ISSN: 2229-6959 (ONLINE)
DOI: 10.21917/ijsc.2017.0190

MODIFIED AES WITH RANDOM S BOX GENERATION TO OVERCOME THE SIDE CHANNEL ASSAULTS USING CLOUD

M. Navaneetha Krishnan[1] and R. Ravi[2]
[1]*Department of Computer Science and Engineering, Manonmaniam Sundarnar University, India*
E-mail: navanee81@gmail.com
[2]*Department of Information Technology, Francis Xavier Engineering College, India*
E-mail: directorresearch@francisxavier.ac.in

Abstract

Development of any communication system with secure and complex cryptographic algorithms highly depends on concepts of data security which is crucial in the current technological world. The security and complexity of the cryptography algorithms need to get increased by randomization of secret keys. To overcome the issues associated to data security and for improvising it during encryption and decryption process over the encrypting device, a novel Secure Side Channel Assault Prevention (SSCAP) approach has been projected which will eliminate outflow of side channel messages and also provides effective security over the encrypting device. An effective Enriched AES (E-AES) encryption algorithm is proposed to reduce the side channel attack; the modified algorithm in this research shows its improvement in the Generation of Random Multiple S - Box (GRM S-Box) which makes it hard to the attacks to break the text which is in encrypted form. Our novel SSCAP approach also improves the security over the original information; it widely minimizes the leakage of the side channel information. Attackers cannot easily get a clue about the proposed S-Box Generation technique. Our E-AES algorithm will be implemented in cloud environment thereby improving the cloud security. The proposed SSCAP approach is judged against the existing security based algorithms on the scale of encryption and decryption time, time taken for generating the key, and performance. The proposed work proves to outperform over all other methods used in the past.

Keywords:

Encryption, Decryption, AES Algorithm, Side Channel Attack, Random S Box

1. INTRODUCTION

In the rapid growing technological world, millions of information is shared throughout the world in a minute through internet. Our day to day life revolves around the sharing of information, which leads to focusing on the data security and integrity while sharing the information. As a result of the introduction of the cloud resources and cloud services, it has also been a challenging task and it has grown to a threat of providing security over the cloud environment. Cryptosystems are the set of cryptographic algorithms that are needed to provide data security and confidentiality of information. Cryptosystems basically refers to encryption and decryption algorithms along with key generation method. These security algorithms facilitate in avoiding attacks and to provide security over the confidential information. Attack against cryptographic systems can be categorized into active and passive attack. The passive attack will try to have the information accessed illegally without the knowledge of the communicators, since these attacks will not affect or disrupt the information transfer. For instance, actions like eavesdropping and intercepting the communication. Active attacks are the unauthorized access of the data such as modifying the information, denying the access permission of the legitimate users, unauthorized deletion or alteration of the information. These attacks directly affecting the information exchange unlike the passive attack.

Some of the well-known attacks are brute force attack, dictionary attack, birthday attack, timing attack, side channel attack, etc. Side channel assaults depend on the physical information of the cryptographic device. Side channel information doesn’t include a plaintext or a cipher text. In recent years, encryption devices have been secured so that it have additional inputs and outputs other than plain text or cipher text, which makes it hard to break the data, instead these encryption devices produce timing information, power consumption traces, and various radiations which are easily measured, and are labeled to be side channel data, thus paving way to side channel attack along with cryptographic attacks to get back the encrypted key. To avoid this side channel leakage, Advanced Encryption Standard is used for data encryption and information.

Side channel attack has be modified and improved to affect the AES encryption and those attacks breaks the security levels of this encryption. Some of the additional attacks are algebraic side channel, analytic side channel attack, cache based side channel attack, etc. These side channel assaults are effective in deriving the side channel data when the AES encryption and decryption taking place, since they extract additional data from the encryption/decryption. Thus increase the threat of uncovering the original information from the encryption device. These side channel attacks are usually critical and have to be given serious concern to avoid these attacks.

2. REVIEW OF RELATED WORKS

Michael et al. [1] investigated the usage of Preventing CPU-cache based side-channels using cloud and compared with traditional side-channel attacks. They demonstrated that emerging methods are necessary to overcome these types of attacks in a cloud system, and they also suggested the requirements for such solutions.

Fangfei et al. [2] presented a useful operation of the PRIME+PROBE side-channel attack against the last level cache. They measured the capacity of the covert channel the attack creates and demonstrate a cross-core, cross-VM attack on multiple versions of GnuPG. This technique achieves a high attack resolution without relying on weaknesses in the OS or virtual machine monitor or on sharing memory between attacker and victim.

Yang Li et al. [3] have found new Fault-Based Side-Channel Attack using a concept named Fault Sensitivity. In the FSA attack, fault sensitivity is emphasised in which sensitive information leakage are tested using fault injections.

Charles et al. [4] have demonstrated the strength of the tools, to automatically recognize the new attacks on reduced round AES through very low data complexity, and thus to find the enhanced attacks on the AES.

