Modified Advanced Encryption Standard for Image Encryption and Decryption

Yogita Verma*, Neerja Dharmale**

Abstract

In today's world most of the communication is completed using electronic media. Cryptography is widely used to ensure security in communication, data storage and transmission. The protection of this multimedia message can be done by encryption techniques. For encryption many different techniques are used to protect confidential image data from unauthorized access. The cryptography concept is used in many applications like password, visa card number, military communications, banks, satellite channels and some other communication system. Advanced Encryption Standard (AES) has many advantages over data encryption Standard. If symmetric key encryption algorithm is used for complex multimedia data (mostly images) so it is not suitable for real time application. We proposed a modified advanced encryption standard (MAES) to reflect a high level security and better image encryption. In this proposed project modification is done by adjusting the Shift Row step in advanced encryption standard algorithm. The shift row phase of modified advanced encryption standard (MAES) is different from advanced encryption standard (AES) shift row phase. The experimental results of this work gives better performance than AES algorithm.

Keywords: Encryption, Decryption, Data Encryption Standard (DES), Advanced Encryption Standard (AES), Blowfish, Modified Advanced Encryption Standard (MAES)

Introduction

Cryptography is changes the data representation from its original form into another totally different form in order to make it hidden and secured. Cryptography involves two step processes; the first process is the encryption where the original data is converted into secured form using certain steps, Decryption is second process, where the encrypted data is restored to the original form by applying the inverse process to the steps applied in the encryption process. With continuing development of technology, multimedia data is used for communication, in application such as video conferencing, broadcasting etc. here various encryption techniques are available such as AES, RSA, TDES [1, 2], and most of this is mainly used for text or data encryption. It is difficult to use for multimedia data. And if AES algorithm is used for image encryption the entropy of image is 7.9926 but ideally it should be 8 or more near about this value. This paper proposed a new encryption technique as a modified advanced encryption standard algorithm. This modification is mainly done in shift row phase. First check the value of the first row and first column, i.e. (state [0][0]) is even or odd if even then each byte of the second row is shifted by one to the right or if odd then each byte of the second row is shifted by one to the left ,after that examine the value of first row and second column, i.e. (state [0][1]) is even or odd if even then each byte of the third row is shifted by two to the right or if odd then each byte of the third row is shifted by two to the left and at last examine the value of first row and third column, i.e. (state [0][2]) is even or odd if even then each byte of the fourth row is shifted by three to the right or if odd then each byte of the fourth row is shifted by three to the left.

Advanced Encryption Standard Algorithm

The Advanced Encryption Standard is a block cipher. The cipher supports three key sizes: 128, 192, 256 bits. The AES algorithm with 128-bit, 192-bit and 256-bit

* M. Tech Scholar Digital Electronics, Rungta College of Engineering and Technology, Bhilai, Chhattisgarh, India.
Email: yogita.verma305@gmail.com
** Assistant Professor (ET&T), Rungta College of Engineering and Technology, Bhilai, India.
Email: n.dharmale@rediffmail.com

key are referred to as AES-128, AES-192 and AES-256 respectively. Here the key length is denoted by N_k, which represents the number of 32-bit words in the key. The input block, the output block and the intermediate cipher result all have the same length of 128 bits. The block size is denoted by N_b, which reflects the number of 32-bit word in the block. The number of AES round is denoted by N_r, and is finding by key size: 10 rounds for AES128, 12 rounds for AES192, and 14 rounds for AES256. [4]

Table 1 The Block-Key-Round Parameters

Variants	*Block size (N_b words)*	*Key length (N_k words)*	*Number of rounds (N_r)*
AES-128	4	128	10
AES-192	4	192	12
AES-256	4	256	14

The encryption and decryption process of AES algorithm is shown in Figure 1, for encryption the plaintext and a key is taken as input and gives output as cipher-text. The plaintext is represented as a byte matrix with 4 rows and 4 columns. The intermediate cipher text outcome is called as state. After the completion of initial round key addition, the state is transformed by implementing a round operation 10, 12, or 14 numbers of times for 128-bit, 192-bit or 256-bit key sizes, respectively. Function of each round, except the last round; contain four steps of transformations which are Substitute Bytes (SB), Shift Rows (SR), Mix Columns (MC) and Add Round Key (ARK). The final round is slightly different from the first Nr −1 rounds as it does not include the Mix Columns operation.

