Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 5, May 2017, pg.84 – 89

Hardware Implementation of Cryptosystem by AES Algorithm Using FPGA

Prachi V. Bhalerao[1], Rahul D. Ghongade[2], Vishal B. Langote[3]

[1]ExTC Department and SGBAU University, India
[2]H.O.D., ExTC Department and SGBAU University, India
[3]Professor, ExTC Department and SGBAU University, India
[1] prachigarimabhalerao@gmail.com; [2] rahulghongade@rediffmail.com; [3] vishal.langote29@gmail.com

Abstract— ***Advanced Encryption Standard (AES) is an approved cryptographic Algorithm that can be used to protect electronic data. The Advanced Encryption Standard can be programmed in software or built with hardware. However Field Programmable Gate Arrays (fpgas) offer a quicker, more customizable solution, hence we used the FPGA as for implementation purpose. We show how a modified structure in these Hardware devices results in significant improvement of the design efficiency. The conventional scheme of AES is vulnerable for cryptanalysis. Static S-Boxes are implemented using look up tables which will never vary with the input text or input key. This consumes a lot of Memory for the storage of look up table. Another weakness of AES is that it works with a single key. In this paper, a new scheme of AES that is triple key AES is proposed. This overcomes the vulnerability of static S-Boxes and also single key and dual key AES encryption scheme. Hence the triple key AES algorithm is more stronger as compared to the both previous cases and provide more security to the data, images etc.***

Keywords— "Cryptography, AES, FPGA, Static S-Box, Look up tables"

I. INTRODUCTION

In these days use of digital data exchange is increasing day by day in every field. Information security plays very important role in storing and transmitting the data. When we transmit a multimedia data such as audio, video, images etc. over the network, cryptography provides security. As we deal with Cryptography and Networking, the main aim is to achieve the security of the data. Hence, this paper presents "An Equivalent Security in Cryptosystem by Advance Encryption Standard Using FPGA". Advanced Encryption Standard (AES) is an approved cryptographic Algorithm that can be used to protect electronic data. AES is a symmetrical algorithm of encoding intended to replace DES which had already shown certain faults of safety in the data Protection. The Advanced Encryption Standard can be programmed in software or built with hardware. However Field Programmable Gate Arrays (FPGAs) offer a quicker, more customizable solution, hence we used the FPGA as for implementation purpose. We show how a modified structure in these Hardware devices results in significant improvement of the design efficiency. We move on to discuss about the recent modifications that have been done on the AES scheme and their weaknesses.

AES consists of 128 block length of bits and supports 128, 192 and 256 key length bits. The 128 bits are organized Into state matrix which is of the size of 4×4. This algorithm starts with initial transformation of state matrix Followed by nine iteration of rounds. A round consists of four transformations: Byte Substitution

(subbytes), Row Shifting (shiftrows), Mixing of columns (mixcolumns) and followed by addition of Round Key called (addroundkey). From each round, a round key is generated from the original key through key scheduling Process. The last round consists of subbytes, shiftrows and addroundkey transformation. Subbytes Transformation is implemented using S-Box.The S-Box is One of the most time consuming process because it is required in every round.

1.1 Conventional AES

AES is based on rijndael algorithm which is a symmetric block cipher that processes fixed data of 128-bit blocks. It supports key sizes of 128, 192 and 256 bits and consists of 10, 12 or 14 iteration rounds, respectively. In this paper we will focus on the 128-bit version with 10 rounds. Each round mixes the data with a round key, which is generated from the encryption key. Figure 1 illustrates the encryption round operations. The cipher maintains an internal, 4×4 matrix of bytes referred to as state, on which the operations are performed. Initially, state is filled with the input data block and exclusive-ored with the encryption key. Regular rounds consist of operations called subbytes, shiftrows, mixcolumns and addroundkey. Round key generation (key expansion) includes s-box substitutions, word rotations, and xor operations performed on the encryption key. Depending on the security level required for the application, AES uses different key lengths.

