IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p-ISSN: 2278-8735
PP 19-22
www.iosrjournals.org

Design of Efficient Random Number Generation Using Linear Feedback Shift Register

S.Shalini[1], T.Muruganantham[2], V.Subhashini[3]
[1]*PG Scholar, Department of ECE, K.Ramakrishnan College Of Engineering,Samayapuram.*
[2]*Assistant Professor,Department of ECE, K.Ramakrishnan College Of Engineering,Samayapuram.*
[3]*Assistant Professor,Department oof ECE, K.Ramakrishnan college of Engineering,Samayapuram.*

Abstract: *RSA cryptosystem is one of the most widely used public key cryptosystem. We present a new structure to develop 128 bit RSA Encryption. The importance of high security and faster implementations. The whole RSA includes three parts: key generation, encryption and decryption process. The algorithm also requires random prime numbers so a primality tester also design to meet the needs of the algorithm. Main motive of this work is accelerating performance of RSA by using efficient modules. These blocks are coded in verilog and are synthesizes and simulated in Xilinx 14.2 design. Keywords: RSA, Verilog, Cryptosystem, Encryption ,Decryption*

I. Introduction

Cryptography is the study of maintaining the secrecy of information. Prior to the modern age, cryptography was the process of converting information from readable information to apparent unreadable data. The technique used to decode the "unreadable data" back into readable data was known only to the intended recipients, thereby preventing unsolicited persons from gaining access to the readable data. Since World War II and the dawn of computers, the practices used in cryptography (encryption/decryption) have become incredibly complex and its usage has become more extensive.

Cryptography allows people to carry over the confidence found in the physical world to the electronic world, thus allowing people to do business electronically without worries of deceit and deception. Every day hundreds of thousands of people interact electronically, whether it is through e-mail, e-commerce, ATM machines, or cellular phones. The perpetual increase of information transmitted electronically has lead to an increased reliance on cryptography.

Network security consists of the provisions and politics adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or assigned an ID and password or other authenticating information that allows them to access information and programs within their authority. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among business, government agencies and individuals. Secure communications is when two entities are communicating and do not want a third party to listen it in. For that they need to communicate in a way not susceptible to eavesdropping or interception. Secure communication includes means by which people can share information with varying degrees of certainty that third parties cannot intercept what was said. With many communications taking place over long distance, it is necessary to increase that awareness of the importance of interception issues, technology and its compromise to improve the security of the information. Secure communication is necessary in many fields such as military battle field, medical field, government offices etc.

II. Overview Of Rsa

It was developed by Rivest, Shamir &Adleman of MIT in 1977. This method is supposed to be the best and commonly used as a public-key scheme which is based on exponentiation in a finite(Galois) field over integers modulo a prime. The security is due to cost of factoring large numbers. As already explained in RSA cryptosystem there are two key, the public and private key. The public key is advertised to the world and the private key is supposed to kept secret. Therefore an anonymous person will not be able do decrypt the encrypted message if he does not have the private key. The safety depends upon the length of the key, longer the key-length much safer is the data.

Following are the steps involved in the RSA algorithm:

A. Key generation

Key generation is the most important aspect of RSA Algorithm.
The steps are as follows:

“Select two random prime numbers p and q

Calculate n = p x q

– 1) x (q – 1)
Calculate ø(n) = (p

Select integer e such that gcd (ø(n),e) =

1;1<e<ø(n); where e & ø(n) are relatively prime

Calculate d = e

-1 mod ø(n)”

According to the procedure the encryption key e is available but the decryption key d is not known to all. Mathematically this procedure is defined as, M is the actual message, C is the converted message or cipher text by using publicly available encryption key e, and d is the decryption key.

C = M^e(mod m) M = C^d (mod m)
RSA encryption and decryption are mutual inverses and commutative .

III. Rsa Algorithm

RSA algorithm is divided into blocks and each block is then implemented.

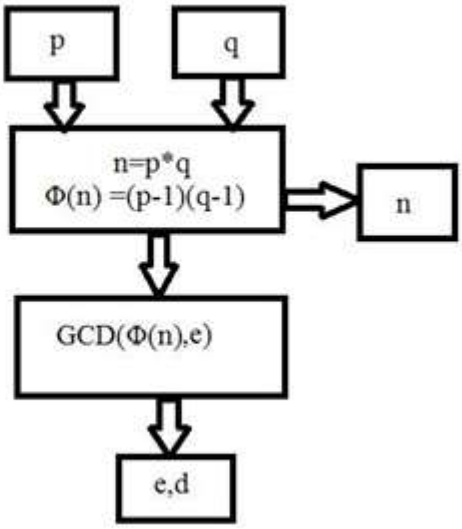

Fig.1. RSA key generation

The first step is the generation of public and private keys which is summarized in fig.1. It starts with a pseudorandom number generator that generates 32-bit pseudo numbers. These pseudorandom numbers are stored in a FIFO. The pseudorandom number generator will stop working as the FIFO is full. Random numbers from the FIFO are pulled out by the primality tester as this happens the PRNG will start again to make sure that the FIFO remains filled. Coming back to the primality tester, it takes a random number as input and check for the number to be prime. If the number is proved as prime, it goes to the Prime FIFO. The primality tester only pulls a number out of the FIFO when the prime FIFO is not full. When there is a requirement of new keys, two random prime numbers are extracted from the prime FIFO.