Chari et al. [5] has demonstrated that the execution of AES, is not adaptable to methods such as SPA and DPA, and can simply be shattered using template attacks with a single sample. Achieving these attacks in feasible circumstances makes the user challenged to handle the noises emerging in each sample.

Itai Dinur and Adi Shamir [6] developed the concept of leakage attacks on block ciphers which are iterated, and the attacker can discover physical searching, power measurement, or any other form of side channel. The unique cube attack needs particularly clean data, however the data given by side channel attacks is noisy.

Zhao et al. [7] have investigated that one of the powerful cryptanalysis method is Algebraic side-channel attack, which is quite special compared to conventional side-channel attacks. To enhance ASCA, they have proposed a Multiple Deductions-based ASCA to cope up with the multiple deductions produced by erroneous measurements or intrusions.

Shaheb et al. [8] have discovered that upon removing the quadratic equations, iterated chosen plaintexts, and cube iteration to progress the SCCA on block ciphers can be identified without any difficulty.

3. BASIC CONCEPT OF AES

AES is a type of block cipher, symmetric in nature which is far more advantageous than conventional DES approach for a wide range of applications. The main highlight about AES is that it is not a Feistel structure. The principle of a classic Feistel structure is that, half of the data block modifies the other half of the data block and after that swapping of the halves occurs. But the entire data block gets processed in AES as a single matrix along with the aid of substitutions and permutations. The input key is expanded as an array of four 32 bit words, *w*[*i*]. Four 32 bit words are processed in four different stages, one of permutation and three of substitution: Substitution makes use of S box to perform a byte-by-byte substitution of block Shift Rows. A simple permutation takes into account the concept of Mix Columns approach, which makes use of arithmetic operations over GF (28). Add Round Key is a simple bitwise XOR of the current block. The structure is quite simple. The Fig.1 depicts the structure of full encryption round. Only the add-round key stage makes use of the key. The add round key stage is in effect a form of vernam cipher and by itself would not be formidable. AES is efficient, highly secure and easily reversible. However the drawback of AES being, the encryption and decryption algorithms are not identical. The structure of which is shown in Fig.2.

3.1 SHIFT ROWS (SHIFT ROWS OPERATION)

The Fig.3 shows the Shift Rows operation, a similar shifting model is adopted for size blocks of 128 bits and 192 bits. The left nth row is shifted by n-1 bytes. As well as, output Shift Rows operation of each column include bytes from the corresponding input column.

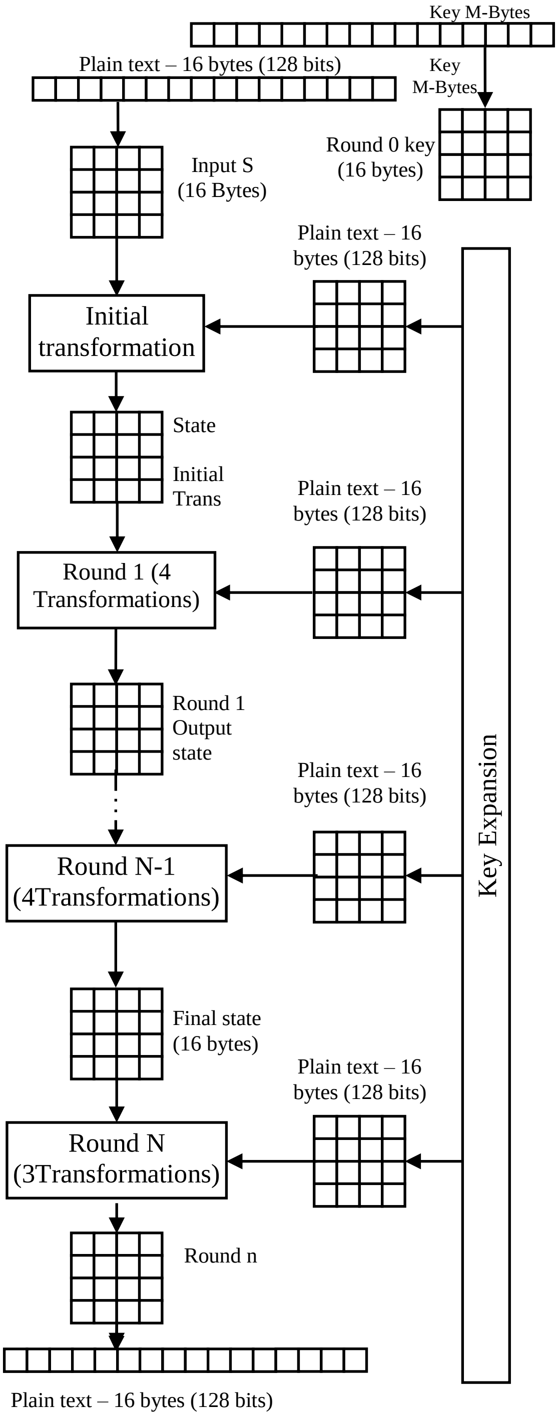


Fig.1. AES Encryption Process

3.2 MIX COLUMNS (MIX COLUMNS OPERATION)

Invertible linear transformation principle is used for four bytes of the column in mix columns operation (Fig.4). Matrix operation includes multiplication and addition of the entries. Entries like eight bit bytes treated as polynomial coefficients. Addition is solely XOR and Multiplication is modulo irreducible polynomial.