For decryption, the ciphertext and a key as two input parameters and give output as plaintext. The four steps of encryption process are Substitute Bytes, Shift Rows, Mix Columns and Add Round Key, can be inverted in reverse order to provide the decryption of the ciphertext. The inverse process of Substitute Bytes, Shift Rows and Mix Columns are named as Inverse Shift Rows, Inverse Substitute Bytes and Inverse Mix Columns, respectively. Note that the inverse function of Add Round Key is itself

A. Substitute bytes and inverse substitute bytes: Substitute is basically a lookup table substitution utilizing a 16×16 double dimension of byte values known as s-box. This dimension comprises of every conceivable combos of 8 bit sequence ($2^8 = 16 \times 16 = 256$). We divide the input byte into two four bit patterns, to find the substitute byte for a given input byte. The s-box isn‘t only an arbitrary stage of these qualities and there is an overall-characterized technique for making the s-box matrix. The S-box is constructed by combining two transformations which include an inverse function and an invertible affine transformation.

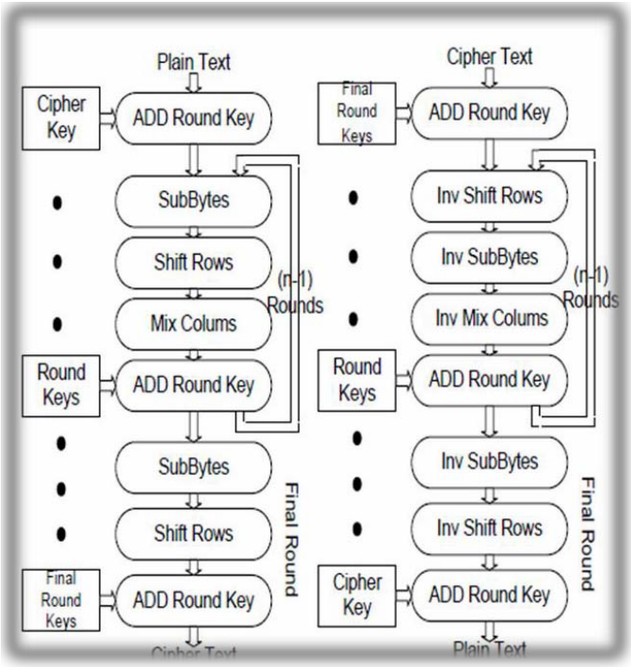


Figure 1 Advanced Encryption Standard

In Inverse substitute bytes Inverse S-BOX is used for the decryption processes. It is retrieving the original byte that was substituted using the S-BOX during the encryption process. The S-BOX is generated by two steps, first find the multiplicative inverse for the numbers 00H-FFH in the GF (2^8), then after that applying the affine transformation on them. On the other hand, the generation of the Inverse S-BOX starts by applying the inverse affine transformation followed by finding the multiplicative inverse.

B. Shift rows and Inverse Shift rows: In the Shift Rows phase, the first row remains same. The bytes in the second, third and forth rows are cyclically shifted by one, two and three bytes to the left, respectively as shown in figure 3.

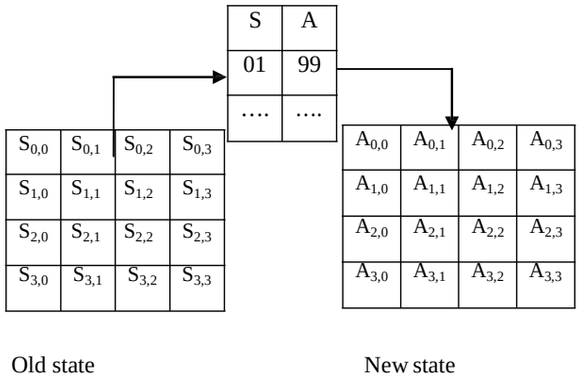


Figure 2 Substitution Operations as a Lookup Table

Inverse shift row is an inverse process of the shift rows; the first row of the state array remains same. And the bytes of the second, third and forth rows are cyclically shifted by one, two and three bytes position to the right, respectively.

