ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Implementation of Network Security Algorithm on FPGA

[I]L.Saranya Devi, [II]S.Selvi
[I]M.E VLSI Design Student, Dr.SACOE, Tiruchendur, Tamilnadu, India
[II]Assistant Professor (SG),Dept of ECE, Dr.SACOE, Truchendur, Tamilnadu, India

Abstract
The electronic data transmission in a network is done in a safe and secure way with the help of Advanced Encryption Standard (AES), a Federal Information Processing Standard (FIPS). The AES can be programmed in software or built with hardware. The paper presents a hardware implementation of the AES algorithm. The AES was implemented in FPGA using Spartan 3E kit and Xilinx ISE development suite. The use of FPGA for cryptographic applications is highly attractive for variety of reasons but at the same time there are many open issues related to the general security of FPGAs. This contribution attempts to provide a state-of-the-art description of this topic. First, the advantages of reconfigurable hardware for cryptographic applications are listed. Second, potential security problems of FPGAs are described in detail, followed by a proposal of some countermeasure. Third, a list of open research problems is provided. Even though there have been many contributions dealing with the algorithmic aspects of cryptographic schemes implemented on FPGAs, this contribution appears to be the first comprehensive treatment of system and security aspects.

Keywords
AES algorithm, Hardware implementation, FPGA, Spartan 3E, Xilinx

I. Introduction

Cryptography is the art and science of protecting information from undesirable individuals by converting it into a form which will be stored and transmitted in a way that is non-recognizable by unauthorized users. Information in the form of online data, or password needs some form of security. Security is a primary prerequisite of any wireless cryptographic protocol. The solution of which is to build a high standard cryptographic algorithm that provides secure communication applications. However, the smart design of an algorithm is necessary if there is a need to maintain a secure data application. Although there exist lot of algorithms, the security is not met with all [1]. Many works from different research groups have been published, analysing cryptographic methods for finding holes in the security strength of today's encryption algorithm. Thus new encryption algorithms are needed that do not have any security defects.

One of the major problems of modern computer security is the design of cryptographic algorithms that have as little vulnerabilities as possible while maintaining their low implementation complexity. Many algorithms that are cryptographically secure are not easily implemented in hardware. Thus the need for hardware implementations of secure algorithms becomes even greater.

In cryptography, the AES, also known as Rijndael algorithm, is a block cipher adopted as an encryption standard by the US government, which specifies an encryption algorithm capable of protecting sensitive data [2, 3]. A combination of factors such as security, performance, efficiency, ease of implementation and flexibility contributed to the selection of this algorithm as the AES. The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. The AES algorithm uses cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits [4, 5].

This paper holds the AES technique, its security issues and the performance while implementing in FPGA. This paper is organized as follows: Section II is presented with algorithm. Section III presents the implementation of AES on FPGA. Finally, section IV concludes this paper.

II. AES Algorithm

AES operates on a 4×4 column-major order matrix of bytes, termed the state. The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. The algorithm includes three main parts: encryption, decryption and Key Expansion. Encryption converts original data (Plain text) into an unintelligible form called cipher text while decryption converts cipher text back into its original form called plaintext. Key Expansion generates a Key Schedule that is used in Cipher and Inverse Cipher procedure. Cipher and Inverse Cipher are generated after some specific number of rounds illustrated in Table 1. For the AES algorithm, the number of rounds to be performed during the execution depends entirely on the key length [6].

Table No 1

AES	Key Length(Nk)	Block Size (Nb)	No of Rounds (Nr)
128	4	4	10
192	6	4	12
256	8	4	14

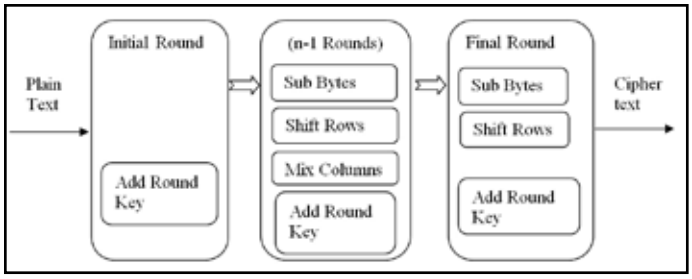

Fig. 1 : AES Block Diagram

Each encryption round consists of four stages:

- Substitute Bytes
- Shift rows
- Mix columns
- Add round key

The initial stage requires only add round key to change the plaintext order. Other rounds involve the operation of all the four stages and the final round with three stages.

Substitute Bytes- Is a non linear byte substitution, using a substitution table (S-box) each byte from the input state is replaced by another byte. S-box translates each nibble into a new nibble.

Shift rows: In the Shift Rows transformation, the first row of the

state array remains unchanged [7] and other rows are affected by a state of diffusion. Mix columns: The forward mix column transformation, called MixColumns, operates on each column individually. Each last column is multiplied with a standard matrix format. Addroundkey: In the forward add round key transformation, called AddRoundKey, the 128 bits of state are bitwise XORed with the 128 bits of the round key obtained from key expansion. The overall AES algorithm is shown in figure 1.The decryption works with the inverse operation of the above mentioned stages.