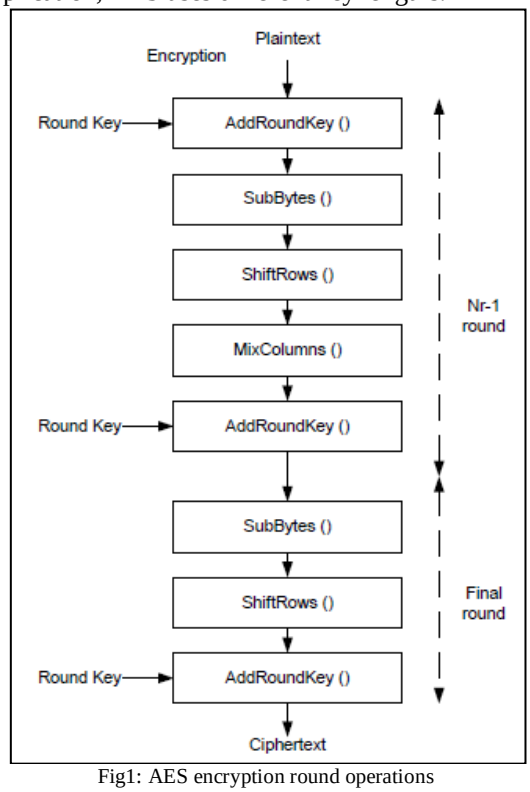

Fig1: AES encryption round operations

A. Shift row transformation: The rows of the state matrix are cyclically shifted but each row is shifted with a different offset. The offset of right shift varies from zero to three bytes. Since this transformation doesn't contribute to the complexity of the cipher text, we leave this transformation unchanged.

B. Mix column transformation: It is a linear byte substitution transformation. Here we replace the elements of the state matrix by mixing them with a constant matrix. The state matrix is multiplied with a constant matrix to obtain the new matrix. Matrix multiplication is done over galois field. In this transformation, the bytes are treated as a polynomials rather than numbers. The implementation complexity of this transformation is high and hence this algorithm is also used as it is.

C. Add round key: This transformation involves a bitwise xor operation between the state array and the resulting round key that is output of the key expansion algorithm.

D. Sub byte transformation: Sub byte transformation is a non-linear byte substitution transformation, unlike the mix columns where each element of the state matrix is replaced by a new element from a look up table. The main complexity of the algorithm lies in this transformation. The implementation of this transformation is very simple and hence all the modifications presented in this paper are on this transformation. The modifications increase the Cipher complexity by a huge volume and marginally increase the implementation complexity. All these operations are repeated 10 times in an AES-128 scheme The decryption scheme retraces the steps performing the inverse of every transformation to obtain the plain text.

II. LITERATURE REVIEW

From the rigorous review of related work and published literature, it is observed that many researchers have designed AES algorithm by applying different techniques. These related works have been mentioned as follows:

Abhiram.L.S, Gowrav.L, Punith Kumar.H.L & Sriroop.B.K, Manjunath.C.Lakkannavar gives the concept of dual key AES algorithm. In this paper, author presents a synthesizable algorithm which involves the use of conventional bitwise operations for generation of key dependent S-Boxes. Also a dual key based AES is presented in this paper which is FPGA implementable. The algorithm is reliable in terms of security and also suitable for hardware implementation. Mathematical analyses have been carried out on the algorithm to compute the reliability. The level of security provided by AES is usually measured in terms of time taken for cryptanalysis. The higher the time taken for cryptanalysis, the higher is the level of security offered. [1]

