An image encryption and decryption using AES algorithm

Priya Deshmukh

Abstract— These In today's world data security is the major problem which is to be face. In order to secure data during communication, data storage and transmission we use Advance encryption standard(AES). AES is a symmetric block cipher intended to replace DES for commercial applications.it uses 128-bit block size and a key size of 128, 192, or 256 bits. The AES algorithmis use to secure data from unauthorized user. The available AES algorithm is used for text data as well as for image data. In this paper an image is given as input to AES encryption algorithm which gives encrypted output. This encrypted output is given as input to AES decryption algorithm and original image is regained as output. The AES algorithm for image encryption and decryption which synthesizes and simulated with the help of MATLAB software

Index Terms— AES, cipher, DES, image encryption, image decryption, MATLAB

1 Introduction

Now a day's the uses of devices such as computer, mobile and many more other device for communication as well as for data storage and transmission has increases. As a result there is increase in no of user's also there is increase in no of unauthorized user's which are trying to access a data by unfair means. This arises the problem of data security. To solve this problem a data is stored or transmitted in the encrypted format. This encrypted data is unreadable to the unauthorized user. Cryptography is a science of information security which secures the data while the data is being transmitted and stored.

Every encryption and decryption process has two aspects: the algorithm and the key use for the encryption and decryption. However, it is the key used for encryption and decryption that makes the process of cryptography secure. There are two types of cryptographic mechanisms: symmetric key cryptography in which the same key is use for encryption and decryption. In case of asymmetric key cryptography two different keys are used for encryption and decryption. Symmetric key algorithm is much faster and easier to implement and required less processing power as compare to asymmetric key algorithm.

The advance encryption standard (AES) specifies a federal information processing standards publication (FIPS) approved cryptographic algorithm that can be used to protect electronic data. It was publish by National Institute of Standard and Technology (NIST) in 2001 developed by Joan Daemen and Vincent Rijmen, an algorithm called Rijdael[2].

Because of the drawbacks in the 3DES such as the algorithm is relatively slow in software and it uses 64-bit block size which is less for more security large block size is required. Due to this reason AES is intended to replace 3DES. AES has advantages over 3DES such as high computational efficiency, 128-bit block size,and cryptanalysisresistance is strong against differential truncated differential, linear, interpolation and square attacks [1][9].

The application of image processing is mainly found in military communication, Forensic, Robotics, intelligent system, etc. In this paper, we implemented the AES algorithm on image with the help of MATLAB software.

2 AES Algorithm Specification

AES algorithm is of three types i.e. AES-128, AES-192 and AES-256. This classification is done on the bases of the key used in the algorithm for encryption and decryption process. The numbers represent the size of key in bits. This key size determines the security level as the size of key increases the level of security increases. The AES algorithm uses a round function that is composed of four different byte-oriented transformations. For encryption purpose four rounds consist of:

- Substitute byte
- Shift row
- Mix columns
- Add round key

While the decryption process is the reverse process of the encryption which consists of:

- Inverse shift row
- Inverse substitute byte
- Add round key
- Inverse mix columns

There is a number of round present of key and block in the algorithm. The number of rounds depends on the length of key use for Encryption and Decryption.

TABLE I
Key-Block Round combinations

	Key length(in word/byte/bits)	Block size(in word/byte/bits)	Numbers of rounds
AES-128	4/16/128	4/16/128	10

Priya. Deshmukh is M.E student ofProf. Ram Meghe Institute of Technology and Research,Badnera

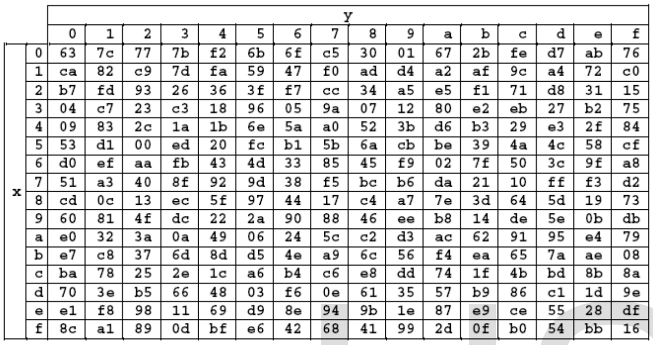
AES-192	6/24/192	4/16/128	12
AES-256	8/32/256	4/16/128	14

AES algorithm uses a round function for both its Cipher and Inverse Cipher. This function is composed of four different byte-oriented transformations.