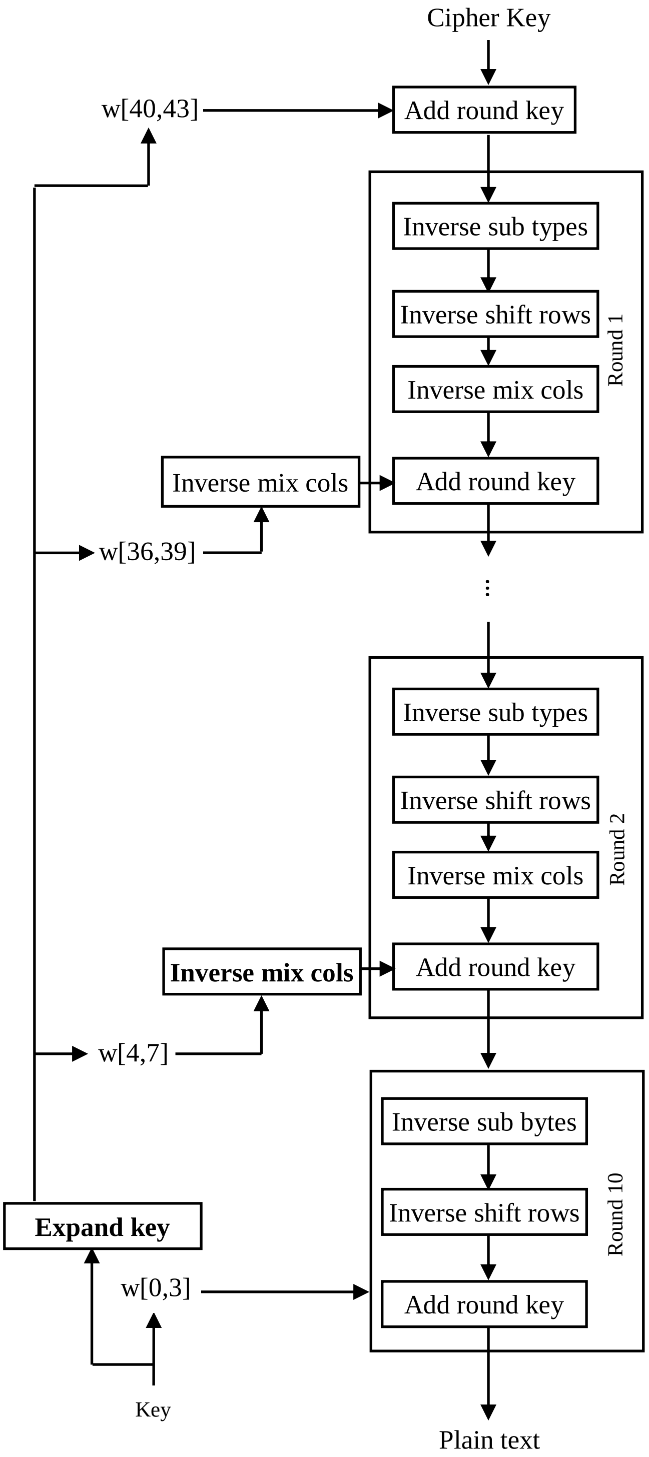
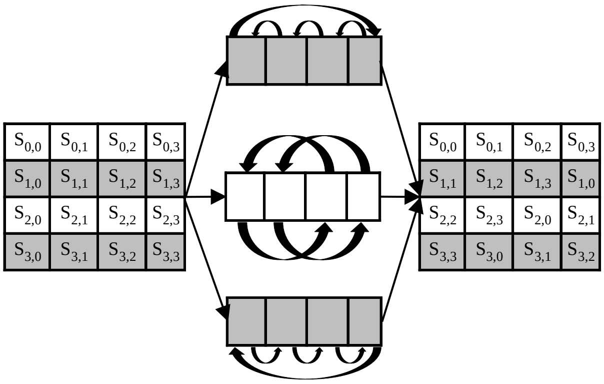


Fig.2. AES Decryption Process

4. SIDE CHANNEL ANALYSIS

Performance of AES can be improved by using a number of table look ups. Since these tables do not completely go well with the cache size, cache hits or misses are common during encryption, which leads to different come into view up times, and hence variable encryption times transform according to the input text and the encryption key.