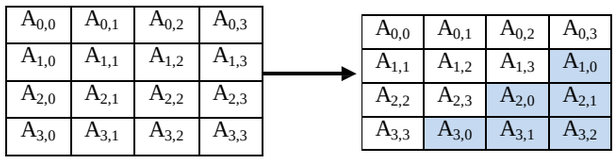


Figure 3 Shift Rows Operation

C. Mix columns and inverse mix columns: Mix column transformation is a word substitution yet it makes utilization of math of GF (2^8). Every segment is worked on separately. Every byte of a segment is charted into another esteem that is a capacity of each of the 4 bytes in the section. This step replaces every byte of a column i.e. one word by a function of all the bytes in the same column.

More compactly, the column operations can be given as

$$\begin{matrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{matrix} \times \begin{matrix}s0,0 & s0,1 & s0,2 & s0,3\\ s1,0 & s1,1 & s1,2 & s1,3\\ s2,0 & s2,1 & s2,2 & s2,3\\ s3,0 & s3,1 & s3,2 & s3,3\end{matrix} = \begin{matrix}S'0,0 & S'0,1 & S'0,2 & S'0,3\\ S'1,0 & S'1,1 & S'1,2 & S'1,3\\ S'2,0 & S'2,1 & S'2,2 & S'2,3\\ S'3,0 & S'3,1 & S'3,2 & S'3,3\end{matrix}$$

The Inverse Mix Columns transformation is the inverse of the Mix Columns operation. The inverse column operation given below,

$$\begin{matrix}0E & 0B & 0D & 09\\ 09 & 0E & 0B & 0D\\ 0D & 09 & 0E & 0B\\ 0B & 0D & 09 & 0E\end{matrix} \times \begin{matrix}s0,0 & s0,1 & s0,2 & s0,3\\ s1,0 & s1,1 & s1,2 & s1,3\\ s2,0 & s2,1 & s2,2 & s2,3\\ s3,0 & s3,1 & s3,2 & s3,3\end{matrix} = \begin{matrix}S'0,0 & S'0,1 & S'0,2 & S'0,3\\ S'1,0 & S'1,1 & S'1,2 & S'1,3\\ S'2,0 & S'2,1 & S'2,2 & S'2,3\\ S'3,0 & S'3,1 & S'3,2 & S'3,3\end{matrix}$$

D. Add round key: The AddRound Key transformation adds the state bytes to round key by using a simple bitwise XOR operation. The key used in each round for this transformation is derived from the secret key by employing the key schedule which is described below. Each round key has the same size as the state.

$$\begin{matrix}a1,0 & a1,1 & a1,2 & a1,3\\ a2\,0 & a2\,1 & a2\,2 & a2\,3\\ a3;0 & a3;1 & a3;2 & a3;3\end{matrix} \otimes \begin{matrix}k1,0 & k1,1 & k1,2 & k1,3\\ k2\,0 & k\;1 & k2\,2 & k2\,3\\ k3;0 & k3;1 & k3;2 & k3;3\end{matrix} = \begin{matrix}b0,0 & b0,1 & b0,2 & b0,3\\ b1,0 & b1,1 & b1,2 & b1,3\\ b2,0 & b2,1 & b2,2 & b2,3\\ b3,0 & b3,1 & b3,2 & b3,3\end{matrix}$$

E. Expansion algorithm: The AES Key Expansion algorithm is used to obtain the 128-bit round key for each round from the original 128-bit encryption key. In the same manner as the 128 bits input block is arranged in the form of a state array, arranged the 16 bytes of the encryption key in the form of a 4×4 array of bytes, as given below

$$\begin{matrix}k1 & k5 & k9 & k13\\ k2 & k6 & k10 & k14\\ & & \Downarrow & \\ k3 & k7 & k11 & k15\end{matrix}$$

[w0 w1 w2 w3]

the generating the four words of the round key for given round from the corresponding four words of the round key for the previous round. Let‘s say that we have the four words of the round key for the ith round:

wi ,wi+1 ,wi+2 ,wi+3

For these to serve as the round key for the ith round, i must be a multiple of 4. These will obviously serve as the round key for the (i/4)th round. For example, for round one; w4, w5, w6, w7 words are used as key, for second round; w8, w9, w10, w11 words are used as a key and so on..