1. Encryption process

1.1Substitute byte transformation

The Substitute bytes transformation is a non-linear byte substitution that operates independently on each byte of the State using a substitution table S-box. The operation of substitute byte is shown in figure 1.



		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
x	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

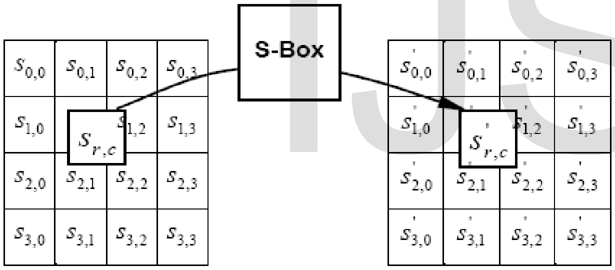
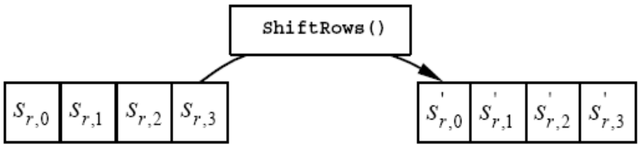

S-Box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{r,c}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{r,c}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Table.2. S-box

Fig.1.Operation of substitute byte

1.2 Shift rows transformation

In the Shift Rows transformation, the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes. The first row, r = 0, is not shifted.This has the effect of moving bytes to "lower" positions in the row while the "lowest" bytes wrap around into the "top" of the row.


ShiftRows()
$s_{r,0}$ $s_{r,1}$ $s_{r,2}$ $s_{r,3}$
$s'_{r,0}$ $s'_{r,1}$ $s'_{r,2}$ $s'_{r,3}$

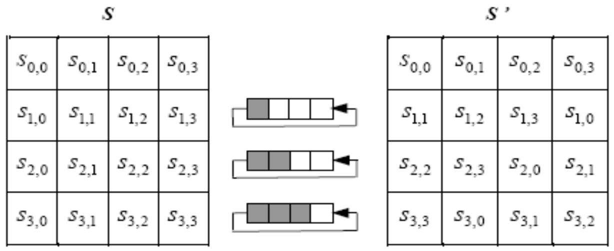

S
S '
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$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

Figure.2.Cyclic shift row operation

1.3 Mix columns transformation

The Mix Columns transformation operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF(2^8) and multiplied modulo x 4 + 1 with a fixed polynomial a(x), given by

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

The resultant columns are shown in the figure below. This is the operation of mix columns.

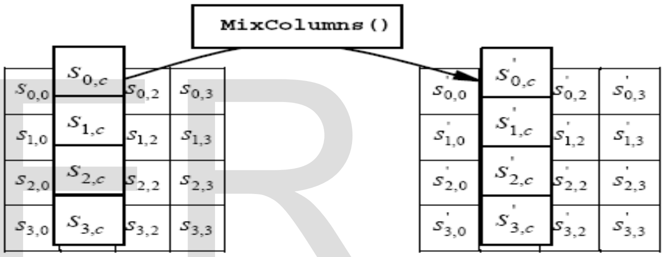

MixColumns()
$s_{0,0}$ $s_{0,c}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,c}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,c}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,c}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,c}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,c}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,c}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,c}$ $s'_{3,2}$ $s'_{3,3}$

Figure.3.Mix columns operation

1.4 Add round key transformation

In the Add Round Key transformation, a Round Key is added to the State by a simple bitwise XOR operation. The Round Key is derived from the Cipher key by means of key schedule process. The State andRound Key are of the same size and to obtain the next State an XOR operation is done per element:

b (i, j) = a (i, j) ⊕ k (i, j)