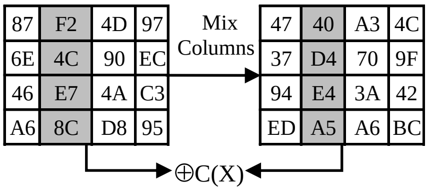
Side-channel attacks depend on the relation between the leaked information and the confidential data and the methods to overcome the same can be classified into two categories: Eliminating the release of any unnecessary data comes into the first category. The second class is elimination of the correlation between the leaked information and the confidential data, which can be made possible by making the leaked information isolated to the secret data.



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

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

Fig.3. Shift Rows Operation



87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

Fig.4. Mix Columns Operation

4.1 TIMING ATTACK

Time taken to perform operations taking into consideration, the concerned key forms the basis of timing attack. Cryptographic algorithms take different time to process each input, the reason being to optimize the performance for the operations. Encryption key and input data forms the main crux of Performance characteristics. (e.g. plaintext or cipher text). Timing characteristics which are not planned before the execution of the programs lead to information leakage from the cryptographic algorithms. Arithmetic models can be developed from timing measurements which gives data regarding the estimated key bit with some degree of accuracy.

4.2 POWER MONITORING ATTACK

Power analysis may be a form of cryptographic attacks with in which the wrongdoer gets a view about power consumption of a hardware device (such as a wise card, tamper-resistant “black box”, or integrated circuit). The attack facilitates the extraction of cryptographic keys and secret data from the device.

5. PROPOSED WORK

To overcome the issues related to side channel attacks and to improve security during encryption and decryption over the encrypting device, a novel Secure Side Channel Attack Prevention (SSCAP) approach has been proposed. An effective

Enriched AES (E-AES) encryption algorithm is proposed to prevent the side channel attack, which shows some improvement in the Generation of Random Multiple S-Box (GRM S-Box) which makes it hard to the attacks to break the encrypted text from. A novel SSCAP approach also improves the security over the original information; it widely minimizes the leakage of the side channel information. Attackers will find it difficult to guess the proposed S-Box Generation technique. Our E-AES algorithm will be implemented in the cloud environment to improve the cloud security.

The Fig.5 shows the proposed enriched AES algorithm of random S-box generation approach, which is evaluated with the existing security based algorithms in the context of execution time, encryption time, key generation time, decryption time and performance. The proposed work proves to outrun all the other existing methods.

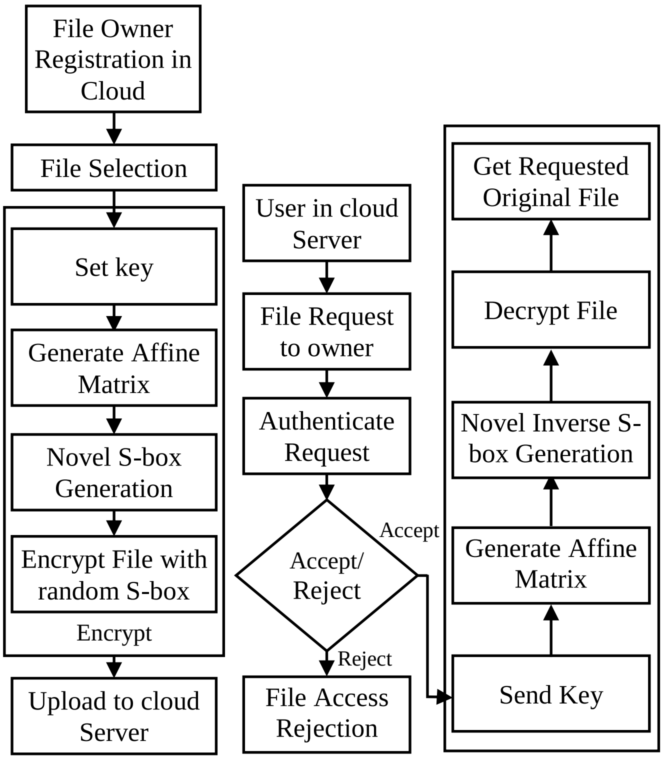


Fig.5. Proposed Enrich AES Algorithm of random S box Generation

5.1 SYSTEM ARCHITECTURE

Once a domain is obtained from the particular cloud, a file can be registered by a user. But it cannot be assured that the file would be viewed only by particular users he sets so, in order to assure that the owner's file is safe and protected from all users, he can secure it by setting a key under which the file he created would be always under his supervision. It is done using Enriched AES (Advanced Encryption Standard). The procedure below describes how a key is set to secure a file:

Generate an Affine matrix using which our file is secured in form of codes. A Random S-box is created in which our generated file is stored. By using this we can encrypt our file with random S-box. Now our file will be safe and secured under our key constructed. This secured file is uploaded in cloud. Any user desire to use the file would send a request to the owner. Once the owner receives a request its owner's wish either to accept or reject the request. If the authentication is done positively then the particular key for the entire file he requested would be issued using which user can access the file. Once the key is given correctly to the cloud then the cloud retrieves the requested file. Then the file is decrypted from encrypted from using Enriched AES using which we can get the original file requested from the owner.