(i). Key Expansion

In the AES algorithm, the key expansion module is to generate round keys for every round. The original key consists of 128 bits arranged into a 4x4 matrix of bytes. This matrix is expanded by adjoining 40 more columns. There are two approaches to provide round keys [8], [9]. One is to pre-compute and store all the round keys, and the other one is to produce them on-the-fly. In this paper it has been implemented with first approach. The key expansion has three steps:

- Byte Substitution subword()
- Rotation rotword()
- XOR with RCON (round constant)

Keyrotate

The function Keyrotate takes a four-byte word and rotates one byte to the left.

Keysub-bytes

The Keysub-bytes operation takes four-byte input word by substituting each byte in the input to another byte according to the S-Box.

KeyRcon

The first byte of a word is XORed with the round constant. Each value of the Rcon table is a member of the Rijndael finite field. Add round key is same for the both encryption and decryption.

III. Field Programmable Gate Array (FPGA)

FPGA is an integrated circuit that can be reconfigured by designers themselves. With each reconfiguration, which takes only a fraction of a second, an integrated circuit can perform a completely different function. FPGAs can be used for specific operational behaviour, or general purpose CPU functionality depending on the complexity of the device.

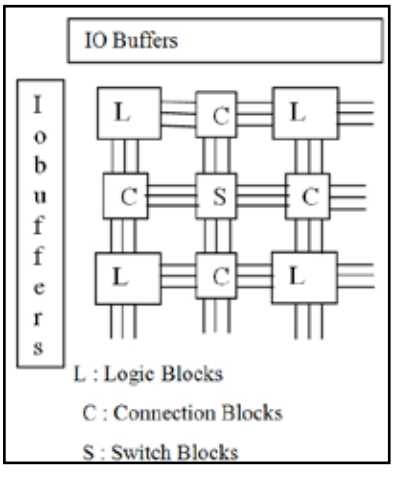

Fig. 2: FPGA Architecture

FPGA applications include DSP applications, imaging, speech recognition, cryptography, hardware emulation and for many other application specific uses. Many FPGAs are implementing shared general purpose CPUs on chip for shorter latency times between operations resulting in higher performance of the overall system. FPGA is an integrated circuit (IC) that includes a two-dimensional array of [11]

(i). Programming FPGA

Different techniques for programming FPGAs (i.e. to load the configuration bits into SRAM cells) were recognized. The flow then proceeds through compilation, simulation, programming, and verification in the FPGA hardware.

Steps for designing FPGA:

1. Design the circuit which has to map to Xilinx part on the FPGA.
2. Synthesize the design using the XST synthesis tool.
3. Generate a UCF file to hold constraints (pin assignments). Use the Plan Ahead tool to generate this file.
4. Assign the I/O pins of the design to the pins on the FPGA that are in need to connect.
5. Implement the design to the specific FPGA on the Spartan-3E board.
6. Generate the programming .bit file that has the bit stream that configures the FPGA.
7. Connect Spartan3 board to the computer and use the IMPACT tool to program the FPGA using the bit stream.

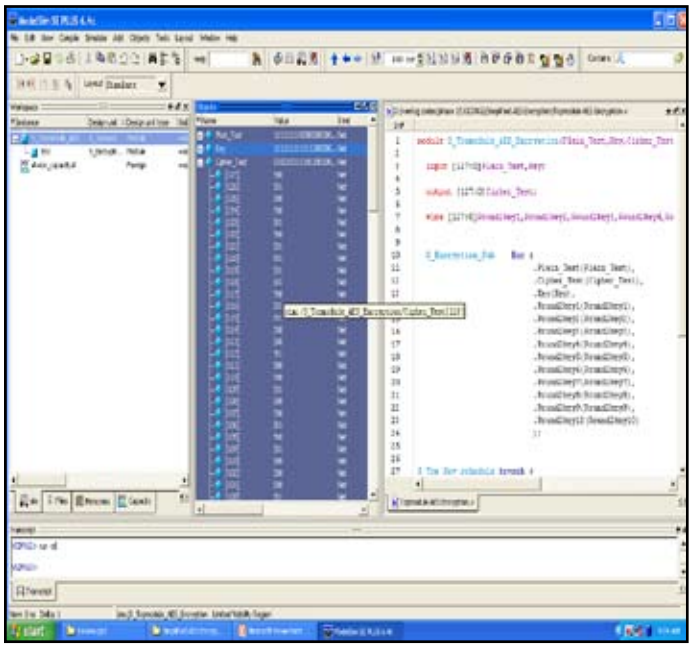

Fig. 3: Encryption Results

(ii). Simulating Result

The design has been coded by Verilog HDL and results shown in figure above are synthesized and simulated on the Xilinx ISE design suite. AES algorithm simulation is shown in figure 3.

