Journal of Advanced Research in Technology and Management Sciences
Volume: 01 Issue: 03 ISSN: 2582-307 November 2019
Available online at: http://www.jartms.org

HIGH PERFORMANCE FPGA ARCHITECTURE OF AES ENCRYPTION FOR SECURED CRYPTOGRAPHY

[1]JAYASHREE CHAVAN, [2]Dr.AMIT JAIN
[1]Research Scholar, Department of ECE, Sunrise University, Alwar, Rajasthan, India
[2]Research Guide, Department of ECE, Sunrise University, Alwar, Rajasthan, India

ABSTRACT: In this digital age of communication, private and confidential data is exchanged over internet and stored in digital mediums. Cryptography is one of the techniques to protect sensitive data. The cryptographic techniques are the one that have been implemented to enhance the data security level in all of such communication systems. The symmetric cryptographic technique of Advanced Encryption Standard (AES) is considered suitable for this security enhancement. This standard has become one of the most widely used encryption method and has been implemented in both software and hardware. Hardware implementation would be faster and secure as compared to software implementation. However Field Programmable Gate Arrays (FPGA) implementation offers quicker solution and can be easily upgraded to incorporate any protocol changes. This research investigates high performance FPGA architecture of AES encryption for secured cryptography. High performance AES encryption is designed and implemented in FPGA, which is shown to be more efficient in terms of speed and area. This paper provided a guideline for implementation of a more secure way with cryptographic algorithm of AES.

KEYWORDS: communication systems, data security, Cryptography, Advanced Encryption Standard (AES).

I. INTRODUCTION

It has become a pressing issue to ensure 5G communication system application like Internet of Things (IoTs), Cloud applications, Software defined networks are secure and trusted as they are widely used and gain popularity. Compromised communication systems may lead to loss of confidentiality, integrity of not only the devices have easier physical access than servers housed in data-centers. Cryptographic-system is an important part of total security. It is used to protect not only the data communicated but also the system itself. Cryptography provides solutions regarding fool proof secrecy, security, and reliability of given information [1]. It is keeping its vital function in different applications which includes online banking system, Cellular networks, computer hardware emulations, medical imaging, software defined radios, bioinformatics and wireless communication etc. Reconfigurable platform like FPGA are the best for implementation of cryptographic algorithms. These platforms are reconfigurable to provide time and cost effective solutions as compared to Application Specific Integrated Circuit (ASIC). A reconfigurable platform provides improved performance than software implementations and can also be reconfigured on the fly to store the updated encryption standard [2].

To minimize the algorithmic process time in term of plenty of data, it is very much inevitable to adopt and implement the algorithm of hardware, despite the fact that software implementation can only meet the requirement of low cost for users. In order to attain a balance between the cost and time, an efficient method must be explored and implement for various combinations of hardware and software to realize algorithmic best solutions of different requisite. The

Advanced Encryption Standard (AES) is a computer security standard issued by the National Institute of Standards and Technology (NIST) intended for protecting electronic data. The Advanced Encryption Standard can be implemented in either software or hardware. Hardware implementation can be used to perform the operation more efficiently than possible in software [3].

II. LITERATURE SURVEY

Abdullah al mamun [4] in this paper they used random byte to modify s box by performing ex-or on all bytes of s box and hence discussing parameters like time security, avalanche effect and strict avalanche criterion. But this algorithm didn't have key dependent sbox. In [5], an advanced masked AES is proposed to protect the AES from DPA and glitch attacks. Masking is done at all of the four stages of AES algorithm. Results achieved in this are: area reduction by 36.4% with 4.5Mbits/slice and protection against the DPA and glitch attacks. The main purpose of this work is the throughput. In [6], Ahmed Alahmadi et.al worked on primary user emulation attacks in cognitive radio networks operating in the white spaces of the digital TV (DTV) band. Authors proposed a reliable AESassisted DTV scheme, in which an AES-encrypted reference signal, is generated at the TV transmitter and used as the sync bits of the DTV data frames.