5.2 PROPOSED AES ENCRYPTION

The Fig.6 shows the proposed AES Encryption, owner of the file who needs to register his file in the cloud would prefer to keep his file secured; it is done using AES Encryption. The file to be uploaded must be selected for which a key is generated in an affine matrix and the concerned file is generated as our codes to be encrypted. Our file is then generated and encrypted using Random S-box. The modified S-box tends to provide higher security to the encrypted file. The encrypted file with a secured key is uploaded in the cloud again.

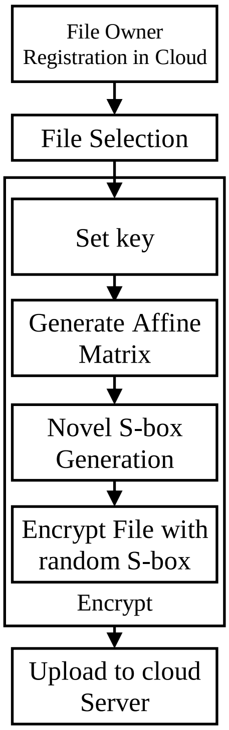


Fig.6. Proposed AES Encryption

5.3 PROPOSED AES DECRYPTION

The Fig.7 shows the AES decryption receiver of the file also needs to be registered in the cloud, who gives request for the owner to get accessed for the files. Once the owner receives a request, the owner sends the key to the receiver so that the file can be viewed by the latter. Affine matrix is generated for the owner's key by the receiver. From the affine matrix, inverse S-box is generated which is different from a normal S-box. The modified inverse S-box tends to provide higher security to the file from the attackers. The file requested by the receiver to the owner is then decrypted using the inverse S-box. The requested file can thus be made use by the receiver which is received from the file owner without getting accessed by the attackers or any third party.

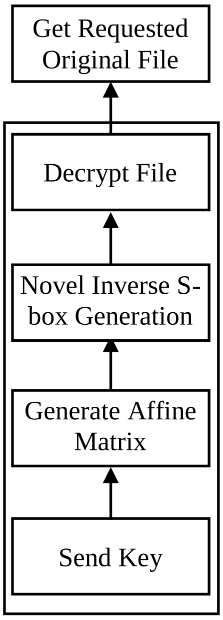


Fig.7. Proposed AES Encryption

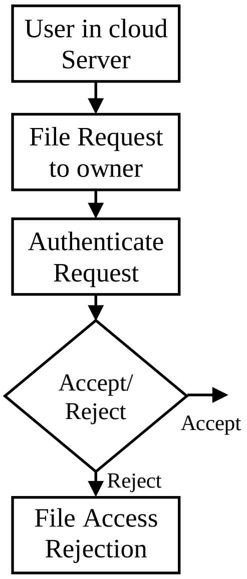


Fig.8. Cloud Architecture

5.4 CLOUD ARCHITECTURE

The Fig.8 shows could architecture where the cloud has many users and owners assessing various types of file at a time where there are many chances for our files to be accessed by unknown users. The receiver who tends to make use of the owner's file sends a request to owner to get the key for accessing the file. The owner who decides to give the file to the receiver authenticates the request of the receiver by issuing a key to the user. If the owner doesn't wish the user to access the file, owner rejects the request given by the receiver, else accepts the requests and sends user a key for the accessing the file by the. Using our file is always secured and kept safe.

6. MODIFIED AES ALGORITHM

At first the key is generated and the total value is calculated as the total bytes allocated for the keys. Now using this value the mean is calculated as follows,

M = Value/Key size from which threshold

$Th = M/100$ is calculated.

Using this threshold value the Random S-box is generated by creating 16×16 8 bit matrixes.

For every round the entire matrix gets updated.

Algorithm 1:

Input:

$K \leftarrow$ key for data encryption

Input ← file or data to encrypt (plaintext)

Output:

Cipher Text or Encrypted Data

Procedure:

```
Let "k" is key to encrypt file (key size of AES Algorithm)
Val = 0
S[k.size]=k.bytes()
For i = 0 to N, where N is the Size of key
        Val = val + Ascii(s[i]);
end
Compute mean of Val;
M = Val/Key Size;
Th = M/100;        //Fixing Threshold
```

Algorithm 2:

Generation of Random Multiple S-Box:

```
Computation of 16×16 S-Boxes
Initialize Th;
Initialize k = 0;
To create 16×16 8 bit Matrix;
For i = 0 to N where N = 16
        For j = 0 to M where M = 8
                Set R = Th×2;
                If R > 1
                        B(k) = 1;
                        k = k + 1;
                        R = R - 1;
                Else
                B(k) = 0;
                End if else
        End for
End for
End
```

