Design of a New Cryptography Algorithm using Reseeding-Mixing Pseudo Random Number Generator

S. Dilli Babu & Madhu Kumar Patnala

ECE Dept., Sree Vidyakethan Engineering College, Tirupati, Chittoor, A.P, India
E-mail : setterydilli98@gmail.com , madhukumar.patnala@gmail.com

Abstract - In this paper, we propose the application of cryptography algorithm to ensure secure communication across the virtual networks. In cryptography, encryption is the process of encoding messages or information in such a way that hackers cannot read it. In an encryption scheme the message or information is encrypted using an encryption algorithm, turning it into an unreadable cipher text. This is usually done with the use of an encryption key. Any adversary that can see the cipher text should not know anything about the original message. To decode the cipher text using an algorithm that usually requires, a secret decryption key. An encryption scheme usually needs a key generating algorithm to randomly produce keys. 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. The output sequence of RM-PRNG is used as a key to the encryption and decryption modules. The simulation results are obtained by using modelsim 6.3g_p1.

Keywords - PRNG, encryption, reseeding, decryption, mixing, RM-PRNG.

I. INTRODUCTION

Pseudo random number generator (PRNG) is a random number generator that produces sequence of numbers. Pseudo random numbers are important for their speed in number generation. Pseudo random numbers are useful for a variety of purposes, such as generating data encryption and decryption keys. Applications of PRNGs are in Monte Carlo simulations, test pattern generation, cryptography and telecommunication systems. To produce long period random number sequence linear PRNGs are very useful, some of the linear PRNGs are linear feedback shift registers (LFSRs), linear congruential generators (LCGs), and multiple recursive generators (MRGs). These linear PRNGs are good in hardware cost and throughput rate. But due to their linear structure output random numbers of these generators are easily predictable. To overcome the predictability problem nonlinear chaos-based PRNGs (CB-PRNGs) [8] were proposed, it is efficient in hardware cost, but due to quantization error there exists short periods in such nonlinear PRNGs. They produce only one bit per iteration hence throughput rate is low. 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 [1], [8]. 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.

II. RM-PRNG

RM-PRNG is implemented by nonlinear module, Reseeding module, Vector mixing module. Nonlinear module is implemented by 32-bit register and next state construction circuitry. The state register stores the state value (Xt) which is set to seed1 by using the start command. The next state construction is used to produce the next state value (Xt+1) by using recursive formula Xn+1=F (Xn) [1].

Reseeding module is implemented by reseeding multiplexer (RMux) and reseeding control unit (RCU) [1]. The RCU compares the values of Xt and Xn+1 for checking the fixed point condition (Xn+1= Xt) for each
generated state value, and also increases the reseeding counter (RC) at the same time. When RC reaches the reseeding period $T_r$ or the fixed point condition is detected then RC will be reset and the reseeding operation will be activated. The sate register will be loaded through the RMux, when reseeding is activated [1]. The value of $X_{t+1}$ is directly loaded into the state register if the reseeding is not activated. Vector Mixing Module is implemented by an auxiliary linear generator (ALG) and output construction. By mixing $X_{t+1}$ with the output $Y_{t+1}$ from ALG in Vector Mixing Module, we obtain the output of the RM-PRNG (32-bit implementation).

![Structure of RM-PRNG](image)

**Fig. 1**: Structure of RM-PRNG

### A. Nonlinear Module

In non linear module we are using logistic map (LGM) as the next-state construction so that

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

By choosing $\gamma = 4$ and $X_0 \in (0,1)$ as an initial seed, it makes the LGM chaotic and also simplifies the equation (1) to left shifting the product of $X_t$ and (1 - $X_t$) by 2 bits. The dynamics $X_t$ and (1 - $X_t$) in the equation (1) are the same hence the state size decreases from 32 to 31 b. When the LGM is digitized fixed points (at $X_t = 0$ and 0.75) as well as short periods exist. We obtain all other periods for the 32-b LGM ($P_{LGM}$) without reseeding for all of the $2^{32}$ seeds. By using only the Nonlinear Module the performance of a CB-PRNG is unsatisfactory. To solve the fixed points and short-period problem, a Reseeding Module is needed.

### B. Reseeding Module

To remove the short periods and fixed points the reseeding mechanism [1] is obvious. The value $Z_{t+1}$ is loaded to the state register, when the reseeding period is reached or the fixed point condition is detected, according to the formula

$$Z_{t+1} = \{X_{t+1}[j], \quad 1 \leq j \leq 32 - L;\} \cup \{R[i], \quad 33 - L \leq j \leq 32, \quad i = j - 32$$

Where $i$ and $j$ are the bit-index, $L$ is an integer, and $R$ should be not equal to 0 ($R \neq 0$). To improve statistical properties of chaos dynamics, the magnitude of the perturbations for the fixed pattern $R$ should be small compared with $X_t$.

