Study and Literature Survey of Advanced Encryption Algorithm for Wireless Application

Vedkiran Saini, Parvinder Bangar, Harjeet Singh Chauhan

Abstract— ***Today increases any wireless communication security is crucial during data transmission. The encryption and decryption of data is the main challenge faced in the wireless communication for security of the data transmission source to destination. In this paper we present the literature study of cryptography security AES algorithm and its present application in communication, data communication and wireless communication. In this paper, we use the Advanced Encryption Standard (AES) which works on a 128 bit data encrypting it with 128 bits of keys for ensuring security. In this paper literature study of AES algorithm and selection AES algorithms for wireless communication application and design verilog AES sub block add round key, mix column, s-box using Xilinx ISE 9.1i software for a Spartan3 FPGA device***

Index Terms— ***Advanced Encryption Standard (AES), Rinjdael, Cryptography,***

I. INTRODUCTION

In today's digital world, encryption is emerging as a disintegrable part of all communication networks and information processing systems, for protecting both stored and in transit data world .Security has become an increasingly important feature with the growth of complex electronic communication. The AES also known as the Rijndael algorithm was selected as a Standard by National Institute of Standards and Technology (NIST).Advanced Encryption Standard (Rijndael Block Cipher) became the new US Federal Information Processing Standard on November 26, 2001[1,23] in order to replace the Data Encryption Standard (DES) which was used for more than 20 years as a common key block cipher for FIPS.Encryption is the transformation of plain data (known as plaintext) into unintelligible data (known as cipher text) through an algorithm referred to as cipher. Encryption is the transformation of data into a form that is as close to impossible as possible to read without the appropriate knowledge (a key). Its purpose is to ensure privacy by keeping information hidden from anyone for whom it is not intended, even those who have access to the encrypted data.. Cryptography has a main role in embedded systems design. Today technology going on deep submicron technology as the number of devices and applications which send and receive data are increasing rapidly the data transfer rates are becoming higher. In many applications, this data requires a secured connection which is usually achieved by cryptography. Cryptography is divided in two categories first is symmetric key and second is asymmetric key. Symmetric key cryptography sender and receiver share the same key and asymmetric key cryptography sender and receiver shares different keys. Figure 1 shown the general encryption and decryption model.

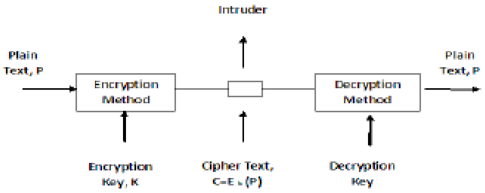


Figure 1: The Encryption Model [16]

Public key encryption algorithm is a fundamental and widely using technology around the world. But it has hardware limitations as like memory and battery, so now days the portable electronics and embedded has made low power design extremely desirable it is not applied to the embedded system because embedded system low power main characteristics and portability for hardware complexity high [2].application of Smart card PDA, wireless security like military communication, bluetooth and mobile telephony where there is a greater emphasis on the speed of communication.

II. LITERATURE REVIEW

Xinmiao Zhang et al (2002) 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. In non-feedback mode, sub pipelining can achieve maximum speed up and the best speed/area ratio. Algorithmic optimization exploits algorithmic strength inside each round unit. Various methods to reduce the critical path and area of each round unit are presented. Resource sharing issues between encryptor and decryptor are also discussed. They become important issues when both encryptor and decryptor need to be implemented in a small area [1].P.Prasithsangaree et al (2003) have presented and analyzed the Energy Consumption of RC4 and AES Algorithms in Wireless LANs.

Manuscript received on April 16, 2014.

Ms. Vedkiran Saini, Department of ECE, CBS Group of Institution, Jhajjar, India.
Mr. Parvinder Bangar, CBS Group of Institution, Jhajjar, India.
Prof. Harjeet Singh Chauhan, Registrar, BM Group of Institution, Gurgaon, India.