Now we need to determine the words

wi+4 wi+5 wi+6 wi+7

From the words

wi wi+1 wi+2 wi+3.

From Figure 4, we write

$$wi+5 = wi+4 \otimes wi+1$$
$$wi+6 = wi+5 \otimes wi+2$$
$$wi+7 = wi+6 \otimes wi+3$$

So now we only need to find out wi+4. This is the starting word of each 4-word grouping in the key expansion. The wi+4 =wi⊗g(wi+3).That is, the first word of the new 4-word grouping is to be obtained by XOR'ing the first word of the last grouping with what is returned by applying a function g() to the last word of the previous 4 word grouping.

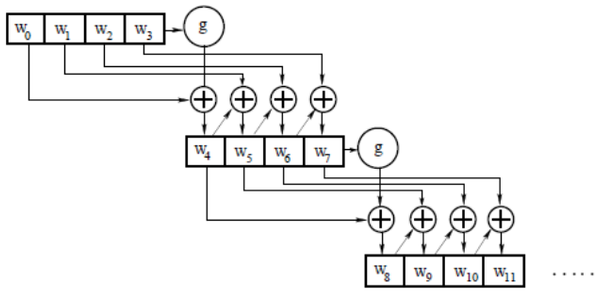


Figure 4 Key Expansion takes Place on a Four-Word to Four-Word

Modified Advanced Encryption Standard

We want to modify the AES to be more efficient and secure method by adjusting the Shift Row section. Shift Row Phase: rather than the initial Shift row, we tend to modify it as:

- Check (state [0][0]) i.e. the value in the first row and first column, is even or odd if even then each byte of the second row is shifted by one to the right or if odd then each byte of the second row is shifted by one to the left.
- after that examine the value of first row and second column, (state [0][1]) is even or odd if even then each byte of the third row is shifted by two to the right or if odd then each byte of the third row is shifted by two to the left .
- At last examine the value of first row and third column, (state [0][2]) is even or odd if even then each byte of the forth row is shifted by three to the right or if odd then each byte of the forth row is shifted by three to the left.

For decryption, the corresponding step shifts the rows in exactly the opposite fashion. In inverse Shift Rows step shifting after monitor the (state [0][0]) i.e. the value in the first row and first column, is even or odd if even then each byte of the second row is shifted by one to the left or if odd then each byte of the second row is shifted by one to the right. after that examine the value of first row and second column, (state [0][1]) is even or odd if even then each byte of the third row is shifted by two to the left or if odd then each byte of the third row is shifted by two to the right. At last examine the value of first row and third column, (state [0][2]) is even or odd if even then each byte of the forth row is shifted by three to the left or if odd then each byte of the forth row is shifted by three to the right.

If $A_{0,0}$, $A_{0,1}$ and $A_{0,2}$ are even:

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

→

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

If $A_{0,0}$, $A_{0,1}$ and $A_{0,2}$ are odd

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

→

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

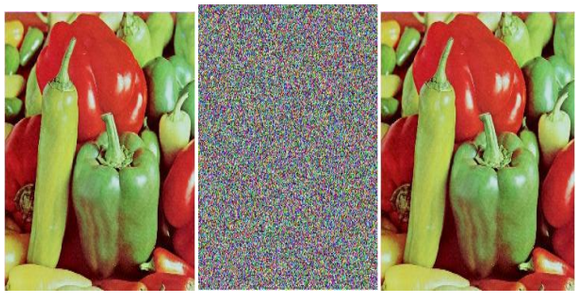
Figure 5 Shift Rows Operation

Results

The experiment is performed on three different sizes of image namely _lena, pepper and jellyfish of size 256x256, 512x512 and 1024x 1024 respectively. The original image, encrypted images and decrypted image is shown in Figures. The encrypted images are shown in Figures. 6b, 7b and 8b.The cipher image regions are totally invisible. The decrypted images are shown in Figs. 6c, 7c and 8c.the proposed MAES is applied successfully in both encryption and decryption process.

