efficient in throughput rate and hardware cost, but the output random numbers of such generators are predictable due to their linear structure. Some nonlinear PRNGs in dealt with the predictability problem, but incurred higher hardware cost and more process time. Recently, nonlinear chaos-based PRNGs (CB-PRNGs) with lower hardware cost were proposed. However, there exist short periods in simple CB-PRNG due to quantization error. The throughput performance of these CB-PRNGs is usually low due to the fact that they can only produce one random bit in iteration, and there is no assurance for the output sequences of these CB-PRBGs to have satisfactory statistical properties. And then to produce long periods and high throughput rate reseeding-mixing PRNG (RM-PRNG) were proposed. The RM-PRNG consists of a CB-PRNG and MRG. The reseeding method removes the short periods in the CB-PRNG and by mixing MRG with CB-PRNG the overall system period length increases. In this brief, we propose a new encryption and decryption method, in an encryption scheme the message or information is encrypted by using an encryption algorithm, changing it into unreadable cipher text by XORing with the RM-PRNG key. Any adversary that can see the cipher text should not able to determined anything about the original message. However the cipher text is converted to plain text (original message) by using decryption algorithm.

The main objective of this project is to design a secure cryptography algorithm using reseeding mixing pseudo random number generator. The cryptography is the practice and study of the techniques to ensure secure communications in the presence of third parties.

II. LITERATURE SURVEY

Reseeding technique is widely used in LFSR for test pattern generation [10], [11] and in CB-PRNG for period extension. Cernák [4] presented a reseeding method either to perturb the state value or the system parameter of digitized logistic map (LGM) for removing the short periods of a CB-PRNG. In

1998, Sang *et al.* applied a different reseeding method in perturbing a CB-PRNG to extend its period length up to 3.3672×1029 [5], and the lower bound of the reseeded system can be calculated. Li *et al.* [6] discovered that the reseeding technique not only removes the short periods but also improves the statistical properties of CB-PRNG. Recently, these merits of reseeding were confirmed by exhaustive simulation [12] in a 32-b implementation of a CB-PRNG. On the other hand, the mixing technique has been used for nonlinearity enhancement of cellular automata [2], [3] and for improving statistical performance of nonlinear PRNGs. For example, Lü *et al.* [7] proposed a software implementation of mixing multiple spatial-temporal CB-PRNGs to obtain high security, fast encryption (decryption) speed, and reliable robustness. Li *et al.* [8] extended a single-bit pseudo random bit generator (PRBG) to multiple-bit PRBG. The mixing technique is also widely applied in period extension of nonlinear PRNGs. Gammel *et al.* [1] mixed several nonlinear feedback shift registers (NLFSRs) to obtain a long-period and high-throughput-rate stream cipher. Addabbo *et al.* [9] mixed two simple CB-PRNGs to form a combined PRNG whose period length can be calculated analytically. In general, mixing multiple CB-PRNGs results in higher hardware cost, lower throughput rate, and longer but unpredictable period length. Furthermore, one cannot be sure that the random numbers produced by these mixed PRNGs will have acceptable statistical properties. Since higher hardware cost is due to implementation of multiple CB-PRNGs which are more complex than linear PRNGs, mixing a CB-PRNG with a linear MRG instead of mixing two CB-PRNGs will reduced the hardware cost. In our proposed RM-PRNG, which consists of a CB-PRNG and an MRG, the period length is considerably extended because the period length of the MRG is much longer than that of the CB-PRNG while the short periods of the CB-PRNG can be removed by our reseeding algorithm. The lower bound of the period length in RM-PRNG can be calculated analytically in terms of the period length of the CB-PRNG and that of the MRG. In addition, the throughput rate is enhanced using a vector-mixing technique in the proposed RM-PRNG. Finally, the statistical properties are improved because the linear structure of the MRGs is broken by mixing with a CB-PRNG.

III. SYSTEM DESIGN

The RM-PRNG is composed of three modules: Nonlinear Module, Reseeding Module, and Vector Mixing Module. In a 32-b implementation, the Nonlinear Module has a controlled 32-b state register and a Next-State construction circuitry. The controlled register stores the state value X_t which can be set to Seed1 by the Start command. The Next-State construction circuitry produces the next state value according to the recursive formula. For each generated state value, the reseeding control unit (RCU) in the Reseeding Module compares the values of and for checking the fixed point condition and increases the reseeding counter (RC) at the same time. The RC will be reset and the reseeding operation will be activated when either the fixed point condition is detected or the RC reaches the reseeding period.