Pallavi Atha et al [7] have present The AES algorithm uses cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt datas and This methodology uses VHDL implementation over FPGA. They have programmed in Xilinx – 10.1 xst software and implemented on FPGA families which are Spartan2, Spartan3 and Virtex2 and calculations of Time, Speed & power have been done for appropriate output. Pachamuthu Rajalakshmi et al [8] have presented a compact hardware-software co-design of Advanced Encryption Standard (AES) on the field programmable gate arrays (FPGA) designed for low-cost embedded systems. The design uses MicroBlaze, a soft-core processor from Xilinx. The computationally intensive operations of the AES are implemented in hardware for better speed. By incorporating the processor in the AES design, the total number of slices required to implement the AES algorithm on FPGA is proved to be reduced. The entire AES system design is validated using 460 slices in Spartan-3E XC3S500E, which is one of the low-cost FPGAs.

Krishnamurthy [9] in this paper they created dynamic behavior in sbox by rotating the sbox and this rotation value was key dependent and compared avalanche effect of basic AES with their proposed AES but this scheme has a more complicated decryption algorithm. Xinmiao Zhang et al [10] have presented various approaches for efficient hardware implementation of the Advanced Encryption Standard algorithm. They optimization methods can be divided into two classes: architectural optimization and algorithmic optimization. Architectural optimization exploits the strength of pipelining, loop unrolling and sub-pipelining. Speed is increased by processing multiple rounds simultaneously at the cost of increased area. Architectural optimization is not an effective solution in feed-back mode. Loop unrolling is the only architecture that can achieve a slight speedup with significantly increased area.

III. HIGH PERFORMANCE SECURE CRYPTOGRAPHY USING AES

In this section, high performance AES architecture for secured cryptography is discussed in brief with the computations, rounds and steps involved in encryption and decryption. Encryption and decryption in AES are carried out by specific ciphers and its inverse. The plain text is applied as input along with cipher key for encryption. The encrypted output will be considered as input for the Decryptor. The encrypted output with inverse cipher is applied to Decryptor. The algorithm is composed of three main parts: Cipher, Inverse Cipher and Key Expansion. The first part Cipher is used for encryption, inverse cipher for decryption and key expansion is used for deriving add round key for each round. Cipher will convert data into encrypted form so that resultant is in unintelligible form, called as Cipher text. Fig. 1 shows architecture of AES algorithm for secure cryptography.

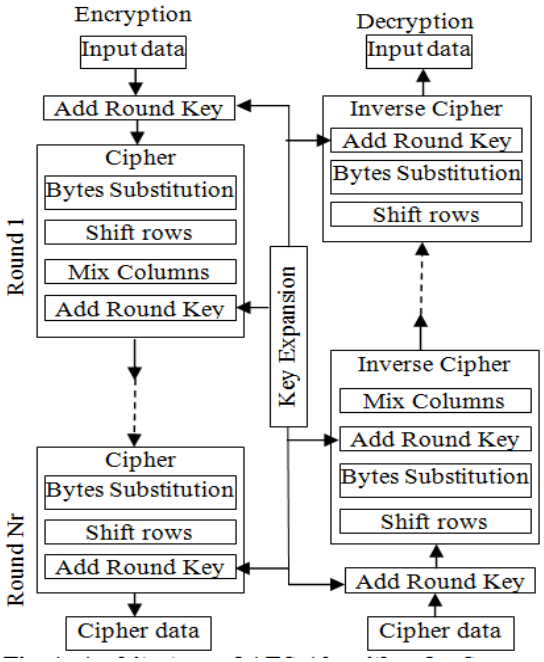

Fig. 1: Architecture of AES Algorithm for Secure Cryptography

Byte oriented transformations will give encrypted or decrypted states. Each round key is derived from cipher key using Key expansion. Four different byte transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey are performed in each round of Cipher and Inverse Cipher. Advance encryption system is symmetric cipher which uses the same key for encryption as well as decryption. AES is a block cipher which performs operations on blocks of data. AES uses 128-bit plaintext and uses variable keys viz 128,192 and 256. Depending on the size of the key it performs different rounds of operations (Nr-Number of rounds) i.e.10 rounds for 128 bit key, 12 rounds for 192 bit key and 14 rounds for 256 bit key. At present, the most used key size is 128 bit. At internal level of AES, cipher key is expanded to 11, 13 and 15 keys for 10, 12 and 14 rounds respectively. Then the plaintext is copied into an array named state array. State array is 4x4 matrix where each array contains byte of plaintext arranged horizontally.