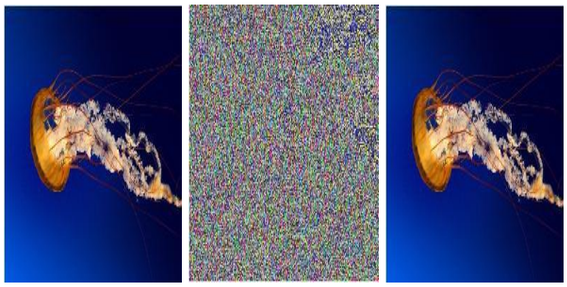
(a) (b) (c)

Figure 6 Original Image, Encrypted Image and Decrypted Image of Lena



(a) (b) (c)

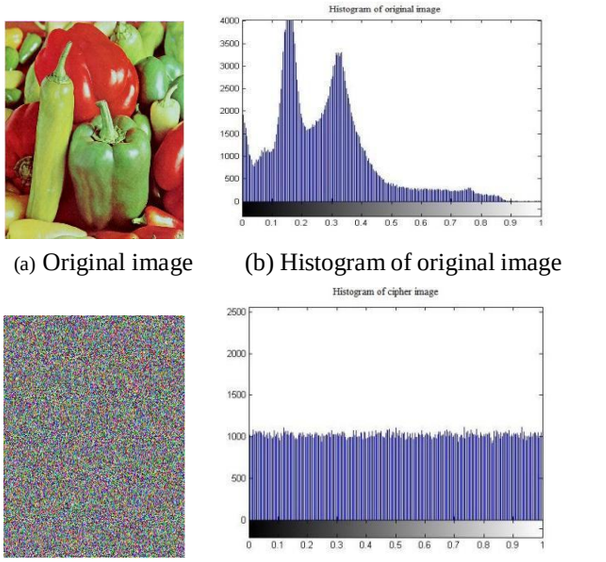
Figure 7 Original Image, Encrypted Image and Decrypted Image of Pepper



(a) (b) (c)

Figure 8 Original Image, Encrypted Image and Decrypted Image of Jellyfish

Histograms analysis: Histogram analysis of an image gives the information about the how number of pixels in an image at each color intensity. It blocks the leakage of message to an opposer, and it is a graphical representation of the pixels distribution of a digital image. It plots the number of pixels for each color intensity; the analyzing the histogram for an image will be able to judge the entire pixels distribution at a glance. The horizontal axis of the plot represents the intensity variations, and the vertical axis represents the number of pixels in that particular color intensity. Figure 9 shows the histogram of original image and histogram of cipher image, histogram of plain image contains large spikes, and histogram of cipher image is uniformly distributed. Hence both histogram analyses are significantly different from each other. So here no clue to employ any statistical attack on the proposed image encryption algorithm.



(a) Original image (b) Histogram of original image

(c) Encrypted image (d) Histogram of Encrypted image

Figure 9 Histograms of the Plain Image and Ciphered Image

Correlation: correlation gives the relationship between two variables, in other words it is a measure of similarity between two variables, it is a good parameter to judge the quality of encryption, any image is said to be good, if encryption algorithm hides property of an original image (i.e. plaintext) and cipherimage is random and uncorrelated. If original image and cipherimage are totally different then their correlation coefficient should be very low. If correlation coefficient is equal to one, then two images are similar. Encryption algorithm completely fails if correlation coefficient is equal to one, because the encrypted image is same as the original image. When correlation coefficient is - 1 then encrypted image is negative of original image. Then, calculate their correlation coefficient using the following two formulas:

$$\mathrm{cov}(x, y) = E(x - E(x))(y - E(y)) \quad (1)$$

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{D\,\square\;\; D(\square)} \qquad (2)$$

Table 2 Correlation Coefficient of Original and Encrypted Image

Correlation of image	*Plainimage*	*Cipherimage*
256x256	.72352	-.53572
512x512	.9961	-.050462
1024x1024	1	-.61512

Information entropy: entropy defined as the average amount of information contained in message. Modern information theory concerned with cryptography, error correction and data compression and related topics. To calculate entropy H (m) of m source the formula is given below:

$$H(m) = {}_{i=0}\, P(mi) \log_{2\ P(mi)} \qquad (3)$$

Where P (m_i) is probability of symbol m i and unit of entropy is bits per message.