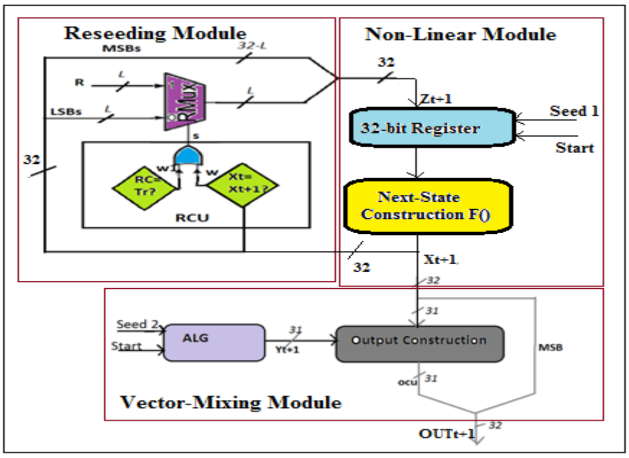

Fig.1: Block Diagram of RM-PRNG

A. NONLINEAR MODULE

We use the LGM as the next-state construction function in the Nonlinear Module so that

$$X_{t+1} = F(X_t) = \gamma X_t(1 - X_t), t \geq 0 \qquad \text{----(1)}$$

With $\gamma = 4$ and $X_0 \in (0,1)$ as an initial seed. Choosing a value 4 for γ not only makes the LGM chaotic but also simplifies the implementation of (1) to merely left-shifting the product of X_t and $(1-X_t)$ by 2b. However, the state size decreases from 32 to 31 b, because the dynamics X_t and $(1-X_t)$ in (1) are the same. This is equivalent to a degradation of resolution by 1 b. In addition, fixed points (at $X_t = 0$ and 0.75) as well as short periods exist when the LGM is digitized. From exhaustive runs for all of the 2^{32} seeds, we obtain all other periods for the 32-b LGM (P_{LGM}) without reseeding, with the longest period (18 675) and the set of short periods Ts (≤1338). Clearly, the performance of a CB-PRNG using only the Nonlinear Module is unsatisfactory. To solve the fixed points and short-period problem, a Reseeding Module is in order.

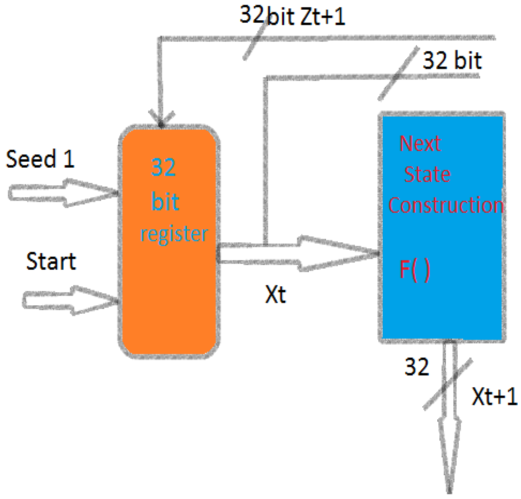

Fig. 2 : Block Diagram of Non-linear Module

B. RESEEDING MODULE

The removal of the fixed points by the reseeding mechanism is obvious. When the fixed point condition is detected or the reseeding period is reached, the value Z_{t+1} loaded to the state register will be perturbed away from X_{t+1} in the RCU by the fixed pattern R according to the formula

$$Z_{t+1}[j] = \begin{cases} X_{t+1}[j], & 1 \le j \le 32 - L; \\ R[i], & 33 - L \le j \le 32, \; i = j + L - 32 \end{cases} \qquad \text{-----(2)}$$

Where subscripts i,j are the bit-index, L is integer, and R ≠ 0. In order to minimize the degradation of the statistical properties of chaos dynamics, the magnitude of the perturbation of the fixed pattern R should be small compared with X_t. Here, we set L = 5 so that the maximum relative perturbation is only $(2^5 - 1)/2^{32}$ and the degradation can be ignored. Clearly, the effectiveness of removing short-periods depends on the reseeding period Tr as well as the reseeding pattern R. However, choosing the optimal reseeding period and the reseeding pattern is nontrivial. First, the reseeding period should avoid being the values or the multiples of the short periods Ts of the unperturbed digitized LGM. Otherwise, if the 5 LSBs of X_{t+1} equal to R when the reseeding procedure is activated, Z_{t+1} will be equal to X_{t+1}. Then no effective reseeding will be realized and the system will be trapped in the short-period cycle. Hence, prime numbers should be used as the reseeding period candidates. Here , we use Tr=643 and R = “18(10010)". One can see that the set of short periods Ts is indeed eliminated. The lowest period, the maximum period and the average period of the reseeded PRNG are, respectively, 1929, 2 330 875, and 2 321 423.005. Note that the period after reseeding is a multiple of the reseeding period. Although the average period of the reseeded PRNG has increased more than 100 times relative to that of the non-reseeded counterpart, the period can in fact be extended tremendously in the Vector Mixing Module described below.