3.1 Cipher

The input data is converted to state array then the round function is performed. The round function is consists of four different byte transformations. The Cipher converts input data to an unintelligible form called ciphertext. After an initial Round Key addition, the State array is transformed by implementing a round function with the final round differing slightly from the first round. The round function is parameterized using a key schedule that consists of a onedimensional array of four-byte words (Round Key) derived using the Key Expansion routine. All Nr rounds except final rounds are identical. In the final round

MixColumn transformation are not carried out. The AES algorithm will take initial Cipher key and performs a key expansion to generate Key Schedule.

3.2 Inverse Cipher

The output of the Encryptor called as Ciphertext will be given as input to the Decryptor. Decryption is carried out using Inverse Cipher. Decryption processing blocks are shown in Fig. 1. The Ciphertext is copied to the state array before operating using Inverse Cipher. The last round key is added and after that the State array is transformed. The round function used for state array transformation is of AddRoundKey transformation and three different inverse transformations. Decryption is carried out using reverse order Round Keys. The final round is different from the first Nr – 1 round. This procedure converts ciphertext back to its original form called plaintext. All Nr rounds are identical with the exception of the final round, which does not include the Inverse Mix-Columns transformation. Encryption and decryption blocks are implemented as hardware blocks. The Sbox is implemented as a look up table and row transformation is handled by barrel shifter. Each block is coded as described in NIST FIPS document.

3.3 Key Expansion

The AES algorithm requires 4 words of round keys for each encryption round. That can be a typical of four*(Nr+1) round keys thinking about the initial set of keys required for the number one upload round Key transformation. All of the round keys are derived from the cipher key itself. The AES key expansion algorithm takes as input a four-word key and produces a linear array of 44 words. The Key is copied into the primary 4 words of the expanded key. Then the rest of the key is filled in 4 words at a time. Each added word w[i] relies upon on the straight away preceding word, w[i−1], and the word four positions returned w[i−4]. In three out of four instances, a simple XOR is used. Every round uses four of these words and each word incorporates 32 bytes which suggest each subkey is 128 bits long.

3.4 Bytes substitution

This transformation operates on each byte of the state using substitution table which is a non-linear byte. Substitution table consists of ff rows/columns.

3.5 Shift rows

With the Shift Row, transformation first row is not shifted and the remaining three rows are shifted circularly. In the second row, one byte is left shifted circularly. For the three row, a 2-byte round left shift is done. For the fourth row, a three-byte round left shift is accomplished. And for the decryption technique, it will be shifting towards right circularly.

3.6 Mix Columns Transformation

This transformation is based on Galois field multiplication. Each byte of a column is changed with the different value that may be a feature of all 4 bytes in the given column. The transformation operates on the state column, treating each column as a polynomial. The columns are considered as polynomials over GF (28)

3.7 AddRoundKey

The AddRoundKey operation is meant as a cipher; all the 128 bits of the state unit of activity XORed with four, 32-bit words of the extended key on account of key enlargement. AddRoundKey is the most

effective operation that entails using the key to making sure safety. The AES key expansion set takes a four-word (16-byte) as input and produces a linear array of 44 words.

IV. RESULTS

Hardware implementation results are targeted for Xilinx Virtex-6 FPGA. The design has been implemented using Xilinx System Generator tool and test bench are finally synthesized and simulated. AES architecture is implemented in two parts Encryption and Decryption. The input data is processed in the 128 bits block format. The encryption takes 1900 ms for processing 10 rounds. The standard NIST keys are used for encryption. Due to pre-calculated vectors decryption requires less time. The decryption cycle is 1800ms. The decrypted output is two line text same as applied to the encryption. Fig. 2 shows that the FPGA based AES algorithm records the fastest encryption and decryption time than conventional AES encryption time.

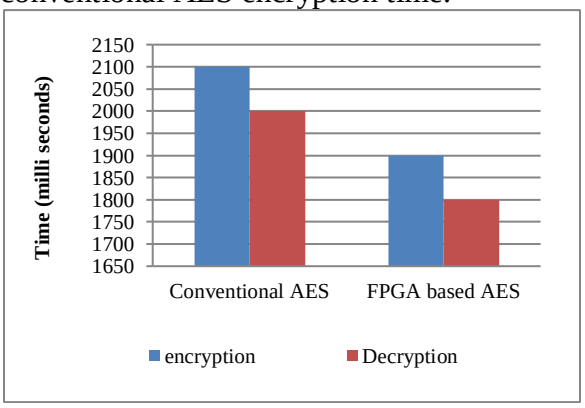

Fig. 2: Result Analysis of Encryption and Decryption Time Consumption

Fig. 3 shows that the FPGA based AES algorithm records the less Look Up table (LUT) utilization for encryption implementation than conventional AES .