Milind Mathur & Ayush Kesarwani gives the comparison between DES, RC2 , RC6 , Blowfish & AES. This paper presents a performance evaluation of selected symmetric encryption algorithms. The selected algorithms are AES, DES, RC6, Blowfish and RC2. Several points can be concluded from the simulation results. There is no significant difference when the results are displayed either in hexadecimal base encoding or in base 64 encoding. In the case of changing packet size, it was concluded that Blowfish has better performance than other common encryption algorithms used, followed by RC6. In the case of changing data type such as image instead of text, it was found that RC2, RC6 and Blowfish has disadvantage over other algorithms in terms of time consumption. Finally in the case of changing key size it can be seen that higher key size leads to clear change in the battery and time consumption. [2]

Ankita Nampalliwar & Sheeja Suresh had design and implement AES algorithm using FPGA. In this paper author presented a hardware implementation increasing throughput for AES encryption algorithm. By using an efficient inter-round and intra-round pipeline design, they get the 20 clock cycles in 1 sec and hence achieved a throughput much higher than any other implementations reported in other literatures. [3]

Ahmed A. Mohamed & Ahmed H. Madian had given a modified Rijndael algorithm and its implementation using fpga. In this paper, a modified Rijndael algorithm that performs encryption process through three dependent stages is presented. A new mirror stage has been added to increase the complexity of algorithm as shown in fig 2. The modified algorithm has been realized and simulated using VHDL. The implementation of the modified algorithm has been done using FPGA Virtex XCV800. The simulation and implementation results have been summarized and discussed. [4]

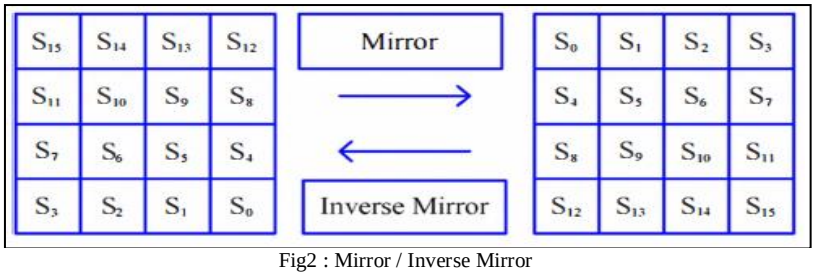

Fig2 : Mirror / Inverse Mirror

Kenneth Stevens & Otmane Ait Mohamed gives the concept of single-chip fpga implementation of a pipelined, memory-based AES Rijndael encryption design. In this paper, they presents a fully synchronous, memory-based, single-chip FPGA implementation of the recent AES Standard, Rijndael encryption algorithm. The dual-width encryption data path uses lookup table (LUT) architecture to perform encryption with internally generated round keys. Design partition allowed for an iterative loop structure where the block cipher was implemented using the Electronic Code Book mode of operation. Their encryption RTL design focuses on a memory-based, byte-sized arithmetic pipeline structure that processes one round at a time. The dual-width encryption pipeline permits the incorporation of a 32-bit DSP core into the byte-size data path while reducing latency issues associated with DSP cores having smaller bandwidths. [5]

III.PROPOSED METHODOLOGY

In the triple key AES algorithm, we have to encode the 128 bit data. For these three keys are used respective of the 128 bit. Following figure 3 shows the concept of triple key AES algorithm. It consists of several blocks which are used to do the encoding and decoding of data. In this system, 128 bit data is given as an input along with the three 128 bit keys.