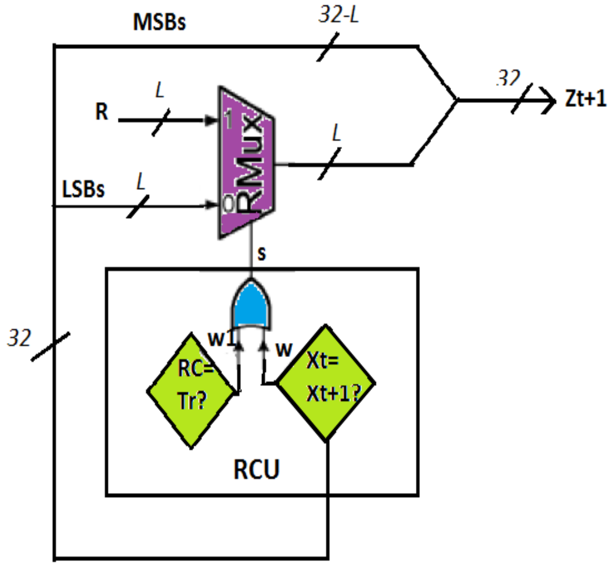

Fig. 3: Block Diagram of Reseeding Module

C. VECTOR MIXING MODULE

An efficient MRG, called the DX generator, serves as the ALG in Vector Mixing Module. Specifically, we choose the DX generator with the following recurrence equation:

$$Y_{t+1} = Y_t + B_{DX} \, . Y_{t-1} \, mod \, M, t \ge 7 \qquad \text{-----(3)}$$

Using an efficient search algorithm, we find that the particular choice of $B_{DX} = 2^{28} + 2^8$ and $M = 2^{31} - 1$ gives the maximum period of the DX generator. The LSBs of Y_{t+1} and that of X_{t+1} are mixed in the Output Construction unit using a XOR operation to obtain the least significant bits of the output according to the equation

$$OUT_{t+1}[1:31] = X_{t+1}[1:31] \oplus Y_{t+1}[1:31] \qquad \text{-----(4)}$$

Then, the most significant bit (MSB) of X_{t+1} is attached to $OUT_{t+1}[1:31]$ to form the full 32-b output vector $OUT_{t+1.}$

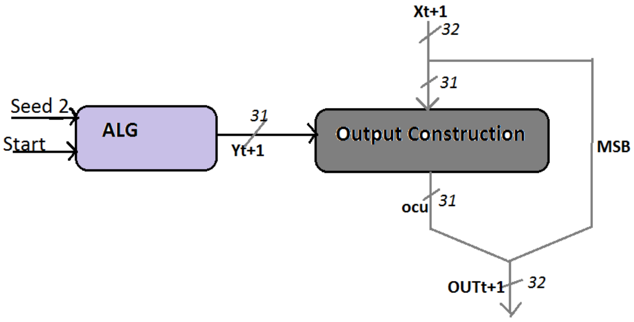

Fig. 4: Block Diagram of Vector Mixing Module

The Output Construction in the Vector Mixing Module involves simple exclusive OR operation and its implementation is trivial. On the other hand, hardware implementation of the DX generator (the ALG) has not been reported before. Hence, more detailed implementation of the DX generator is given below

D. DX GENERATOR(ALG)

An efficient MRG, called the DX generator [13], [14], serves as the ALG in Vector Mixing Module. Specifically, we choose the DX generator with the following recurrence equation

$$Y_{t+1} = Y_t + B_{DX} \, .Y_{t-7} \text{ MOD } M \qquad \text{--------(5)}$$

Using an efficient search algorithm [14], we find that the particular choice of $B_{DX} = 2^{28}+2^8$ and $M = 2^{31}-1$ gives the maximum period of the DX generator. To implement with the multiplier as $B_{DX} = 2^{28}+2^8$, the multiplication and modulus operations are replaced by circular-left shift (CLS) and end-around-carry (EAC) addition operations. In this way, the hardware implementation cost can be reduced dramatically. Fig.5 shows the proposed implementation of the DX generator. First, the eight-word register is implemented by flip-flops with 1935 gates. Signal Y_{t-7} is circular-left-shifted 28 and 8 b for generating two partial products, using module CLS-28 and CLS-8, respectively. Then, a circular 3-2 counter combines three 31-b operands into two 31-b operands, which consumes 247 gates. Finally, a 31-b EAC carry look ahead adder (CLA), with 348 gates, is utilized to evaluate Y_{t+1} . Fig. 6 shows the schematic design of the 31-b EAC-CLA, which includes four units: propagation and generation (PG) generators, end-around-carry (EAC) generator, internal carry

(IC) generator, and CLAs. After EAC is generated by group-PGs, EAC is then fed to the IC generator and then to the least-significant 8-b CLA. The final addition is performed on CLAs.

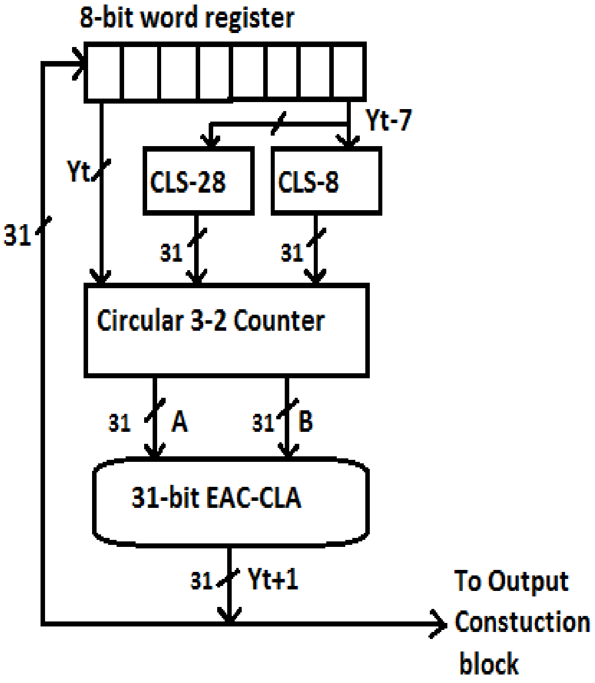

Fig. 5: Block Diagram of DX-Generator (ALG)

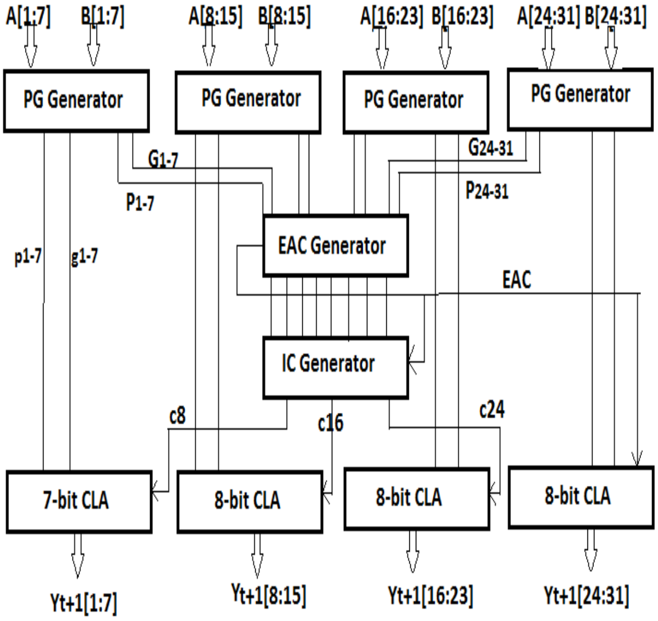

Fig. 6: Block Diagram of EAC-CLA

IV. SIMULATION RESULTS AND COMPARISON

RM-PRNG is designed using Verilog language. The simulation results are obtained by using Modelsim and Synthesis is observed by Xilinx ISE 10.1. The simulated output of non-linear, reseeding module and DX-generator given in figures 7,8 & 9.

A. Non-Linear Module

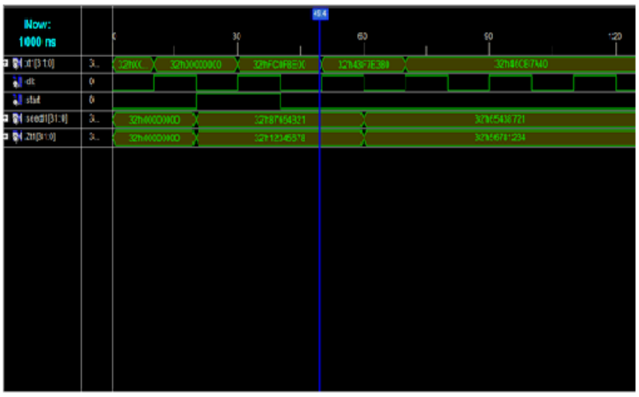

Fig. 7: Simulated output of non-linear module

B. Reseeding module

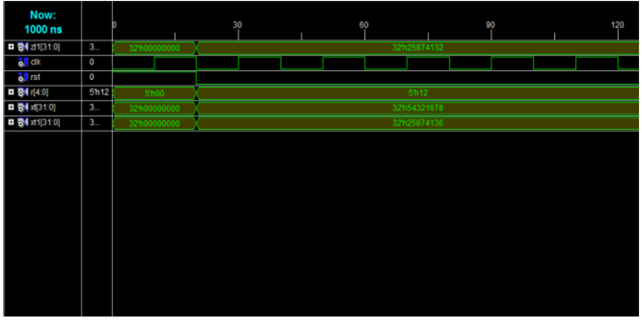

Fig. 8: Simulated output of reseeding module

C. DXGenerator(ALG)

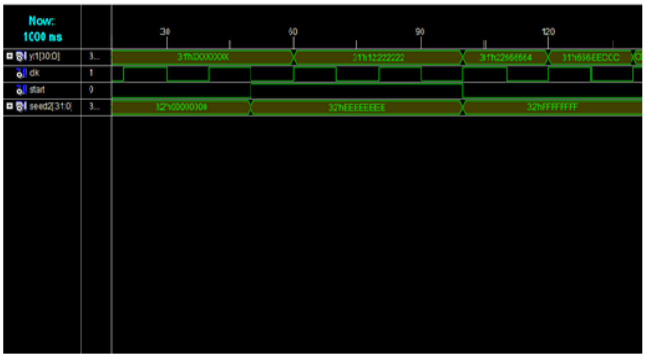

Fig. 9: Simulated output of DX-Generator(ALG)

D. The final output of RM-PRNG is shown in figure 10.

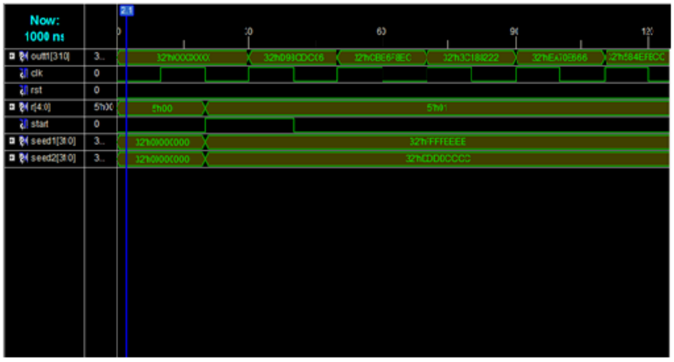

Fig. 10: Simulated output of RM-PRNG

Table I : Comparisons of PRNG'S in Throughput(Th) and Hardware efficiency(Go)

PRNG	Th (Mbit/sec)	Go
LFSR	500	2.24
LCG	6200	7.311
DX-Generator	6200	2.45
LGM	200	0.02
Sawtooth	200	0.05
NLFSR	400	0.159
RM-PRNG	6400	0.538

Table1 tabulates simulation results for linear generators of LFSR (x28+x3+1), LCG, DX- generator, nonlinear generators of reseeding LGM[12] , sawtooth map PRNG[9] , NLFSR[1] , and our RM-PRNG in terms of throughput (Th) and hardware efficiency G0. Comparing with previous linear PRNGs, the proposed RM-PRNG generates nonlinear sequences at a high-throughput rate. On the other hand, when compared with other nonlinear PRNGs, our RM-PRNG is the fastest and its period length is longest with the best hardware efficiency.

V. CONCLUSIONS

Here, we proposed a hardware implementation of RM-PRNG to offer long periods and high throughput rate while adhering to established statistical standards for PRNGs. The reseeding mechanism solves the short-period problem originated from the digitization of the chaotic map, while mixing a CB-PRNG with a long-period DX generator extends the period length to the theoretically calculated value greater than 2^{253}. Replacing a hardware-demanding CB-PRNG with a hardware-efficient MRG, the hardware cost is reduced and the hardware efficiency achieves 0.538 Mb/s-gate. In addition, the high throughput rate (>6.4 Gb/s) is attained because RM-PRNG can generate multiple random bits in an iteration. For randomness enhancement, the proposed reseeding-mixing method successfully improved the statistical properties of CB-PRNG.With all these advantages, the proposed nonlinear RM-PRNG can a good candidate for potential applications in test pattern generation, telecommunication system and even cryptography if the security issue can be addressed properly.

VI. ACKNOWLEDGMENT

I would like to thank the innumerable people throughout my life who taught me ,including my parents, my academic teachers for their generous support and my guide who had been guiding throughout the project and supporting me in giving technical ideas about the paper and motivating me to complete the work efficiently and successfully.

REFERENCES

[1] B. M. Gammel, R. Goettfert, and O. Kniffler, ―An NLFSR-based stream cipher,‖ in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 2917–2920.

[2] D. Mukhopadhyay,D. R. Chowdhury, and C. Rebeiro, ―Theory of composing non-linear machines with predictable cyclic structures,‖ in *Proc. 8th Int. Conf. Cellular Autom. Res.Ind.*, 2008, pp. 210–219, Springer.

[3] D. Mukhopadhyay, ―Group properties of non-linear cellular automata,‖ *J. Cellular Autom.*, vol. 5, no. 1, pp. 139–155, Oct. 2009.

[4] L. Y. Deng and H. Xu, ―A system of high-dimensional, efficient, longcycle and portable uniform random number generators,‖ *ACM Trans. Model Comput. Simul.*, vol. 13, no. 4, pp. 299–309, Oct. 1, 2003.

[5] T. Sang, R. Wang, and Y. Yan, ―Clock - controlled chaotic keystreamgenerators,‖ *Electron. Lett.*, vol. 34, no. 20, pp. 1932–1934, Oct. 1998.

[6] S. Li, X. Mou, and Y. Cai, ―Pseudo-random bit generator based on couple chaotic systems and its application in stream-ciphers cryptography,‖ in *Progr. Cryptol.-INDOCRYPT*, 2001, vol. 2247, pp. 316–329, Lecture Notes Comput. Sci.. .

[7] H. Lü, S.Wang, X. Li, G. T. J. Kuang, W. Ye, and G. Hu, ―A new spatiotemporally chaotic cryptosystem and its security and performance analyses,‖ *Chaos*, vol. 14, no. 3, pp. 617–629, Sep. 2004.

[8] P. Li, Z. Li, W. A. Halang, and G. Chen, ―Analysis of a multipleoutput pseudo-random-bit generator based on a spatiotemporal chaotic system,‖ *Int. J. Bifurc. Chaos*, vol. 16, no. 10, pp. 2949–2963, 2006.

[9] T. Addabbo, M. Alioto, A. Fort, A. Pasini, S. Rocchi, and V. Vignoli, ―A class of maximum-period nonlinear congruential generators derived from the Rènyi chaotic map,‖ *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, pp. 816–828, Apr. 2007.

[10] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, ―Built-in test for circuits with scan based on reseeding of multiplepolynomial linear feedback shift registers,‖ *IEEE Trans. Comput.*, vol. 44, no. 2, pp. 223–233, Feb. 1995.

[11] J. Lee and N. A. Touba, ―LFSR-reseeding scheme achieving low-power dissipation during test,‖ *IEEE Trans. Comput.-Aided Design (CAD) Integr. Circuits Syst.*, vol. 26, no. 2, pp. 396–401, Feb. 2007.

[12] C. Y. Li, T. Y. Chang, and C. C. Huang, ―An nonlinear PRNG using digitized logistic map with self-reseeding method,‖ in *Proc. IEEE Int. Symp. VLSI Design,*

[13] L. Y. Deng and H. Xu, ―A system of high-dimensional, efficient, longcycle and portable uniform random number generators,‖ *ACM Trans. Model Comput. Simul.*, vol. 13, no. 4, pp. 299–309, Oct. 1, 2003.

[14] L. Y. Deng, ―Efficient and portable multiple recursive generators of large order,‖ *ACM Trans. Modeling Comput. Simul.*, vol. 15, no. 1, pp. 1–13,jan2005.