The performance metrics were encryption throughput, CPU work load, energy cost and key size variation. Experiments show that the RC4 is fast and energy efficient for encrypting large packets and AES was more efficient than RC4 for a smaller packet size. From the results, it appears that we can save energy by using a combination of RC4 and AES to provide encryption for any packet size.[11]Pachamuthu Rajalakshmi et al (2010) 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 [6].Archna Garg et al (2013) has presents an efficient FPGA implementation approach of the Advanced Encryption Standard (AES) Algorithm. In this paper they present two different architectures of AES named Basic AES and Fully Pipelined AES have been designed in VHDL. The codes have been synthesized using Xilinx ISE 9.2i software for a Virtex 3 FPGA device. AES and Fully Pipelined AES algorithm result comparison on the basis of power consumption, Maximum pin Delay, Clock delay, Slice Flip flops.[8]Pallavi Atha et al (2013) 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.[9]M. komala subhadra et al (2013) have proposes an efficient FPGA implementation of AES using VHDL. An AES encryptor is designed and implemented in FPGA. An AES decryptor is also designed and integrated with the AES encryptor to yield a full functional AES en/decyptor. Xilinx software is used for the simulation and optimization of the synthesizable VHDL code. All the transformations of both Encryption and Decryption are simulated using an iterative design approach in order to minimize the hardware consumption.[10]

III. ADVANCED ENCRYPTION STANDARD

The AES (advanced encryption standard) [3] is an encryption standard as a symmetric block cipher. It was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) on November 26,2001. The Rijndael algorithm was developed by Joan Daemen of Proton World International and Vincent Fijmen of Katholieke University at Leuven. The AES consists of mainly two units which are Data processing unit and the other one is Key Expansion unit. The Data processing units have four main modules or transformations in which sub byte transform, shift rows, mix column and add round key are involved and the Key Expansion unit generate the round key for the next round. The AES operates on 128-bit blocks of data. The algorithm can encrypt and decrypt blocks using secret keys. The key size can either be 128 bit, 192 bit, or 256 bit. The actual key size depends on the desired security level. The different versions are most often denoted as AES-128, AES-192 or AES-256.

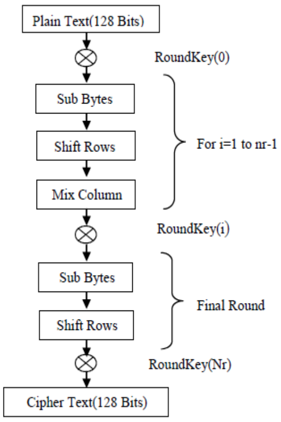


Figure 2 AES Algorithm Encryption structure[19]

Table1. Key-Block-Round Combinations [19]

	Key Length (32-bitword)	Block Size (32-bit word)	Number of Rounds
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

IV. TRANSFORMATION IN AES ALGORITHM

A. Sub byte transformation

The sub byte transform is shown in figure 3.In AES algorithm the function of the sub byte is only nonlinear function and that operates independently on each byte of the state using a substitution table (Sbox). It substitutes all bytes of the state array using a LUT which is a 16x16 matrix of bytes, often called S-box. In AES hardware implementation, S-box design contributes a major role in optimization two approaches for S-box design. Design a multiplicative inversion and affine transformation separately or Construct a logic circuit defining the input and output of the S-box function [8].

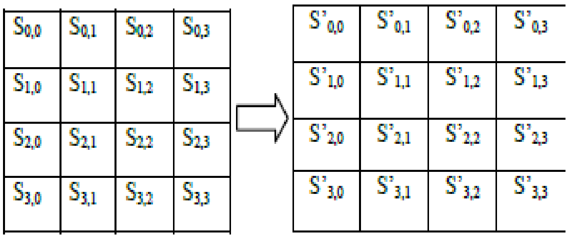
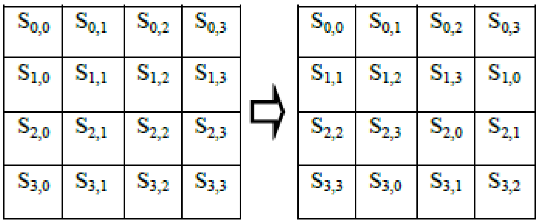