After one round of GRM S-box,

Assume result is,

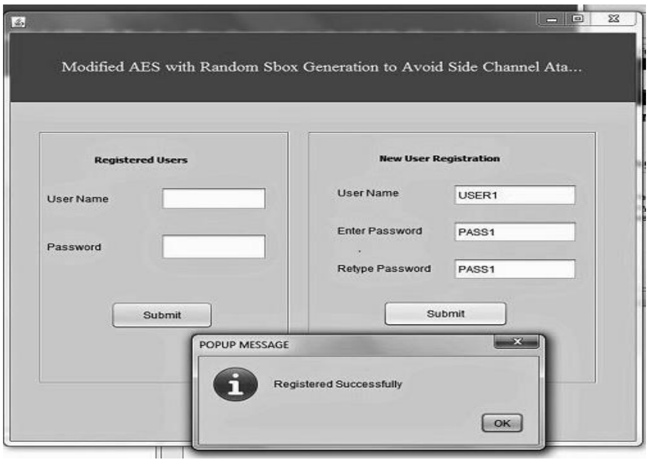
11010101 10101010 11100101 10101001 10010101 1100011 ……………….

D5 AA E5 A9 95 63……. In S-Box

By this algorithm 16×16 S-box Matrix will be generated. Further encryption and decryption is similar to AES algorithm. The Inverse S-Box is individual for each S-Box.

7. RESULTS AND PERFORMANCE

The authorized user is denoted by registering your details in the form. They are only allowed to send a data from source to destination. The registration should have only the valid details. This is shown in Fig.9.



Modified AES with Random Sbox Generation to Avoid Side Channel Ata...

Registered Users

User Name

Password

Submit

New User Registration

User Name: USER1

Enter Password: PASS1

Retype Password: PASS1

Submit

POPUP MESSAGE

Registered Successfully

OK

Fig.9. User Register structure

7.1 KEY SIZE VS TIME CONSUMPTION

Comparison of different key size with time taken to generate key in milliseconds, for 128 bits its taking a minimum of 225 milliseconds compared to 192 and 256 bits as shown Fig.10.

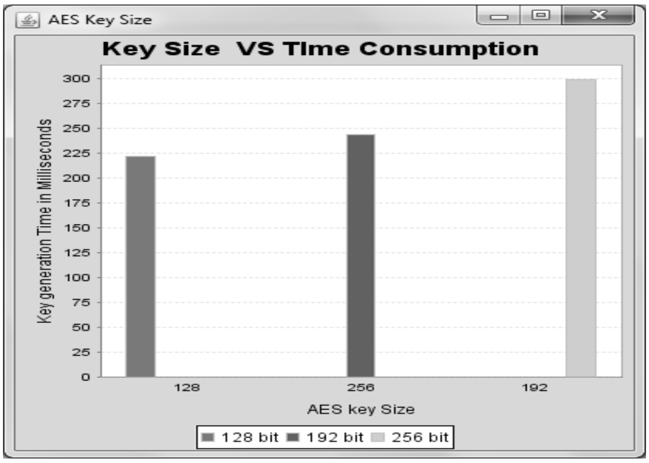


Fig.10. Key size vs. time consumption

7.2 ENCRYPTION TIME VS INPUT SIZE

Encryption time of M-AES is much less compared to other algorithms for encrypting files with similar size, as shown Fig.11.

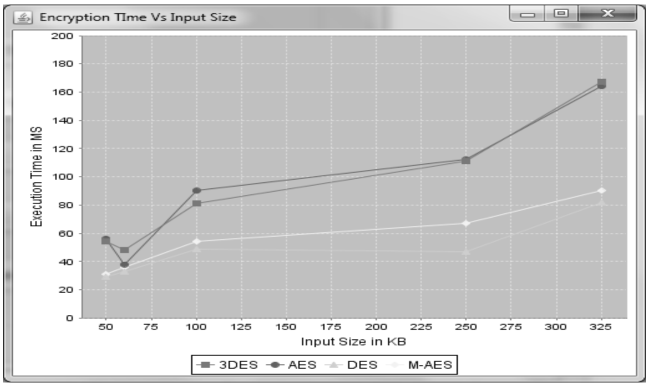


Fig.11. Execution time vs. Input size

7.3 DECRYPTION TIME VS PACKET SIZE

Decryption time of M-AES is much less compared to other algorithms for decrypting files with similar size, as shown Fig.12.

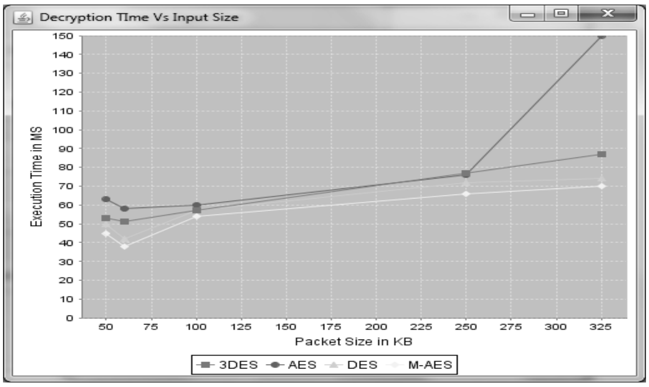


Fig.12. Execution time vs. packet size

7.4 COMPARISONS OF ENCRYPTION TIME

Comparing the encryption time of different algorithms with different bit sizes shows that the performance of M-AES is higher, as shown Fig.13.