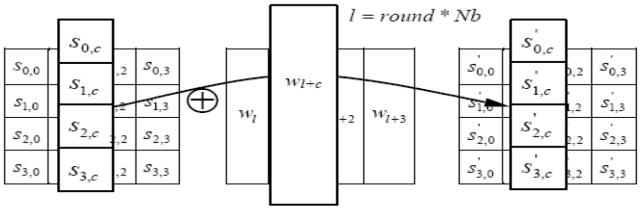

$l = round * Nb$
$s_{0,c}$ $s_{1,c}$ $s_{2,c}$ $s_{3,c}$
$s_{0,0}$ $s_{1,0}$ $s_{2,0}$ $s_{3,0}$
$s_{0,3}$ $s_{1,3}$ $s_{2,3}$ $s_{3,3}$
⊕
w_l
w_{l+c}
w_{l+3}
$s'_{0,c}$ $s'_{1,c}$ $s'_{2,c}$ $s'_{3,c}$
$s'_{0,0}$ $s'_{1,0}$ $s'_{2,0}$ $s'_{3,0}$
$s'_{0,3}$ $s'_{1,3}$ $s'_{2,3}$ $s'_{3,3}$

Figure 4. Add round key operation

2. Decryption process

2.1. Inverse shift row transformation

Inverse Shift Rows is the inverse of the Shift Rows transformation. The bytes in the last three rows of the State are cyclically shifted over different numbers of bytes. The first row, r = 0, is not shifted. The bottom three rows are cyclically shifted by Nb-shift(r, Nb) bytes, where the shift value shift(r,Nb) depends on the row number

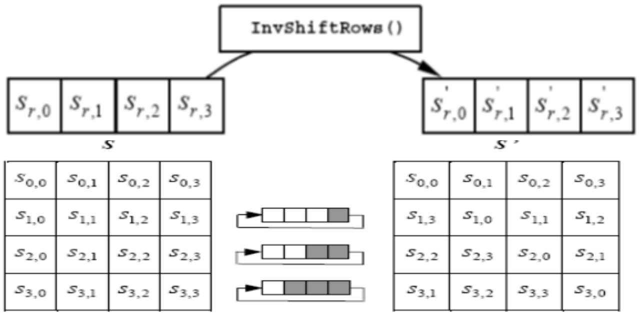
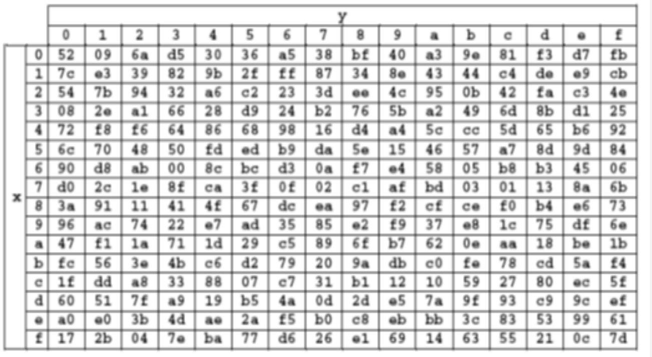

InvShiftRows()
$s_{r,0}$ $s_{r,1}$ $s_{r,2}$ $s_{r,3}$
s
$s'_{r,0}$ $s'_{r,1}$ $s'_{r,2}$ $s'_{r,3}$
s'
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,3}$ $s_{1,0}$ $s_{1,1}$ $s_{1,2}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,1}$ $s_{3,2}$ $s_{3,3}$ $s_{3,0}$

Figure.5. Inverse Shift row operation

2.2 Inverse substitute byte transformation

Inverse Substitute Bytes is the inverse of the byte substitution transformation, in which the inverse S-box is applied to each byte of the State. It is reverse process of Substitute byte transform. This is obtained by applying the inverse of the affine transformation followed by taking the multiplicative inverse in GF (2^8). There is an inverse s-box table for substitute the value.