$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S'_{0,0}$ $S'_{0,1}$ $S'_{0,2}$ $S'_{0,3}$
$S'_{1,0}$ $S'_{1,1}$ $S'_{1,2}$ $S'_{1,3}$
$S'_{2,0}$ $S'_{2,1}$ $S'_{2,2}$ $S'_{2,3}$
$S'_{3,0}$ $S'_{3,1}$ $S'_{3,2}$ $S'_{3,3}$

Figure3Application of S-box to the Each Byte of the State

B. Shift Row Transformation

In AES algorithm the function of the sub byte is only nonlinear function and that operates independently on each byte of the State using a substitution table (Sbox). It substitutes all bytes of the state array using a LUT which is a 16x16 matrix of bytes, often called S-box. The Shift row transform is shown in figure 4 the shift row transformation. Transformation same as Transformation is almost the same in the decryption process except that the shifting offsets have different values. The main goal of this process is to correlate and scramble the byte order inside each 128-bit block. In the shift the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes (offsets).in this process the row 0 is not shifted, row0 is shifted one byte to the left, row 2 is shifted two bytes to the left and row 3 is shifted three bytes to the left.



$$\begin{array}{|c|c|c|c|}\hline S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ \hline S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ \hline S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ \hline S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|c|}\hline S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ \hline S_{1,1} & S_{1,2} & S_{1,3} & S_{1,0} \\ \hline S_{2,2} & S_{2,3} & S_{2,0} & S_{2,1} \\ \hline S_{3,3} & S_{3,0} & S_{3,1} & S_{3,2} \\ \hline \end{array}$$

Figure 4 Shift Row Transformation

C. Mix Column Transformation

This transformation is based on Galois Field multiplication. Each byte of a column is replaced with another value that is a function of all four bytes in the given column. The Mix Columns transformation is performed on the State column-by column [5].The mix column implementation is shown in figure 5. Each column is considered as a four-term polynomial over GF(2^8) and multiplied by $a(x)$ modulo $x4$ + 1, Where

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

This Transformation is for mixing up of the bytes in each column separately during the forward process. The corresponding transformation during decryption is denoted Inv Mix Columns and stands for inverse mix column transformation. The goal is here is to further scramble up the 128-bit input block.

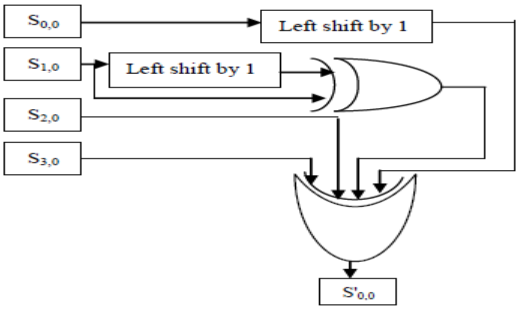


Figure 5 Mix Column Transformations in Matrix Form [19]

D. Add Round key and key expansion

In this operation, the round key is applied to the State by simple bit by bit XOR. Key Expansion unit generates the next round key as for three different key size, AES consist of 10, 12 or 14 rounds. After every round a new round key is produced. This process utilizes the concept of shifting the bytes and substitution of bytes which were used in Data processing unit.

i. Add Round key

Add Round Key step is applied one extra time comparing to the other encryption steps. The first Add Round Key step is applied before starting the encryption iterations, where in the encryption process the first 128 bits of the input key the whole key in case of using key size of 128 bits are added to the original data block as shown in figure 6. This round key is called the initial round key [4]. It is implemented in hardware as a simple exclusive-or operation of the 128 bit data and key.

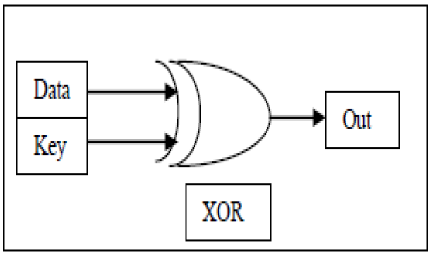


Figure 6 Hardware Implementation of Add Round Key

ii. Key expansion

The key expansion term is used to describe the operation of generating all Round Keys from the original input key. The initial round key will be the original key in case of encryption the whole operation is shown in figure 7.