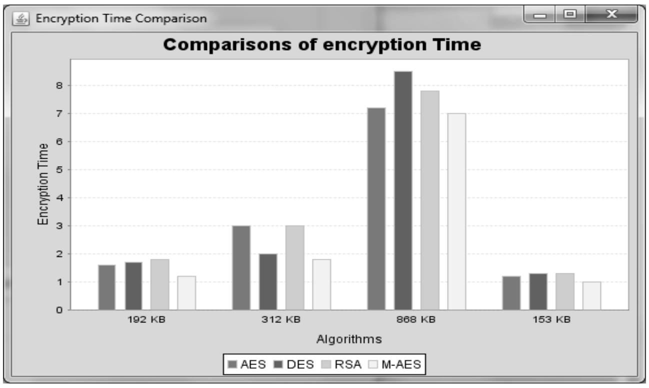


Fig.13. Encryption time comparisons

7.5 COMPARISONS OF DECRYPTION TIME

Comparing the decryption time of different algorithms with different bit sizes shows that the performance of M-AES is higher, as shown Fig.14.

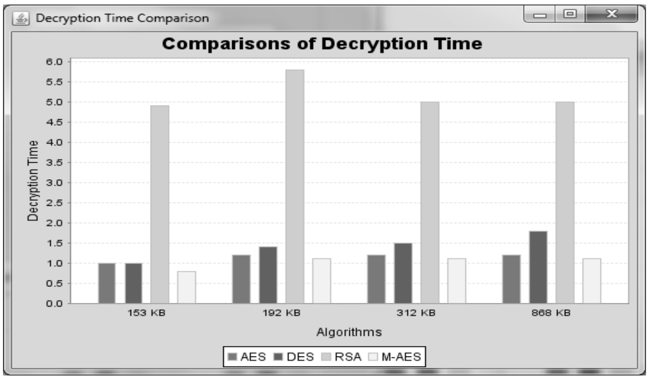


Fig.14. Decryption time comparisons

7.6 THROUGHPUT VS. ENCRYPTION AND DECRYPTION ALGORITHMS

Comparison of Throughput value during encryption for various encryption algorithms with M-AES shows encryption process can be completed in rapid and competent approach, as shown Fig.15.

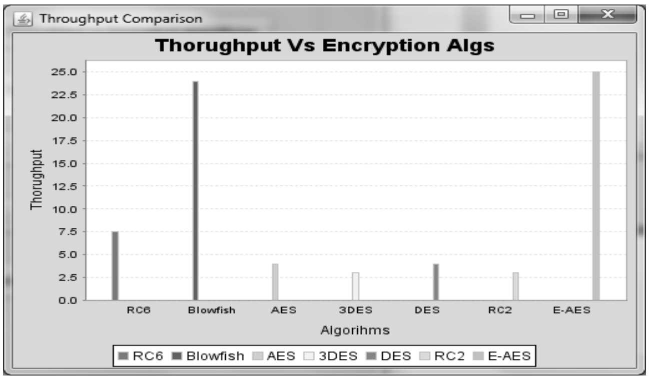


Fig.15. Throughput vs. Encryption algorithms

Comparison of throughput while decryption when comparing with different decryption algorithms-AES performance is higher. This is shown in Fig.16.

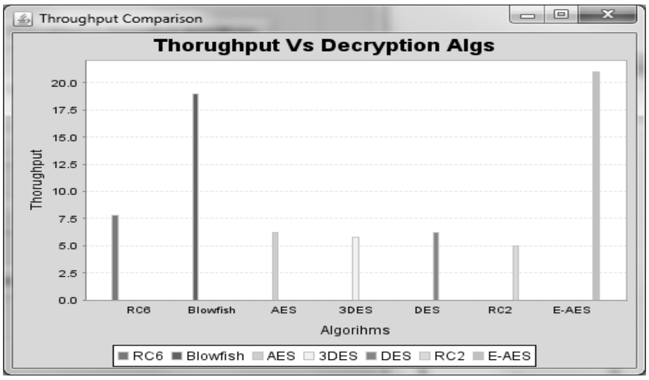


Fig.16. Throughput vs. Decryption algorithms

7.7 KEY VS CORRELATION COEFFICIENT

The Fig.17 shows the comparison of different encryption algorithms of different key size like, 64, 128, 192, 256 to the correlation coefficient.