In reseeding module short-periods can be removed by reseeding period $T_r$ as well as the reseeding pattern $R$. To choose a suitable combination of $T_r$ and $R$ several guidelines were proposed. The reseeding period $T_r$ should avoid being the values or the multiples of the short periods $T_s$ of the digitized LGM. When the reseeding procedure is activated $Z_{t+1}$ and $X_{t+1}$ will be equal, if the 5 LSBs of $X_{t+1}$ equal to $R$. Hence the system will be trapped in the short-period cycle. In this study, we use 643 and 18 values for $T_r$ and $R$ respectively. The average period of the reseeded PRNG has increased more than 100 times compared with non-reseeded PRNG [1]. And the period can be extended tremendously in the Vector Mixing Module described.

### C. Vector Mixing Module

The Vector Mixing Module is constructed by using ALG and output construction. In this module an efficient MRG which is called as DX generator [1], [7] acts as the ALG. By using the following recurrence formula

$$Y_{t+1} = Y_t + BCX \cdot Y_{t+1} \mod M, \quad t \geq 7$$

In output construction unit, to obtain the LSBs of the output the LSBs of $Y_{t+1}$ and that of $X_{t+1}$ are mixed by XOR operation according to the following equation

$$OUT_{t+1}[1:31] = X_{t+1}[1:31] \oplus Y_{t+1}[1:31]$$

To form the full 32-b output vector $OUT_{t+1}$ the MSB of $X_{t+1}$ is added to $OUT_{t+1}[1:31]$.

### D. DX Generator (ALG)

From the figure 2, the implementation [2] of the DX generator is (the ALG) done by using 8-word registers, circular-left-shift (CLS), circular 3-2 counter and End Around Carry- carry look ahead adder (EAC-CLA). By using flip-flops the eight-word register was implemented. For generating two partial products signal $Y_{t+1}$ is circular-left-shifted 28 and 8 b [1], using the modules CLS-28 and CLS-8 respectively. To combine these three 31-b operands into two 31-b operands a
circular 3-2 counter is used, which consumes 247 gates. To evaluate $Y_{t+1}$, 31-b EAC-CLA is used with 348 gates. The schematic design of the 31-b EAC-CLA [4], [9] is shown in Figure 2(b). The schematic design of the 31-b EAC-CLA includes four modules they are propagation and generation (PG) generators, end-around-carry (EAC) generator, internal carry (IC) generator, and CLAs [5]. When EAC is generated by group of PGs, EAC is then fed to the IC generator and then to least-significant 8-b CLA. On CLAs, the final addition was performed.

Fig. 2. (a) Structure of the DX generator. (b) Structure of the 31-b EAC-CLA

III. PROPOSED CRYPTOGRAPHIC ALGORITHM

The increasing application of cryptographic algorithms to ensure secure communications across virtual networks has led to an ever-growing demand for high performance hardware implementations [9] of the encryption and decryption methods.

Cryptography is the practice and study of techniques for secure communication in the presence of third parties. In cryptography, encryption is the process of encoding messages or information in such a way that hackers cannot read it. Cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. In Cryptography [9], encryption is the process of converting ordinary information (called plaintext) into unintelligible gibberish (called cipher text). Any adversary that can see the cipher text should not know anything about the original message. Decryption is the reverse, in other words, moving from the unintelligible cipher text back to plaintext. The statistical properties of cryptographic algorithms [11] are the reason for the excellent pseudorandom testability of cryptographic processor cores.

IV. RESULTS AND SIMULATION

Pseudo Random Number Generator, Encryption and Decryption were designed using Verilog language in ModelSim 6.3. All the simulations are performed using ModelSim 6.3 simulator. The simulated output of Pseudo Random Number Generator, Encryption and Decryption are shown in Figure 4&5.
Fig 4 : Simulation results for RM-PRNG

Fig 5 : Simulation results for cryptography

V. CONCLUSION

In this paper, we proposed a cryptographic algorithm using RM-PRNG to ensure secure communication. This Cryptographic algorithm allows people to carry over the confidence found in the physical world to the electronic world, thus allowing people to do business electronically without worry of deception. With these secure communications, the proposed cryptographic algorithm using RM-PRNG can a good candidate for protect the data in ATM cards, computer passwords and electronic commerce.

VI. ACKNOWLEDGEMENT

S. Dilli Babu would like to thank Mr. P. Madhu Kumar, Assistant professor (senior grade) ECE Department 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.

VII. REFERENCES


★★★★