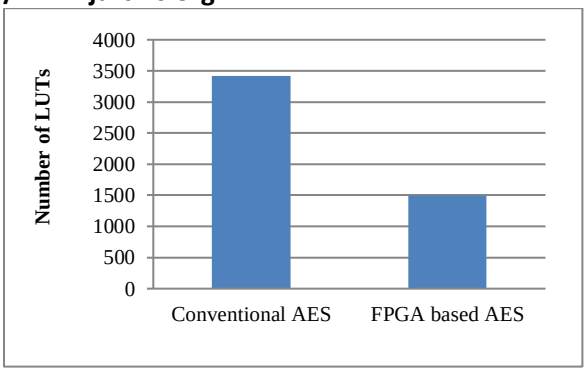

Fig. 3: Utilization of LUTs for Encryption and Decryption

Up next in the table 1 presents that memory used for unit operations for all cryptographic techniques that we studied. Blowfish consumed less memory storage than other types, while RSA uses the highest memory.

Implementation	Frequency	Throughput
AES-128	74.4MHz	7.76Mbits/s
Software AES-8	124.6 Cycles/byte	-
AES using ARM-32	34 Cycles/Byte	-
High performance FPGA based AES-32	288.19 MHz	36.864 Gbps

V. CONCLUSION

In this project, high performance FPGA architecture of AES encryption was designed, modeled and verified using the System hardware description language for secured cryptography. The paper presents reconfigurable platform used with High-level language approach and the work presented here uses efficient implementation

of AES using Xilinx System Generator; the approach not only reduces the overall utilization but also gives good enough clock frequency and latency. Additionally, the hardware implementation of AES encryption algorithm provides final secrecy of the encryption key, with much faster speed compared to software implementation, and higher throughput by means of inherent hardware concurrency.

VI. REFERENCES

[1] Harshali Zodpe, Ashok Sapkal, "An efficient AES implementation using FPGA with enhanced security features", Journal of King Saud University - Engineering Sciences,2018,ISSN 1018-3639.

[2] Harshali Zodpe, Ashok Sapkal, "An efficient AES implementation using FPGA with enhanced security features", Journal of King Saud University - Engineering Sciences,2018,ISSN 1018-3639.

[3] Abdullah Al- Mamun, Shawon S. M. Rahman, Tanvir Ahmed Shaon and Md Alam Hossain,"Security Analysis Of Aes And Enhancing Its Security By Modifying S-Box With An additional Byte", International Journal of Computer Networks & Communications (IJCNC) Vol.9, No.2, March 2017.

[4] Abdullah Al- Mamun, Shawon S. M. Rahman, Tanvir Ahmed Shaon and Md Alam Hossain,"Security Analysis Of Aes And Enhancing Its Security By Modifying S-Box With An additional Byte", International Journal of Computer Networks & Communications (IJCNC) Vol.9, No.2, March 2017

[5] Yi Wang, Yajun Ha, "A Performance and Area Efficient ASIP for Higher-Order DPA-Resistant AES", IEEE Journal On Emerging And Selected Topics In Circuits And Systems, Vol. 4, No. 2, pp. 190-202, June-2014.

[6] Ahmed Alahmadi, Mai Abdelhakim, Jian Ren, and Tongtong Li, "Defense Against Primary User Emulation Attacks in Cognitive Radio Networks Using Advanced Encryption Standard", IEEE Transactions On Information Forensics And Security, Vol. 9, No. 5, pp. 772- 781, May-2014.

[7] Pallavi Atha et al, "Design & Implementation Of AES Algorithm Over FPGA Using VHDL", International Journal of Engineering, Business and Enterprise Applications (IJEBEA)", ISSN (Online): 2279-0039,pp. 58-62,2013

[8] Pachamuthu Rajalakshmi, "Hardware-software co-design of AES on FPGA" International Conference on Advances in Computing, Communications and Informatics, Pages 1118-1122, 2010

[9] Krishnamurthy G N and V Ramaswami, "Making AES Stronger: AES wich Key - Dependent S-Box", International Journal of Computer Science and Network Security,Vol. 8, No. 9, pp. 388-398, 2008.

[10] Xinmiao Zhang and Keshab K. Parhi "Implementation Approaches for the Advanced Encryption Standard Algorithm"IEEE 2002