Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Efficient Implementation of AES

Ritu Pahal
Dept. ECE, Student
SGI Samalkha, Haryana, India

Vikas kumar
Dept. ECE, Asst. Professor
SGI Samalkha, Haryana, India

Abstract: ***With the fast progression of digital data exchange in electronic way, information security is becoming much more important in data storage and transmission. Cryptography has come up as a solution which plays a vital role in information security system against various attacks. This security mechanism uses some algorithms to scramble data into unreadable text which can be only being decoded or decrypted by party those possesses the associated key. Two types of cryptographic techniques are being used: symmetric and asymmetric. In this paper we have used symmetric cryptographic technique AES (Advance encryption standard) having 200 bit block as well as key size. And the same conventional 128 bit conventional AES algorithm is implemented for 200 bit using 5*5 Matrix. After the implementation, the proposed work is compared with 128 bit, 192 bits & 256 bits AES techniques on two points. These points are encryption and decryption time and throughput at both encryption and decryption sides.***

Keywords: Plain text, cipher text, Block cipher, S-Box, Inverse S-Box.

I. Introduction

The rapidly growing number of wireless communication users has led to increasing demand for security measures and devices to protect user data transmitted over wireless channels. Two types of cryptographic systems have been developed for that purpose: symmetric (secret key) and asymmetric (public key) cryptosystems. Symmetric cryptography, such as in the Data Encryption Standard (DES), 3DES, and Advanced Encryption Standard (AES), uses an identical key for the sender and receiver, both to encrypt the message text and decrypt the cipher text. Asymmetric cryptography, such as in the Rivest-Shamir-Adleman (RSA) uses different keys for encryption and decryption, eliminating the key exchange problem. Symmetric cryptography is more suitable for the encryption of a large amount of data. The AES algorithm defined by the National Institute of Standards and Technology (NIST) of the United States has been widely accepted to replace DES as the new symmetric encryption algorithm [2]. The AES algorithm is a symmetric block cipher that processes data blocks of 128 bits using a cipher key of length 128, 192, or 256 bits. Each data block consists of a 4 × 4 array of bytes called the *state*, on which the basic operations of the AES algorithm are performed [2].The proposed algorithm differs from conventional AES as it has 200 bits block size and key size both. Number of rounds is constant and equal to ten in this algorithm. The key expansion and substitution box generation are done in the same way as in conventional AES block cipher. AES has 10 rounds for 128-bit keys,12 rounds for 192-bit keys, and 14 rounds for 256-bit keys [3].

II. Proposed Algorithm

Encryption Algorithm

At the start of encryption, 200 bit input is copied to the State array of 5*5 matrix. The data bytes are filled first in the column then in the rows. Then after the initial round key addition, ten rounds of encryption are performed. The first nine rounds are same, with small difference in the final round. As illustrated in fig.1 each of the first nine rounds consists of 4 transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey. But in final round Mixcolumns transformation is not used.

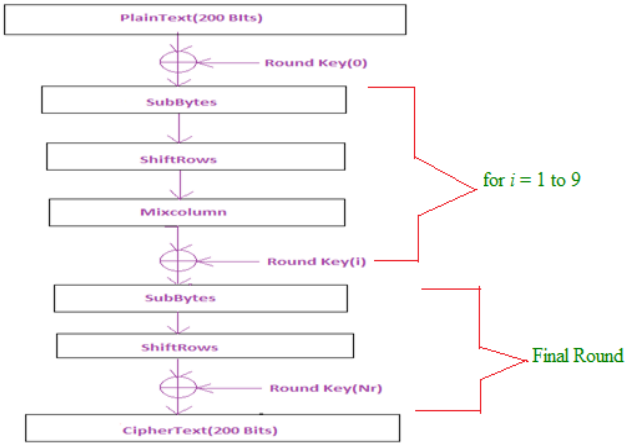

Fig 1. Encryption Structure of the AES algorithm [1]

2.1 SubBytes Transformation— In this transformation, each of the byte in the state matrix is replaced with another byte as per the S-box (Substitution Box). The S-box is generated by firstly calculating the respective reciprocal of that byte in GF (2^8) and then affine transform is applied. The S-box used for this transformation is given in table 1 below:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Table 1: S-Box [3]

2.2 ShiftRows Transformation— In this transformation, the bytes in the first row of the State do not change. The second, third, fourth and fifth rows shift cyclically to the left by one byte, two bytes, three bytes and four bytes respectively, as illustrated in Fig. 2 [1].