		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
x	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Table.3. Inverse S-box

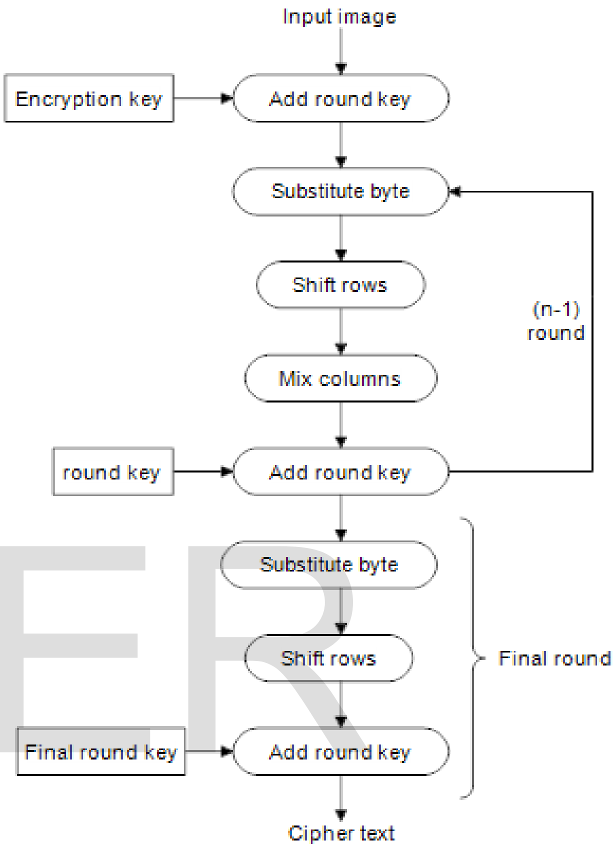
2.3 Inverse mix columns transformation

Inverse Mix Columns is the inverse of the Mix Columns transformation. Inverse Mix Columns operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF(2^8) and multiplied modulo x^ 4 + 1 with a fixed polynomial a^{-1}(x), given by

$$a^{-1}(x) = \{0b\}x\text{^}3 + \{0d\}x\text{^}\ 2 + \{09\}x + \{0e\}$$

3 Implementation

A. Encryption algorithm

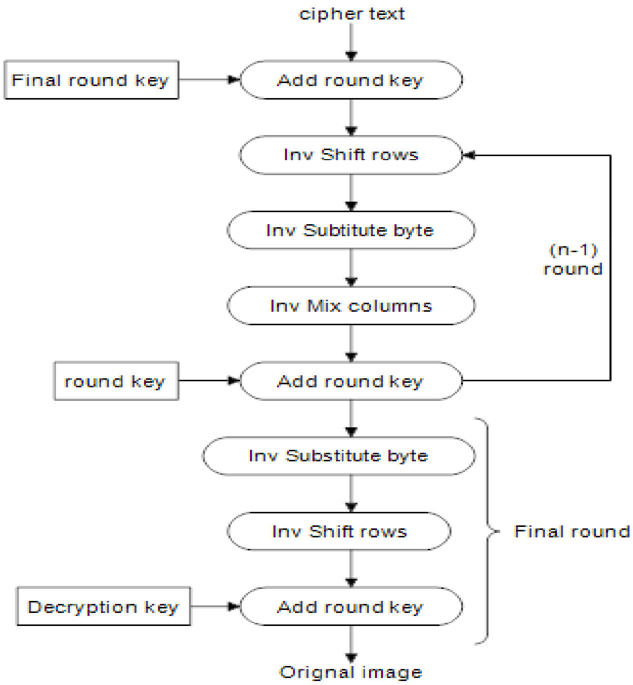


The implementation of the AES-128 encryption and decryption algorithm with the help of MATLAB software is done. In which the input is an image and the key in hexadecimal format and the output is the same as that of input image. For encryption process first, dividing image and making it 4*4 byte state i.e. matrix format. Calculate the number of rounds based on the key Size and expand the key using our key schedule. And there are (n-1) rounds performed which are substitute byte, shift rows, mix columns and add round key. The final round "n" does not consist of mix column in the iteration. Figure 6 shows the flow of algorithm.