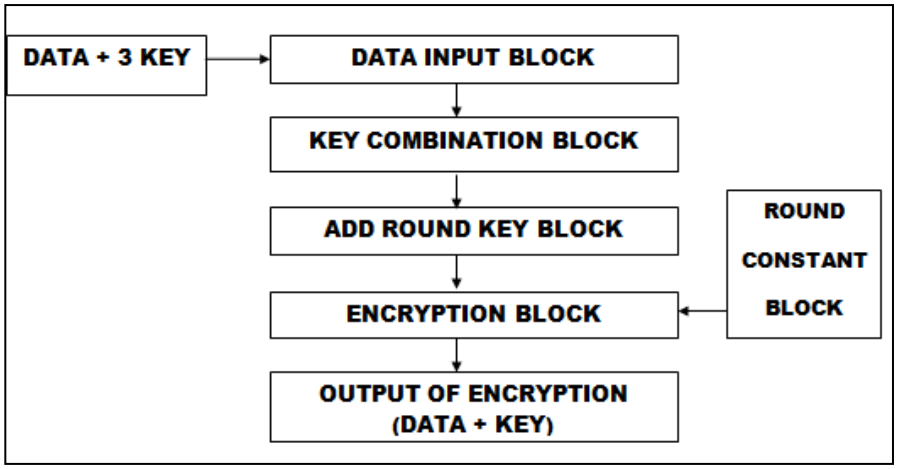

Fig 3: Triple Key AES Algorithm

First of all 128 bit data is given to the data input block along with the three 128 bit keys. These keys are provided to the key combination block. In this block the exoring of keys are performed to get the 128 bit output key. These output key is next given to the add round key block along with the 128 bit input data. In these block the conventional add round operation is performed that is the exoring of data and key is performed. And therefore the output of this block is forwarded to the actual encryption block and the key expansion round constants are also given to it. In this block all the transformation of conventional AES algorithm is performed. It means that transformations like substitute byte transformation, shift row transformation, mix column transformation and add round key transformation are performed. For 128 bit data we have to perform 10 rounds of encryption. After executing these transformations it will provide the final encoded 128 bit data and 128 bit key. For decryption of the data same process is followed in reverse order with the help of inverse transformations of the conventional AES algorithm.

IV. IMPLEMENTATION AND RESULT

This whole assembly is executed on the active hdl software with the help of VHDL programming. After that we have to simulate that algorithm in Xilinx and implemented on FPGA Spartan 3E for configurable hardware purpose. Following figure 4 shows the syntheses report of encryption of the triple key AES algorithm and figure 5 shows the syntheses report of decryption of the triple key AES algorithm From the following result it is clear that the system required less LUTs and logic slices as compared to the total LUTs and logic slices are available. Due to which the memory consumption occure is less. Hence it is very synthesizable.

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	45	9,312	1%
Number of occupied Slices	32	4,656	1%
Number of Slices containing only related logic	32	32	100%
Number of Slices containing unrelated logic	0	32	0%
Total Number of 4 input LUTs	60	9,312	1%
Number used as logic	45		
Number used as a route-thru	15		
Number of bonded IOBs	12	232	5%
Average Fanout of Non-Clock Nets	2.32		

Fig4: Syntheses Result of encryption

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	56	9,312	1%
Number of occupied Slices	32	4,656	1%
Number of Slices containing only related logic	32	32	100%
Number of Slices containing unrelated logic	0	32	0%
Total Number of 4 input LUTs	64	9,312	1%
Number used as logic	56		
Number used as a route-thru	8		
Number of bonded IOBs	140	232	60%
Average Fanout of Non-Clock Nets	2.13		

Fig5: Syntheses Result of decryption

TABLE I
COMPARISON OF 3AES WITH OTHER ALGORITHMS

Sr.No.	Factors	DES	3DES	AES	3AES
1	Encryption Time	24.5 Sec	74.71 Sec	4.14 Sec	8.414 Nsec
2	Decryption Time	24.5 Sec	74.71 Sec	4.41 Sec	8.186 Nsec
3	Throughput	0.244 Kb/Sec	0.08 Kb/Sec	1.448 Kb/Sec	3.084 Bit/Sec

From the above comparison table 1, it has been analyzed that 3AES algorithm takes less time compared to other algorithms. Throughput varies inversely to the encryption or decryption time.