The key expansion term is used to describe the operation of generating all Round Keys from the original input key. The initial round key will be the original key in case of encryption and the last group of the generated key expansion keys in case of decryption – the first and last 16 bytes in case of key sizes of 192 and 256 bits. As mentioned previously this initial round key will be added to the input initially before starting the encryption or decryption iterations. Using the 128 bits key size, 10 groups of round keys will be generated with 16 bytes size for each.

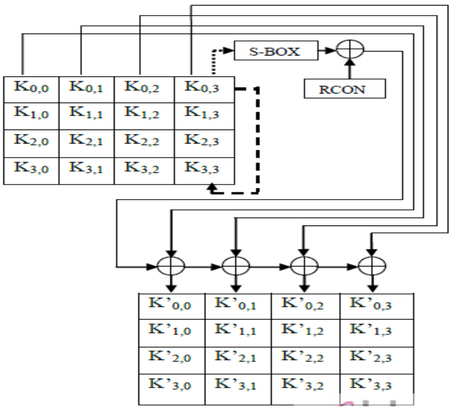


Figure7key expansion[19]

The first 4 bytes column in each group will be generated as follows:

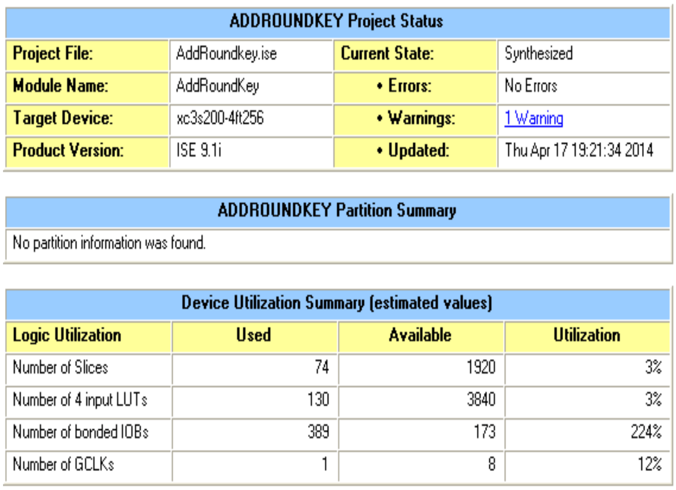
- Taking the S-BOX equivalent to the last column of the previous group (one previous column).
- Perform one cyclic permutation “rotate elements
- Add the round constant.
- Add the result to the first column of the previous

group (four previous columns).

V. SIMULATION RESULT

In this paper design AES algorithm sub block Add round key, Mix column block and S box have been designed in VHDL. The codes have been synthesized using Xilinx ISE 9.1i software for a Spartan3 FPGA device.

Table 1 show the summary of resources utilized Add round key

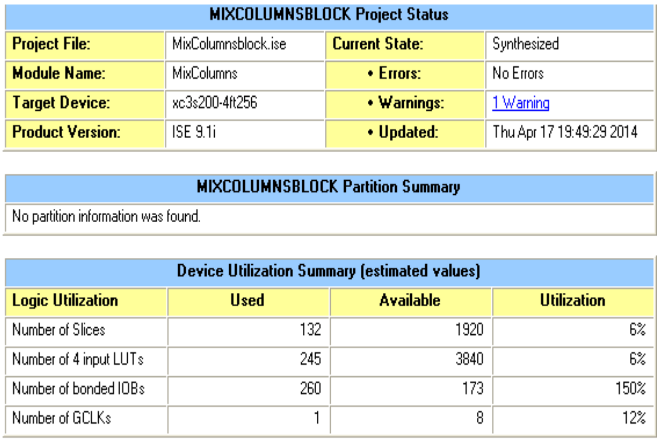


ADDROUNDKEY Project Status			
Project File:	AddRoundkey.ise	**Current State:**	Synthesized
Module Name:	AddRoundKey	**• Errors:**	No Errors
Target Device:	xc3s200-4ft256	**• Warnings:**	1 Warning
Product Version:	ISE 9.1i	**• Updated:**	Thu Apr 17 19:21:34 2014

ADDROUNDKEY Partition Summary
No partition information was found.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	74	1920	3%
Number of 4 input LUTs	130	3840	3%
Number of bonded IOBs	389	173	224%
Number of GCLKs	1	8	12%

Table 2 Shows the summary of resources utilized Mix column

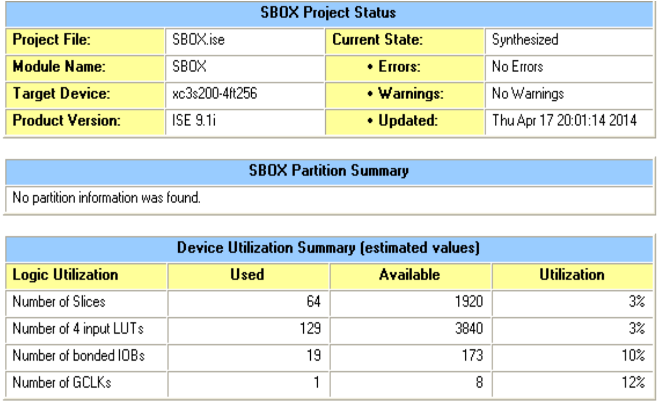


MIXCOLUMNSBLOCK Project Status			
Project File:	MixColumnsblock.ise	**Current State:**	Synthesized
Module Name:	MixColumns	**• Errors:**	No Errors
Target Device:	xc3s200-4ft256	**• Warnings:**	1 Warning
Product Version:	ISE 9.1i	**• Updated:**	Thu Apr 17 19:49:29 2014

MIXCOLUMNSBLOCK Partition Summary
No partition information was found.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	132	1920	6%
Number of 4 input LUTs	245	3840	6%
Number of bonded IOBs	260	173	150%
Number of GCLKs	1	8	12%

Table 3Show the summary of resources utilized S box



SBOX Project Status			
Project File:	SBOX.ise	**Current State:**	Synthesized
Module Name:	SBOX	**• Errors:**	No Errors
Target Device:	xc3s200-4ft256	**• Warnings:**	No Warnings
Product Version:	ISE 9.1i	**• Updated:**	Thu Apr 17 20:01:14 2014

SBOX Partition Summary
No partition information was found.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	64	1920	3%
Number of 4 input LUTs	129	3840	3%
Number of bonded IOBs	19	173	10%
Number of GCLKs	1	8	12%

VI. CONCLUSION

In this paper present the literature survey and study of AES algorithm for high speed and wireless communication application and in this paper also study the process of sub byte transformation, shift row transformation, mix column transformation and add round key and key expansion. In this paper the existing AES algorithm studied and analyzed well to promote the performance of the encryption methods also to ensure the security proceedings. In this paper design AES sub block add round key, Mix column and S-box.

REFERENCES

1. Xinmiao Zhang and Keshab K. Parhi “Implementation Approaches for the Advanced Encryption Standard Algorithm”IEEE 2002
2. X. Zhang and K. K. Parhi, “High-speed VLSI architectures for the AES algorithm,”IEEE Transactions on Very Large Scale Integration Systems, vol.12, issue 9, pp.95 967, Sep. 2004.
3. Hui QIN, Tsutomu SASAO, Yukihiro IGUCHI “An FPGA Design of AES Encryption Circuit with 128-bit Keys”GLSVLSI’05, ACM 2005.
4. Ashwini M. Deshpande, Mangesh S. Deshpande and Devendra N. Kayatanavar “FPGA Implementation of AES Encryption and Decryption” International Conference on Control,Automation, Communication and Energy conservation -2009
5. Chih-Peng Fanand and Jun-Kui Hwang “FPGA Implementations Of High Throughput Sequential And Fully Pipelined AES Algorithm” International journal of Electrical Engineering, vol.15, no.6, pp. 447-455, 2008.
6. Pachamuthu Rajalakshmi, “Hardware-software co-design of AES on FPGA” International Conference on Advances in Computing, Communications and Informatics, Pages 1118-1122, 2010.
7. Mehran Mozaffari-Kermani and Arash Reyhani-Masoleh “Efficient and High Performance Parallel Hardware Architecture for the AES-GCM” IEEE Transactions On Computers, vol.61, no. 8, August 2012.
8. Saambhavi Baskaran and Pachamuthu Rajalakshmi “Hardware Software Co-Design of AES on FPGA” ICACCI ’12,ACM August 2012.
9. 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
10. M. komala subhadra et al, “Advanced Encryption Standard - VHDL Implementation”, International Journal For Technological Research In Engineering, ISSN (Online): 2347 - 4718, Volume 1, Issue 3, pp.132-137 November – 2013.
11. Prasithsangaree.P and Krishnamurthy.P(2003), “Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs,” in the Proceedings of the IEEE GLOBECOM, pp. 1445-1449, 2003.
12. Yoshimura, M. et al, “Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)”, IEEE International Symposium on Page(s):278 – 283, 2013
13. Hui QIN, Tsutomu SASAO, Yukihiro IGUCHI “An FPGA Design of AES Encryption Circuit with 128-bit Keys” GLSVLSI’05, ACM 2005
14. Chih-Peng Fanand and Jun-Kui Hwang “FPGA Implementations of High Throughput Sequential and Fully Pipelined AES Algorithm” International journal of Electrical Engineering, vol.15, no.6, pp. 447-455, 2008.
15. Mehran Mozaffari-Kermani and Arash Reyhani-Masoleh“Efficient and High Performance Parallel Hardware Architecture for the AES-GCM” IEEE Transactions On Computers, vol.61, no. 8, August 2012.
16. Archna Garg et al, “Efficient Field Programmable Gate ArrayImplementation of Advanced Encryption Standard Algorithm using VHDL”, International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 9, pp. 3956-3961,September 2013
17. Saambhavi Baskaran and Pachamuthu Rajalakshmi “Hardware Software Co-Design of AES on FPGA” ICACCI ’12,ACM August 2012.

18. Ashwini M. Deshpande, Mangesh S. Deshpande and Devendra N. Kayatanavar "FPGA Implementation of AES Encryption and Decryption" International Conference on Control, Automation, Communication and Energy conservation -2009.
19. Richa Sharma, Purnima Gehlot, S. R. Biradar, "VHDL Implementation of AES-128, UACEE International Journal of Advances in Electronics Engineering – IJAEE, Volume 3 : Issue 2, [ISSN 2278 – 215X],pp-17-20, 2013
20. X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm,"IEEE Transactions on Very Large Scale Integration Systems, vol.12, issue 9, pp.95 967, Sep. 2004.
21. Jin Gong ,Wenyi Liu, Huixin Zhang "Multiple Lookup Table- Based AES Encryption Algorithm Implementation" Elseveir- 2012 vol.25 pg no.842 – 847.
22. Biham, Eli and Adi Shamir, Differential Cryptanalysis of the Data Encryption Standard, Springer Verlag, 1993.
23. National Institute of Standards and Technology, "Federal Information Processing Standards Publication 197", 2001
24. Jin Gong ,Wenyi Liu, Huixin Zhang "Multiple Lookup Table- Based AES Encryption Algorithm Implementation" Elseveir- vol.25 pg no.842 – 847, 2012.

AUTHORS PROFILE

Ms. Ved Kiran Ms. Ved Kiran has completed her B.tech in Electronics and communication from Dronacharya college of Engineering Gurgaon and Pursuing her M.Tech in Electronics & communication Engineering from MITS, Jhajjar, India. Her research interest includes network security and privacy

Mr. Parvinder Bangar is a HOD in the department of electronics & communication engineering at CBS Group of Institution, jhajjar, India. He received his B.E and M.Tech degrees in Electronics and Communication Engineering.

Prof. Harjeet Singh Chauhan is Working as a Professor at BM Group of Institution, Gurgaon . he have around 13+ years of teaching experience. He had obtained his PhD in reliability engineering in year 0f 2003. He have presented 15 national and international journal and conferences.