Table 3 Entropy of Images Using AES and MAES

Image size (in pixels)	*Image size on disk*	*Entropy value using AES*	*Entropy value using MAES*
256x256	22kb	7.9926	7..9985
512x512	109kb	7.9978	7.9998
1024x1024	145kb	7.9990	7.9995

Encryption time: It is basically time taken by algorithm to convert a simple message to unreadable format. In this work, encryption time for modified advanced encryption standard algorithm applied on image has been calculated. In this algorithm, the set of operation i.e. substitution, shift left, add round key and mix column are applied to plain text in each round.

Table 4 Encryption Time of Images Using AES and MAES

Image size (in pixels)	*Image size on disk*	*Encryption time in sec with AES*	*Encryption time in sec with MAES*
256x256	22kb	13.4323	23.0845
512x512	109kb	93.5542	186.0872
1024x1024	145kb	291.0191	397.0191

Conclusion

In this work following pixels size of image have been taken 256x256, 512x512 and 1024x1024.According to experimental analysis it has been found that using modified advanced encryption standard algorithm for the image of lena, pepper and jellyfish the Encryption time is 23.0845, 186.0872 and 397.0191 sec respectively, and decryption time is 2`1.1329, 1703877 and 384.1147 sec respectively. Entropy of those images is 7.9985, 7.9998 and 7.9995. Hence modified advanced encryption standard algorithm is take more time to encrypt and decrypt the image but it gives better entropy value and low correlation of cipher image as compared to advanced encryption standard algorithm. So this proposed method is more secure than AES algorithm.

References

Agrawal, M., & Mishra, P. (2012). A comparative survey on symmetric key encryption techniques. *International Journal on Computer Science and Engineering (IJCSE),* 4(5), 877- 882.

Asok, S. B., Karthigaikumar, P., Sandhya, R., Naveen J. K., SivaMangai, N. M. (2013). *A Secure Cryptographic Scheme for Audio Signals,* 748-752, IEEE.

Davis, R. (1978). The data encryption standard in perspective, Proceeding of Communication Society magazine, IEEE, 16(6), 5-6.

Hesham, S., Salem, A. B. E. G., & Hofmann, K. (2014). *High throughput architecture for the advanced encryption standard algorithm,* 978-1-4799-4558- 0/14/$31.00 ©2014 IEEE

Kawle, P., Hiwase, A., Bagde, G., Tekam, E., & Kalbande, R. (2014). Modified advanced encryption standard. *International Journal of Soft Computing and Engineering (IJSCE),* 4(1).

Kumar, M. A., & Karthikeyan, S. (2012). Investigating the Efficiency of Blowfish and Rejindael (AES) Algorithms. *I.J. Computer Network and Information Security,* 2, 22-28.

Mandal, P. C. (2012). Superiority of blowfish algorithm. *International Journal of Advanced Research in Computers Science and Software Engineering*, 2(9).

Manoj. B, Harihar, M. N. (2012). Image encryption and decryption using AES. *International Journal of Engineering and Advanced Technology (IJEAT)* ISSN: 2249-8958, 1(5).

Mukhopadhyay, D. (2009). *An improved fault based attack of the advanced encryption standard,* 421-434.

Saraf, K. K. K., Jagtap, V. P., & Mishra, A. K. (2014). Text and image encryption decryption using advanced encryption standard. *International Journal of Emerging Trends & Technology in Computer Science,* May – June 2014

Schneier, B. (1996). *Applied cryptography* (2nd Ed.). John Wiley & Sons1996.

Singh, G., Singla, A. K., & Sandha, K. S. (2011). Performance evaluation of symmetric cryptography algorithms. *International Journal of Electronics and Communication Technology,* 2(3).

Stallings, W. (2004). *Network security essentials* (Applications and Standards), Pearson Education..

Thambiraja, E., Ramesh, G., & Umarani, R. (2012). A survey on various most common encryption techniques. *International Journal of Advanced Research in Computer Science and Software Engineering,* 2(7).