Encryption

The simulation results of encryption module is shown in Fig 3 where input text and keys are given and output was verified.
Input Plain text: 00145362334455667788 99a35bccddef
Input Cipher Key: 000124604050607080 90a0b0c0d760ef

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

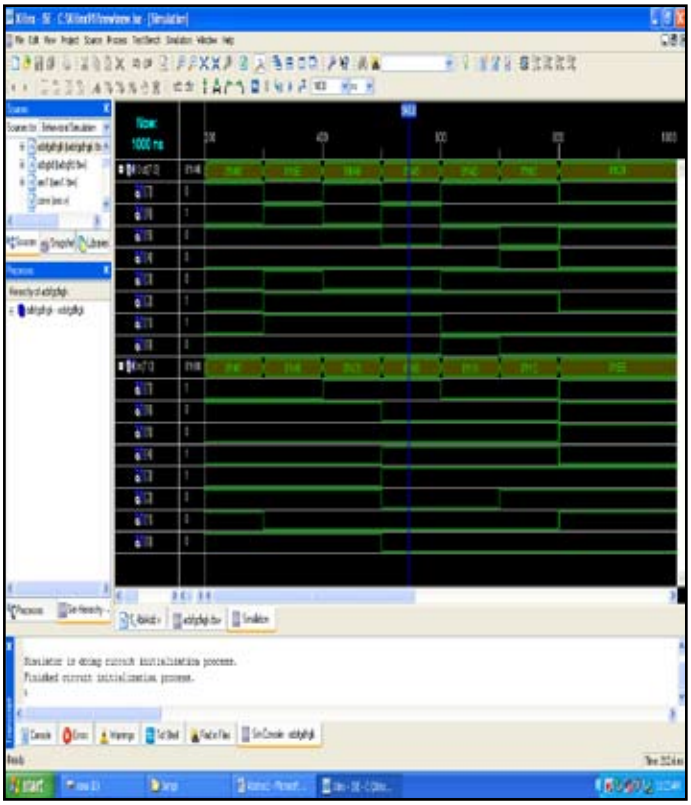

Fig. 4 : Simulation Result

IV. Conclusion

The Advanced Encryption Standard algorithm is a symmetric block cipher that can process data blocks of 128 bits through the use of cipher keys of varying lengths of 128, 192, and 256 bits. An efficient implementation of 128 bit block and 128 bit key AES algorithm has been presented in this paper. The design is implemented on Spartan 3E FPGA.

V. Acknowledgement

L.SARANYA DEVI would like to thank Mrs S.SELVI Assistant professor (SG) of ECE Department, Dr.SACOE, Tiruchendur, who is guiding throughout the project and supporting me in giving technical ideas about the paper and motivating me to complete the work effectively and successfully.

References

[1] *Atul Kahate, "Cryptography and Network Security", Second Edition, Tata McGraw-Hill Edition 2008.*
[2] *National Inst. Of Standards and Technology, "Federal Information Processing Standard Publication 197, the Advanced Encryption Standard (AES)," Nov. 2001*
[3] *J. Daemen and V. Rijmen," AES Proposal: Rijndael," AES Algorithm Submission, Sept. 1999*
[4] *William Stallings, Cryptography and Network Security, Principles and Practices, 4th ed. Pearson Education, pp. 134-161, 2006*
[5] *Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security, Private Communication in a Public World, 2nd ed. Pearson Education, pp. 41-114, 2006*
[5] *DAEMEN, J.—RIJMEN, V. : AES Proposal: Rijndael, The Rijndael Block Cipher, AES Proposal, pp.1–45, 1999 (http://csrc.nist.gov/CryptoToolkit/aes/).*
[6] *Wayne Wolf, "FPGA-Based System Design", Pearson Education.*
[7] *Tessier, R., and Burleson, W., "Reconfigurable computing for digital signal processing: a survey", J.VLSI Signal Process., 2001, 28, (1-2), pp.7-27.*
[8] *Ahmad, N.; Hasan, R.; Jubadi, W.M; "Design of AES S-Box using combinational logic optimization", IEEE Symposium on Industrial Electronics & Applications (ISIEA), pp. 696-699, 2010.*
[9] *Daemen J., and Rijmen V, "The Design of Rijndael: AES-the Advanced Encryption Standard", Springer- Verlag, 2002*
[10] *[Onlin]e Available: http://eetimes.com/design/programmable-logic/4014815/All-about-PGAs?pageNumber=2*

L.Saranya Devi has done her B.E in Electronics and Communication Engineering from PSRR College of Engineering,Sivakasi, India in the year 2012 and currently pursuing her M.E in VLSI Design from Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, India. Her areas of interest include VLSI Design, Digital Electronics and Cryptography.

S.Selvi was born at Tirunelveli, Tamilnadu, India in 1975. She is an Assistant professor of Electronics and Communication Engineering, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, India. She received her B.E. and M.E in Electronics and Communication Engineering and Computer Science Engineering respectively from Manonmaniam Sundaranar University, Tirunelveli ,India. Her research interests include Digital Signal Processing, Distributed Computing and Grid Computing.