These two numbers are used to calculate n and □ (n). □ (n) is forwarded tothe Greatest Common Divider (GCD)

calculator where a number e is selected which will be the public or encryption key if it satisfies the condition that GCD (**□ (n), e) = 1. This will prove that modular inverse d of this number exists** and the modular multiplicative inverse will be our private our decryption key. We got e, d and n, for encryption and decryption.

Modular exponentiation is applied to encrypt or decrypt the data. This is something that has to be focused because the performance of RSA algorithm depends on how modular arithmetic functions are calculated. They are the core of the algorithm. This process is shown in Fig.2

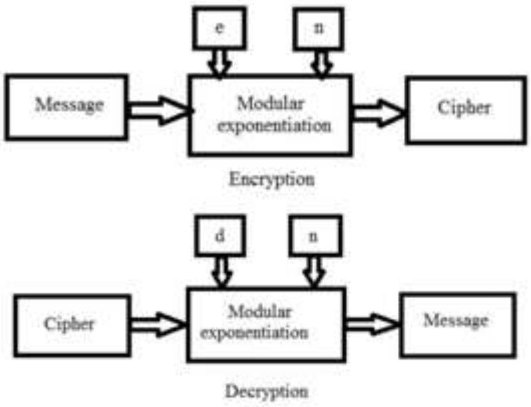

Fig.2. Encryption and Decryption

A. Random Number Generator

The Linear Feedback Shift Register (LFSR) is among the most useful techniques used for generating pseudorandom numbers. Here 32-bit pseudo random numbers are generated using LFSR. LFSR generates a periodic sequence means that the pattern in which numbers are generated will be repeated after certain interval. When using a primitive polynomial, maximum length of an LFSR sequence is 2n-1.A 32-bit LFSR will produce a sequence of over 4 billion random bits, or 500 million random bytes.The polynomial used for generating this sequence of 32-bit pseudorandom numbers is as under.

P(x) = x32+x22 +x2+x1 +1

IV. Simulation Results

LFSR have been successfully written in verilog and tested on Xilinx ISE 14.2. The simulation shows these desired results of these algorithms.

Linear Feedback Shift Register

The pseudo random number generated by a 128 bit Linear Feedback Shift Register which is simulated in Xilinx ISE.

Fig.3. Simulated waveform for pseudo random number Generator using LFSR

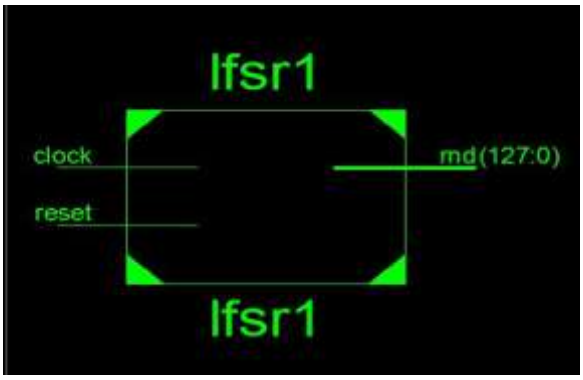

Fig.4. RTL Schematic for random number generator

V. Conclusion

Here implemented a 128 bit RSA circuit in verilog. It is full featured and efficient RSA circuit. Further process included primality testing, key generation, data encryption and decryption process. I have implemented random number generator using 128 bit LFSR.

References

[1]. Symeon (Simos) Xenitellis, "A guide to PKIs and Open– source Implementations", The Open–source PKI Book

[2]. VibhorGarg, V. Arunachalam ,"Architectural Analysis of RSA Cryptosystem on FPGA", International Journal of Computer Applications (0975 – 8887) Volume 26– No.8, July (2011).

[3]. William Stallings, "Cryptography and Network Security Principals and Practices",4th edition, Pearson Education, Inc., (2006).

[4]. Sushanta Kumar Sahu, ManoranjanPradhan, "FPGA Implementation of RSA Encryption System", International Journal of Computer Applications (0975 – 8887), Volume 19– No.9, April (2011)

[5]. Monier, L. "Evaluation and Comparison of Two Efficient Probabilistic Primality Testing Algorithms." Theor.Comput.Sci. 12, 97-108, (1980).

[6]. Rabin, M. O. "Probabilistic Algorithm for Testing Primality." J. Number Th. 12, 128-138, (1980).

[7]. Joe Hurd, "Verification of the Miller-Rabin Probabilistic Primality Test", Computer Laboratory, University of Cambridge

[8]. Chia-Long WU, "An Efficient Montgomery Exponentiation Algorithm for Cryptographic Applications", INFORMATICA, Vol. 16, (2005) No. 3, 449–468.

[9]. AnkitAnand, Pushkar Praveen, "Implementation of RSA Algorithm on FPGA", International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue