Design of AES-512 Algorithm for Communication Network

Ankit K.Dandekar[1], Sagar Pradhan[2], Sagar Ghormade[3]

[1]Research Scholar, Department of ECE, Abha Gaikwad Patil College of Engineering, Nagpur, India
[2]Asst. Professor, Department of ECE, Abha Gaikwad Patil College of Engineering, Nagpur, India
[3] H.O.D., Deptartment of ECE, Abha Gaikwad Patil College of Engineering, Nagpur, India

---***---

Abstract - *The main objective of this paper is to provide stronger security for communication network over the Internet by enhancing the overall strength of the AES algorithm. Rijndael's algorithm was been selected as the Advanced Encryption Standard. The AES algorithm provides much more security without any limitations. But, recently some breaking methods on the AES have been found by cryptanalyst. For overcome this problem number of rounds in AES algorithm need to be increased. In AES algorithm, encryption and decryption involving the number of rounds depends on the length of the key and the number of block columns. So, to improve the strength of the AES the number of rounds is increased. By increasing the key length to 512 bit the strength of the AES algorithm is enhanced and in order to provide a stronger encryption method for secure communication the number of rounds is increased. In order to improve the speed of encryption and decryption code optimization is also done using the 512 bit AES.*

Key Words: **AES algorithm, Cryptography, Decryption, Encryption**

1.INTRODUCTION

Network security is becoming much more important as people spend much more time connected in a network. To protect the value and ongoing usability of assets, the integrity and continuity of operations it involves all activities that institutions, enterprises and organizations undertake. Security attacks include modification of messages or files, denial of service, traffic analysis and unauthorized reading of a message of file. Computer virus is one of the most publicized types of attack on information systems. An effective network security strategy requires identification of threats and then choosing the most effective set of tools to overcome them. Security involving communications and networks is not as simple as it might first appear. The expansion of the connectivity of computers makes various ways of protecting data and messages from tampering and reading. Intruders may reveal the information to an individual or organization, use it to launch an attack or modify it to misrepresent. One of the basic reasons that intruders can be successful in causing threat is that most of the information they can acquire from the system is in a form that they can read and comprehend. And one solution to this problem is by using cryptography. Cryptography ensures that the messages could not be intercepted or read by anyone other than the authorized recipient. It prevents intruders from being able to use the information that can be acquired. Thereafter, cryptography secures information by protecting its confidentiality and can also be used to protect authenticity of data and information about the integrity.

1.1 Related Work

The first encryption algorithm, Data Encryption Standard (DES) was adopted by the National Institute of Standards and Technology (NIST) to protect the confidential and sensitive information as Federal Information Processing Standard 46 (FIPS PUB 46) in 1977. However, the security of DES was reduced due to shorter length of key, existence of weak and semi-weak keys and the complementary property. Differential cryptanalysis attack is capable of breaking DES in less than 255 complexities. For differential cryptanalysis the linear cryptanalysis method can find a DES key given 243 known plain texts, as compared to 247 chosen plain texts. So, to replace the DES it was more essential to find a stronger encryption algorithm. There has been considerable interest in finding an alternative in spite of the vulnerability of DES to a brute-force attack. One approach would be to design a completely new algorithm and another alternative would be the one that preserves the existing algorithm by using multiple encryptions with DES and multiple keys. To solve the problems of DES three other algorithms were found. They are Double DES, Triple DES with two keys and Triple DES with three keys. The main drawback of Triple DES is that it has three times as many rounds as DES and hence it is much slower. Another drawback of Triple DES is it uses a 64 bit block size, because for both efficiency and security a larger block size is needed. These drawbacks of Triple DES are not favorable for long term use. The Rijndael algorithm was being adopted as an encryption standard, the Advanced Encryption System (AES) by the NIST as FIPS PUB 197 (FIPS 197) on November 2001. The AES algorithm was designed to provide more security than the DES. The AES algorithm was believed to have resistance against all known attacks, speed and code compactness on a wide range of platforms and design simplicity. AES has three variable key lengths but block length is fixed to 128 bits. The three key sizes of AES are 128, 192 and 256 bits. For an exhaustive key search AES with 128-bit keys has stronger resistance than DES.

1.2 Drawbacks of AES-256

Rijndael has a very strong resistance against the differential cryptanalysis and linear cryptanalysis attacks since it uses Wide Trail Strategy in its design. Although these linear attacks are invalid for the AES, from recent years they have been expanded in various ways and new attacks have been introduced that are relative to them. In 2005, E. Biham introduced the new attack that is combination of boomerang and the rectangle attack with related-key differentials. It can break some reduced-round versions of AES and can uses the weakness of few non linear transformations in the key schedule algorithm of ciphers. By using 256 different related keys it can break 192-bit 9-round AES. Rijndael exhibits many properties from Square algorithm. Thereafter, the Square attack is also valid for Rijndael which can break round reduced variants of Rijndael up to 6 rounds for AES-128 or 7 rounds for AES-192 faster than an exhaustive key search. The work factor of the attack was reduced due to some optimizations that were proposed by N. Ferguson. So, this attack breaks a 256-bit 9-round AES with 277 plain texts under 256 related keys and 2224 encryptions.

1.3. Existing Work

Many implementations were proposed and implemented previously they are 128,192,256 bit. There various implementation for AES support the fact that for the same algorithm different application required different implementation. Some application needs strict area requirement and a compact AES implementation will be very useful to provide security as in the some embedded system cases. On the other hand, some application highly needed the higher level of security that can be obtained without caring about the area and time limitation.

1.4. Proposed Work

This paper shows the variation of AES algorithm called as 512 bit. The aim is to present that AES-512 bit can be used when higher level of security throughput are required without increasing overall design area as compared to the original 128 bit AES algorithm. The new algorithm consist of the structure which is similar to original AES algorithm but having slight difference that is instead of using 128 bit the plain text size and key size uses input of 512 bit that has impact on the whole algorithm structure. The AES algorithm consists of four major operations are performed during each round: byte substitution, shifting rows, mixing columns and adding the round key. AES 128 bit key is considered to be secure compared to other existing symmetric cipher algorithm. It is widely used in many applications where the security is most important. The new AES algorithm provides even more security and double throughput. More security comes from using larger key size and more throughputs come from using four times larger block size that the block size used in the original AES. The only disadvantage of AES 512 is the need for more design area.

The proposed AES 512 algorithm has four main different byte based transformation. The first transformation is the byte substitution which substitutes the value of 512 bit and this is achieved by using parallel s-boxes. The second transformation is shifting rows that shift the rows of the output from previous step by an offset equal to the row numbered. The third transformation is mixing column, where each column of the output from previous step is multiplied by different value. The final transformation in the round is adding round key to the result of this round.

2. AES-512 ARCHITECTURE

The top level architecture of the AES-512 bits is shown in Figure 1. The plaintext and the key size are 512-bits each (organized in bytes). The AES-512 algorithm processes the data in 10 rounds. The key and the input data are loaded when the Loadkey control signal is one and zero, respectively. The Encrypt signal starts the encryption process, while reset resets everything to zero. The resulting cipher text is also 512-bits. More details about each of the transformations used in the AES-512 are described in the coming subsections. Where the key expansion procedure is explained later since each round needs its own key generated according to this procedure.

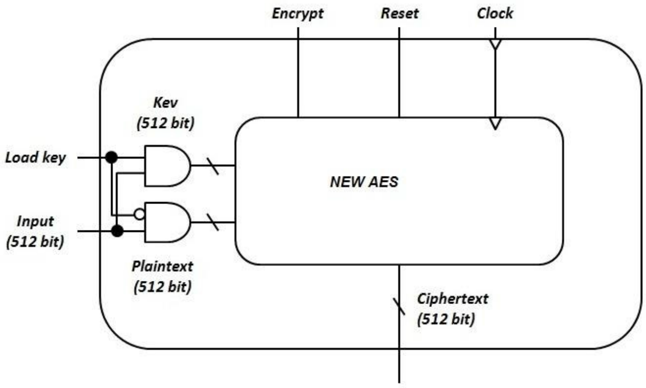

Fig -1: Top level of architecture AES-512

2.1 Implementation of Encryption

AES operates on a 16×32 array of bytes, termed the state. The input key for encryption is 512 bits. To represent the 512 values 9 bits are required. So each entry in S-box of AES 512 is 9 bits long. The cipher is specified in terms of repetitions of processing steps that are applied to make up rounds of keyed transformations between the input plain-text and the final output of cipher-text. The encryption procedure of AES 512 has been illustrated in figure 2.Each round in AES 512 encryption includes four different round transformations namely Substitute Bytes, Shift Rows, Mix Columns and Add Round Key. The last round of AES 512 encryption alone does not include the Mix Columns transformation.

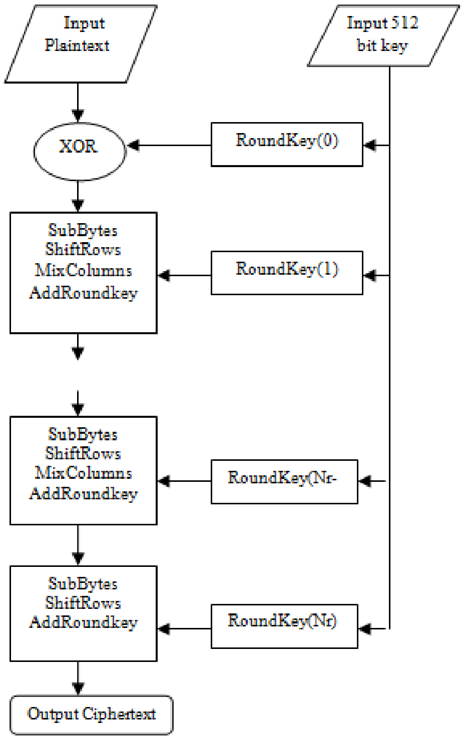
Input Plaintext
Input 512 bit key
XOR
RoundKey(0)
SubBytes ShiftRows MixColumns AddRoundkey
RoundKey(1)
SubBytes ShiftRows MixColumns AddRoundkey
RoundKey(Nr-
SubBytes ShiftRows AddRoundkey
RoundKey(Nr)
Output Ciphertext

Fig -2: Encryption procedure of AES 512

2.2 Implementation of Encryption

A set of reverse rounds are applied to transform cipher-text back into the original plain-text using the same encryption key. The four reverse transformations used are Add Round Key, Inverse Mix Columns, Inverse Shift Rows and Inverse Substitute Bytes. The inverse S-box contains 512 values in its 16x32 array of bytes. Each round in decryption of AES 512 includes all the four reverse transformations except in the first round. The Inverse Mix Column transformation is violated in the first round of decryption since it does not occur in the last round of encryption. The decryption procedure of AES 512 is illustrated in figure 3.

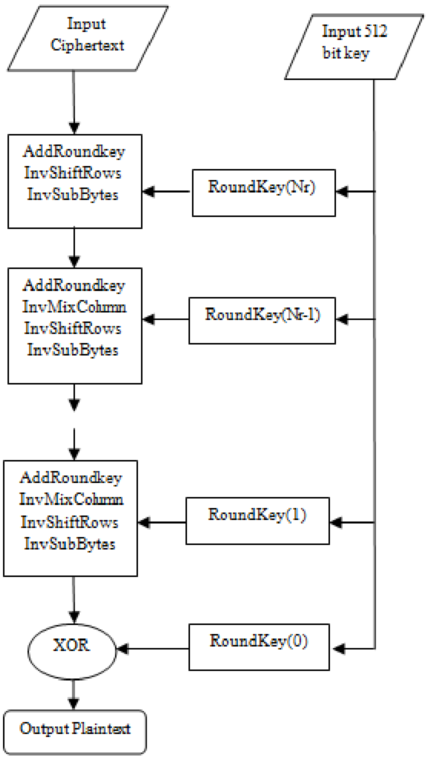
Input Ciphertext
Input 512 bit key
AddRoundkey InvShiftRows InvSubBytes
RoundKey(Nr)
AddRoundkey InvMixColumn InvShiftRows InvSubBytes
RoundKey(Nr-1)
AddRoundkey InvMixColumn InvShiftRows InvSubBytes
RoundKey(1)
XOR
RoundKey(0)
Output Plaintext

Fig -3: Decryption procedure of AES 512

2.3. Stages in Encryption and Decryption

2.3.1 Byte Substitution

The 512-bits input plaintexts are organized in array of 64-bytes and are substituted by values obtained from Substitution boxes. This is done (as in the original AES) to achieve more security according to diffusion-confusion Shannon's principles for cryptographic algorithms design. To overcome the overhead of the huge data size used (512-bits), the Substitution boxes are implemented as lookup tables, and accessed in parallel as shown in Figure 4.

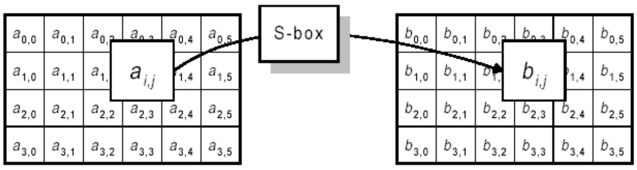
$a_{i,j}$
S-box
$b_{i,j}$
$a_{0,0}$ $a_{0,1}$ $a_{0,4}$ $a_{0,5}$
$a_{1,0}$ $a_{1,1}$ $a_{1,4}$ $a_{1,5}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$ $a_{2,4}$ $a_{2,5}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$ $a_{3,4}$ $a_{3,5}$
$b_{0,0}$ $b_{0,1}$ $b_{0,4}$ $b_{0,5}$
$b_{1,0}$ $b_{1,1}$ $b_{1,4}$ $b_{1,5}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$ $b_{2,4}$ $b_{2,5}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$ $b_{3,4}$ $b_{3,5}$

Fig -4: Byte Substitution

2.3.2 Shift Row

After the original 512-bit data is substituted with values from the S-boxes, the rows of the resulting matrix are shifted in a process called Shift Row transformation. What happened in this part is that the bytes in each row in the input data matrix will be rotated left. The number of left rotations is not the same in each row, and it can be determined by the row number. For example, row number zero is not shifted; the first row is shifted by one byte, and so on.

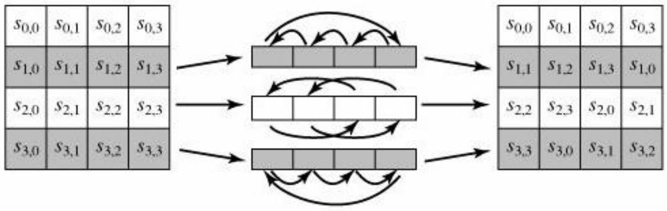
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

Fig -5: Shift Row

2.3.3 Mix Column

Now, and after the rows of the input data are rotated left by different offsets, an operation must be applied to the columns of the data matrix. The Mix Column transformation multiplies the columns of the data matrix by a pre-defined matrix. The AES-512 and original AES process the data in bytes basis. Each byte is considered as polynomials over GF (2^8) with 8 terms. To explain how the Mix Column works, we have to explain the concept of polynomials over GF (2^n) in general and for GF (2^8) as example when n=8.

The conversion might be dictated by the accompanying grid increase on state demonstrated in fig 7.3.3. Every component of the item framework is the entirety of results of components of one line and one segment. For this situation the unique augmentations & multiplication are achieved in GF (2^8).

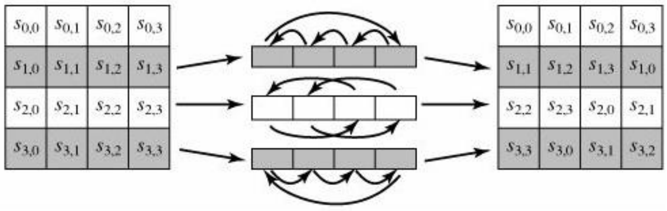
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

Fig -6: Mix Column

2.3.4 Add Round Key

In this process, the 128 bits of state are bitwise XORed through the 128 bits of the round key. The procedure is seen as a column wise process between the word of a state column and one WORD of the round key. This conversion is as basic as would be prudent which benefits in effectiveness yet it additionally influences all of state. To make the relationship between the key and the cipher text more complicated and to satisfy the confusion principle, the Add Round Key operation is performed. This addition step takes the resulting data matrix from the previous step and performs on it a bitwise XOR operation with the sub key of that specific round (addition operation in GF (2n)). We must mention that the round key is 512 bits that is arranged in a square matrix of eight columns where each column has 8 bytes.

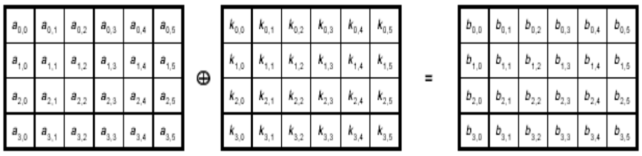
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$ $a_{0,4}$ $a_{0,5}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,4}$ $a_{1,5}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$ $a_{2,4}$ $a_{2,5}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$ $a_{3,4}$ $a_{3,5}$
$\oplus$
$k_{0,0}$ $k_{0,1}$ $k_{0,2}$ $k_{0,3}$ $k_{0,4}$ $k_{0,5}$
$k_{1,0}$ $k_{1,1}$ $k_{1,2}$ $k_{1,3}$ $k_{1,4}$ $k_{1,5}$
$k_{2,0}$ $k_{2,1}$ $k_{2,2}$ $k_{2,3}$ $k_{2,4}$ $k_{2,5}$
$k_{3,0}$ $k_{3,1}$ $k_{3,2}$ $k_{3,3}$ $k_{3,4}$ $k_{3,5}$
=
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$ $b_{0,4}$ $b_{0,5}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$ $b_{1,4}$ $b_{1,5}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$ $b_{2,4}$ $b_{2,5}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$ $b_{3,4}$ $b_{3,5}$

Fig -7: Add Round Key

2.3.5 Key Expansion and Rounds

The 512-bit input key of the new AES-512 algorithm is used to generate ten sub-keys for each of the ten AES rounds. The round ±keys expansion process involves arranging the original 512-bits input key into eight words of eight bytes each. After that, the round keys expansion is performed according to the following equations:
W (I) =W (i-8) XORW (I-1) I is not a multiple of 8
W (I) =W (i-8) XORT (W (I-1)) I is a multiple of 8
Where the T (I) transformation is defined as:
T (I) =Byte Sub (ShiftLeft(W(I)))XORRoundConst
The round constant is defined by the following equation:
RoundConst = 00000010(i-8)/8
I is the round number.

The round structure of the AES-512 algorithm (shown in Figure 8) uses the transformation defined in the previous section. First, byte substitution is performed on 512 bits data, followed by row rotation according to the row number, where left rotations are performed in this step. Then, the columns are multiplied by the new defined matrix column by column in the Mix Column transformation (except in the final round). The last operation will be the bitwise XORing with the round key expanded using the key expansion process. The output at of the final round will be the 512-bit encrypted message.

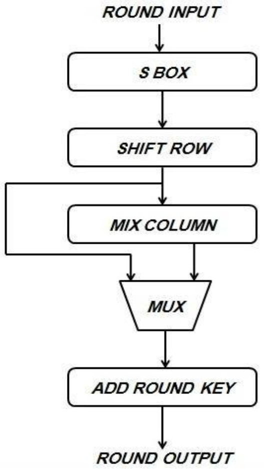

Fig -8: Single Round of AES-512 Algorithm

Table -1: Number of Rounds

VERSION	NUMBER OF ROUND
AES-128	10
AES-192	12
AES-256	14
AES-512	22

3. COMPARISON OF AES 256 AND AES 512

The performance of 256 bit AES algorithm is compared with the performance of AES 512 algorithm. Encryption and decryption of AES 256 is implemented to compare it with AES 512. In terms of security the 256 bit AES algorithm is weaker than the 512 bit AES algorithm. This is because the length of the key used in 512 bit AES increases the number of rounds for both encryption and decryption. But when the number of rounds increases, the encryption and decryption procedures become more complex thereby degrading the speed of the 512 bit AES algorithm. Thus there is a tradeoff between speed and security. The performance is compared in terms of time taken. The time taken for encryption and decryption of AES 256 bit and AES 512 bit are noted to measure their speed. The system time is noted at the start of encryption process and the end time, after encryption completes is also noted. The same process is repeated for decryption also to calculate the time taken for decryption.

Table -2: Comparison of AES-256 and AES-512

Parameters	AES 256 Bits	AES 512 Bits
Key Size	256 Bits	512 Bits
Data Block Size	Not Same as Key	Size Same as Key
Rounds	14	22
Throughput	Less	More
Time to Encrypt 128 Char Message	30-50 Seconds	20-40 Seconds
Security	Less	More
Processor Required	More	Less

4. CONCLUSIONS

We proposed a new variation of AES-512 with 512-bit input block and 512-bit key size compared with 128-bit in the original AES-128 algorithm. When the number of rounds is increased, it improves the complexity of the algorithm making it stronger against the cryptographic attacks. However the length of the key is increased as number of rounds depend on the length of the key used in order to increase the number of rounds involved. Thus the increase in length of the key gives the AES algorithm a strong resistance against the new attacks and has an acceptable speed of data encryption and decryption. A complete software implementation for the new AES-512 was also presented in this paper. After comparing the implementation results, we found that our new design has increased throughput compared with the original AES-128 design. The larger key size makes the algorithm more secure, and the larger input block increases the throughput. The extra increase in area can be accepted and makes the proposed algorithm ideal applications in which high level of security and high throughput are required.

ACKNOWLEDGEMENT

I am profoundly grateful to Prof. Sagar Ghormade, H.O.D., AGPCOE, for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. I am also grateful to Prof. Sagar Pradhan (Co-ordinator) for his support and guidance that have helped me to expand my horizons of thought and expression. I would like to express my deepest appreciation towards Dr.S.M.Chandekar, Principal and Prof. Pragati Patil, Vice-Principal, ECE Department, AGPCOE, whose invaluable guidance supported me in completion of the of this project. At last I must express my sincere heartfelt gratitude to all friends and staff members of ECE Department who helped me directly or indirectly during this course of work.

REFERENCES

[1] U.S. Department of Commerce/NIST, ―Data Encryption Standard,FIPS PUB 46-3, pp. 1-26, October 1999.

[2] NIST, Advanced Encryption Standard, FIPS PUB 197, pp. 1-51, November 2001.

[3] J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, Springer-Verlag, Berlin Heidelberg, 2002.

[4] H. Gilbert and M. Minier, A collision attack on seven rounds of Rijndael, Proceedings of the 3rd AES Candidate Conference, pp.230-241, April 2000.

[5] N. Ferguson, J. Kelsey, S. Lucks, et al. ―Improved cryptanalysis of Rijndael, Lecture Notes in Computer 1-4244-1035-5/07/.2007 IEEE. 221 Science,vol. 1978, pp.213-230, Berlin: Springer-Verlag, 2001.

[6] S. Lucks, ―Attacking seven rounds of Rijndael under 192-bit and 256-bit keys, Proceedings of the 3rd AES Candidate Conference, pp. 215-229, April 2000.

[7] J. Daemen and V. Rijmen, ―The Block Cipher Rijndael, Lecture Notes in Computer Science, vol.1820, pp.277-284, Berlin: Springer-Verlag, 2000.

[8] J. Daemen, and V. Rijmen, ―The Wide Trail Design Strategy, Lecture Notes in Computer Science, vol. 2260, pp.222 - 238, Berlin: Springer-Verlag, 2001.

[9] E. Biham, O. Dunkelman, and N. Keller, ―Related-Key Boomerang and Rectangle Attacks, Lecture Notes in

Computer Science, vol. 3494, pp. 507-525, Berlin: Springer-Verlag, 2005.

[10] G. Jakimoski and Y. Desmedt, —Related-Key Differential Cryptanalysis of 192-bit Key AES Variants, Lecture Notes in Computer Science, vol. 3006, pp. 208-221, Berlin: Springer-Verlag, 2004.

[11] J. H. Cheon, M. J. Kim and K. Kim,et al., —Improved Impossible Differential Cryptanalysis of Rijndael and Crypton, Lecture. Notes in Computer Science, vol. 2288, pp. 39-49, Berlin: Springer-Verlag, 2002.

[12] Abhijith.P.S, Mallika Srivastava, Aparna Mishra, Dept. of Microelectronics, IIITA, India, presented a paper titled "High Performance Hardware Implementation of AES Using Minimal Resources", *International Conference on Intelligent Systems and Signal Processing (ISSP)*, 2013.

[13] Chong Hee Kim published a paper titled "Improved Differential Fault Analysis on AES Key Schedule", *IEEE Transactions on Information Forensics And Security*, VOL. 7, NO. 1, FEBRUARY 2012.

[14] Wang, Man Chen, Zongyue Wang, and Xiaoyun Wang presented a paper in IEEE Transaction titled "Fault Rate Analysis: Breaking Masked AES Hardware Implementations Efficiently", *IEEE Transactions on Circuits And Systems—Ii: Express Briefs*, 2011.

[15] Aditya Rayarapu, AbhinavSaxena, N.Vamshi Krishna,Diksha Mundhra, Department of Computer Science Engineering, Jawharlal Nehru Technological University Vignana Bharathi Institute of Technology,Ghatkesar, Hyderabad, India, presented paper on "Securing Files Using AES Algorithm", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (3), 2013, 433-435 ,ISSN: 0975-9646.

[16] No´emie Floissac and Yann L'Hyver "From AES-128 to AES-192 and AES-256, How to Adapt Differential Fault Analysis Attacks on Key Expansion" *Workshop on Fault Diagnosis and Tolerance in Cryptography*, 2011.

[17] M. Anand Kumar and Dr.S.Karthikeyan , "A New 512 Bit Cipher for Secure Communication ", *I. J. Computer Network and Information Security, 2012*, 11, 55-61 , October 2012 in MECS .

[18] Sagar Pradhan, Dr. Dilip Khairnar "AES-256 Encryption in Communication using LabVIEW" In *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering.* ISSN (Print): 2320-3765, ISSN (Online):2278-8875 Pages (5333-5340), Vol. 4, Issue 6, June 2015.

[19] Sagar Pradhan, Dr. Dilip Khairnar "AES-Encryption in Secured Communication" In Proceeding of 4th International Conference on Innovations in Electronics & Communications Engineering (ICIECE-2015) at Gurunanak Institutions Technical Campus, Hyderabad. Pages (61-64), August 21-22, 2015.