V. HARDWARE OUTPUT

After synthesizing the triple key AES algorithm on Xilinx, now it will be implemented on FPGA Spartan 3e board. As we encode or decode 128 bit data, it is not possible to implement on fpga completely. Hence we can implement complete data in the form of 8 bit on the Spartan 3E fpga board. Following figure 6 and figure 7 shows some of images out of complete implementation of encryption and decryption of 128 bit data respectivly. As we operate the switches on the board regarding to set the bits from "0 H to F H" it will show the respective output, as the LEDs are glowing on the board.

Fig 6: Encryption of Data

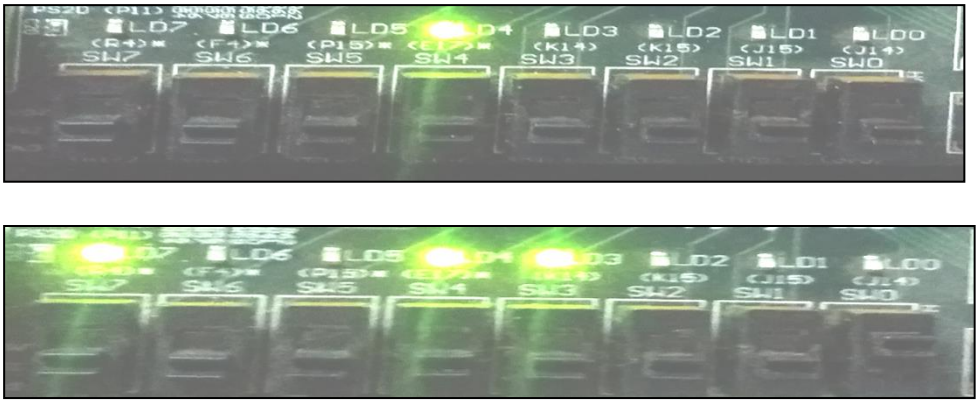

Fig7: Decryption of Data

VI. CONCLUSIONS

Many of modification have been done on AES encryption. In this paper, the architecture of the algorithm for optimal FPGA implementation is also implemented. Also a triple key based AES is presented in the paper which is FPGA implementable. The algorithm is reliable in terms of security and also suitable for hardware implementation. The triple keys lead in increasing the complexity degree within the same time during the encryption and decryption processes. The results show that the present proposed algorithm has good cryptographic strength, with the added benefit that is resistant to linear and differential cryptanalysis, which

require that the S-boxes be known beside the encryption key. Also the performance evaluation of different algorithm is shown, hence it concluded that 3AES is more secure as compared to other.

REFERENCES

[1] Abhiram.L.S, Gowrav.L, Punith Kumar.H.L & Sriroop.B.K, Manjunath.C.Lakkannavar, "Design and synthesis of Dual Key based AES Encryption", MSRIT, Bangalore, India, 21-22 November 2014 978-1-4799-6546-5/14/$31.00©2014 IEEE.

[2] Milind Mathur & Ayush Kesarwani, "Comparison Between DES , 3DES , RC2 , RC6 , BLOWFISH and AES National Conference on New Horizons in IT - NCNHIT 2013 ISBN 978-93-82338-79-6.

[3] Ankita Nampalliwar & Sheeja Suresh , "Design and Implementation of AES Algorithm Using FPGA" ISSN: 2321-7782 (Online) ,Volume 2, Issue 1, January 2014 ,International Journal of Advance Research in Computer Science and Management Studies.

[4] Ahmed A. Mohamed & Ahmed H. Madian, "A Modified Rijndael Algorithm and its Implementation using FPGA 978-1-4244-8157 ©2010 IEEE, ICECS 2010.

[5] Kenneth Stevens & Otmane Ait Mohamed, "Single-chip FPGA Implementation of a Pipelined, Memory-Based AES Rijndael Encryption Design" 0-7803-8886 ©2005 IEEE CCECE/CCGEI, Saskatoon, May 2005.

[6] Spartan-3E FPGA Starter Kit Board User Guide.

[7] William Stallings, "Cryptography and Network Security Principles and Practice", Sixth Edition.