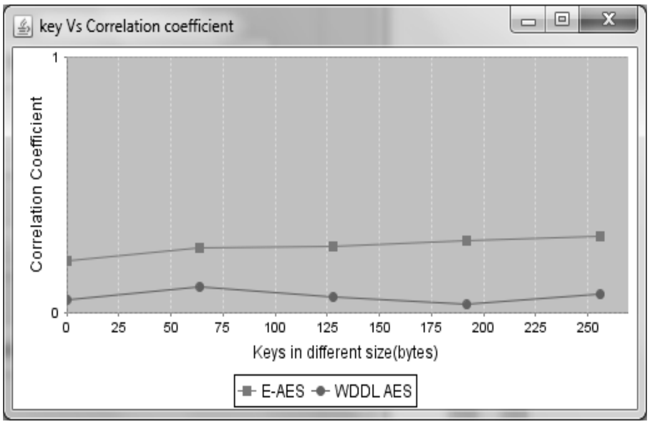


Fig.17. Key size vs. correlation coefficient

7.8 NUMBER OF PLAIN TEXTS VS. CORRELATION COEFFICIENT

The Fig.18 shows the comparison of different encryption algorithms of different plain text sizes the correlation coefficient.

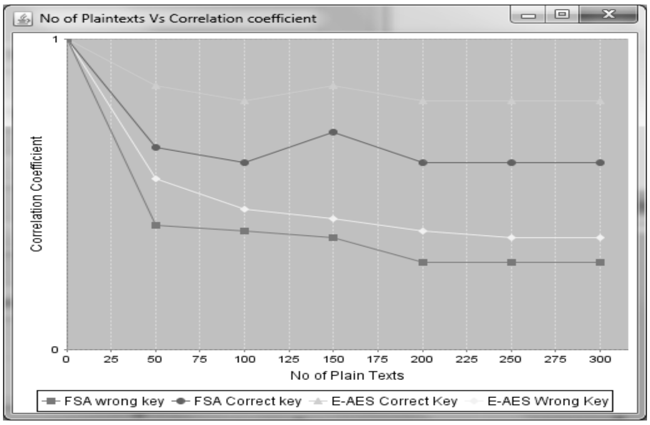


Fig.18. Plain texts vs. correlation coefficient

8. CONCLUSIONS

The proposed method constructs the success of algorithm and the prevention methods are also functional to defend adjacent to side channel attacks which depends user data and the network channel. To defeat the issues associated to data security and to improve the same while encryption and decryption over the encrypting device, a novel Secure Side Channel Assault Prevention (SSCAP) approach has been projected which will avoid side channel information leak and also provides effective security over the encrypting device. An effective Enriched AES (E-AES) encryption algorithm is projected to overcome the side channel attack; our proposed algorithm shows its improvement in the Generation of Random Multiple S - Box (GRM S-Box) which makes it hard to the attacks to break the encrypted text type. Our novel SSCAP approach also improves the security over the original information; it widely minimizes the leakage of the side channel information. Attackers will find it rigid to analyze our proposed S-Box Generation technique. Through of various prevention techniques against the SCA, the data is transferred securely and the attackers are unable to retrieve data. This increases the level of security in AES by preventing it from side channel assault and the uses AES efficiently and also provide secured system for the data flow of user's data.

REFERENCES

[1] Michael, M., Godfrey and Mohammad Zulkernine, "Preventing Cache-Based Side-Channel Attacks in a Cloud Environment", *IEEE Transactions on Cloud Computing*, Vol. 2, No. 4, pp. 395-408, 2014.

[2] Fangfei Liu, Yuval Yarom, Qian Ge, Gernot Heiser and Ruby B. Lee, "Last-Level Cache Side-Channel Attacks are Practical", *Proceedings of IEEE Symposium on Security and Privacy*, pp. 605-622, 2015.

[3] Yang Li, Kazuo Ohta and Kazuo Sakiyama, "New Fault-Based Side-Channel Attack using Fault Sensitivity", *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 1, pp. 88-97, 2012

[4] Charles R Harrell and Ken Tumay, "Promodel Tutorial", *Proceedings of 24th Conference on Winter Simulation*, pp. 405-410, 1992.

[5] Suresh Chari, Josyula R. Rao and Pankaj Rohatgi, "Template Attacks", *Proceedings of* 4th *International Workshop Cryptographic Hardware and Embedded Systems*, Vol. 2523, pp. 13-28, 2002.

[6] Itai Dinur, Orr Dunkelman and Adi Shamir, "Improved attacks on full GOST", *Proceedings of 19th International Workshop Fast Software Encryptio*n, Vol. 7549, pp. 9-28, 2012.

[7] Xinjie Zhao, Fan Zhang, Shize Guo, Tao Wang, Zhijie Shi, Huiying Liu and Keke Ji, "MDASCA : An Enhanced Algebraic Side Channel Attack for Error Tolerance and New Leakage model Exploitation", *Proceedings of 3rd International Workshop Constructive Side Channel Analysis and Secure Design*, Vol. 7275, pp. 231-248, 2012.

[8] Asif Shahab, Faisal Shafait and Andreas Dengel, "ICDAR 2011 Robust Reading Competition Challenge2: Reading Text in Scene Images", *Proceedings of International Conference on Document Analysis and Recognition*, pp. 1491-1496, 2011.