Fig.6. Flowchart of AES Encryption algorithm

B. Decryption algorithm

The AES decryption process is the revers process that of the encryption process. The above figure shows flow of the AES decryption algorithm. Which consist of cipher text as the input, the key is same for decryption process which for



encryption. In case of decryption the inverse substitute byte, inverse shift rows and the inverse mix columns are to be implemented. While the add round key remains the same

Figure.7. Flowchart of AES decryption algorithm

4 Result

The original input image given to the algorithmis of JPG And of 8.32 Kb size. The unreadable image is the encrypted image and by applying the decryption algorithm the original image is obtained in JPG format. In this paper, For Encryption and the decryption the same key is used. The key is in hexadecimal form and length of key is 128 bits.
Key used= 0123456789abcdef.
Input= Image in JPG format.

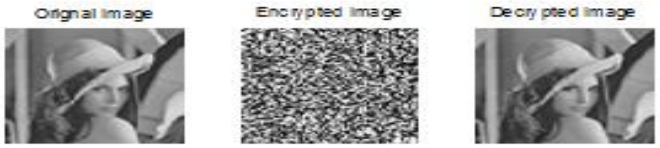


Figure.8. Final output window

5 Conclusion

In this paper, Image Encryption and Decryption using AES algorithm is implemented to secure the image data from an unauthorized access. A Successful implementation of symmetric key AES algorithm is one of the best encryption and decryption standard available in market. With the help of MATLAB coding implementation of an AES algorithm is synthesized and simulated for Image Encryption and Decryption. The original images can also be completely reconstructed without any distortion. It has shown that the algorithms have extremely large security key space and can withstand most common attacks such as the brute force attack, cipher attacks and plaintext attacks.

References

[1] William Stallings, “Advance Encryption Standard,” in *Cryptography and Network Security*, 4th Ed., India:PEARSON,*pp. 134–165*.

[2] AtulKahate, “Computer-based symmetric key cryptographic algorithm”, in *Cryptography and Network Security*, 3th Ed. New Delhi:McGraw-Hill, *pp*. 130-141.

[3] Manoj .B,Manjula N Harihar (2012, June). “Image Encryption and Decryption using AES”, International Journal of Engineering and Advance Technology (IJEAT) volume-1, issue-5, pp.290-294.

[4] KundankumarRameshwarSaraf, Vishal PrakashJagtap, Amit Kumar Mishra, (2014, May-June).”Text and Image Encryption Decryption Using Advance Encryption Standard”, International Journal of Emerging Trends and Technology in computer science(IJETTCS) volume-3, issue-3, pp.118-126.

[5] VedkiranSaini, ParvinderBangar, Harjeet Singh Chauhan, (2014, April).”Study and Literature Survey of Advanced Encryption Algorithm for Wireless Application”, International Journal of Emerging Science and Engineering (IJESE) volume-2, issue-6, pp.33-37.

[6] Sourabh Singh, Anurag Jain, (2013, May). “An Enhanced Text to Image Encryption Technique using RGB Substitution and AES”, International Journal of Engineering Trends and Technology (IJETT) volume-4,issue-5,pp.2108-2112.

[7] R.Gopinath, M.Sowjanya, (2012, October).”Image Encryption for Color Images Using Bit Plane and Edge Map Cryptography Algorithm”, International Journal of Engineering Research and Technology (IJERT) volume-1, issue-8, pp.1-4.

[8] Kundankumar R. Saraf,Sunita P. Ugale, “Implementation of text encryption and decryption in Advance Encryption Standard”, ASM’S International E-journal of ongoing Research in Management and IT.

[9] Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani, (2010, March), “New Comparative Study Between DES, 3DES and AES within Nine Factors”, Journal of computing,volume2-,issue-3,pp.152-157