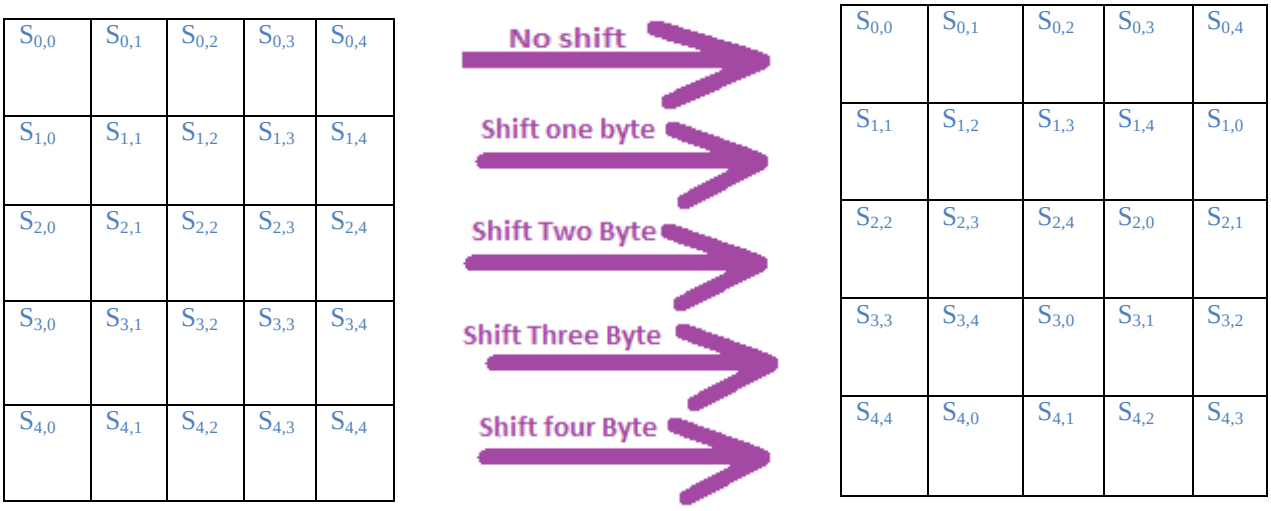

Fig.2 Shiftrows Transformation

2.3 MixColumns Transformation— It is the operation that mixes the bytes in each column by the multiplication of the state with a fixed polynomial matrix [2]. It completely changes the scenario of the cipher even if the all bytes look very similar. The Inverse Polynomial Matrix does exist in order to reverse the mix column transformation. Here, both of the matrixes are drawn below in fig.3 and fig.4 respectively.

$$\begin{vmatrix} 02 & 04 & 03 & 01 & 01 \\ 01 & 02 & 04 & 03 & 01 \\ 01 & 01 & 02 & 04 & 03 \\ 03 & 01 & 01 & 02 & 04 \\ 04 & 03 & 01 & 01 & 02 \end{vmatrix}$$

Fig 3: Polynomial Matrix for mix column transformation [3]

$$\begin{vmatrix} E0 & 7D & 09 & 8A & 4C \\ 4C & E0 & 7D & 09 & 8A \\ 8A & 4C & E0 & 7D & 09 \\ 09 & 8A & 4C & E0 & 7D \\ 7D & 09 & 8A & 4C & E0 \end{vmatrix}$$

Fig.4: Inverse Polynomial Matrix for mix column transformation [3].

2.4 AddRoundKey Transformation—In AddRoundKey transformation, a roundkey is added to the State by bitwise Exclusive-OR (XOR) operation.

Decryption Algorithm

Decryption is the process of extracting the plaintext from cipher text. The Decryption structure of proposed algorithm as shown in figure.5 is obtained by inverting the encryption structure which is shown above. Corresponding to the transformations in the encryption, InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey are the transformations used in the decryption as shown in Fig. below. The roundkeys are the same as those in encryption generated by Key Expansion, but are used in reverse order [1].

3.1 InvSubBytes Transformation—InvSubBytes is the inverse transformation of SubBytes, in which the inverse *S*-box is applied to individual bytes in the State. The inverse *S*-box is constructed by first applying the inverse of the affine transformation in (1), then computing the multiplicative inverse in GF(2^8). The inverse S-box used for this transformation is given in table 2 below:

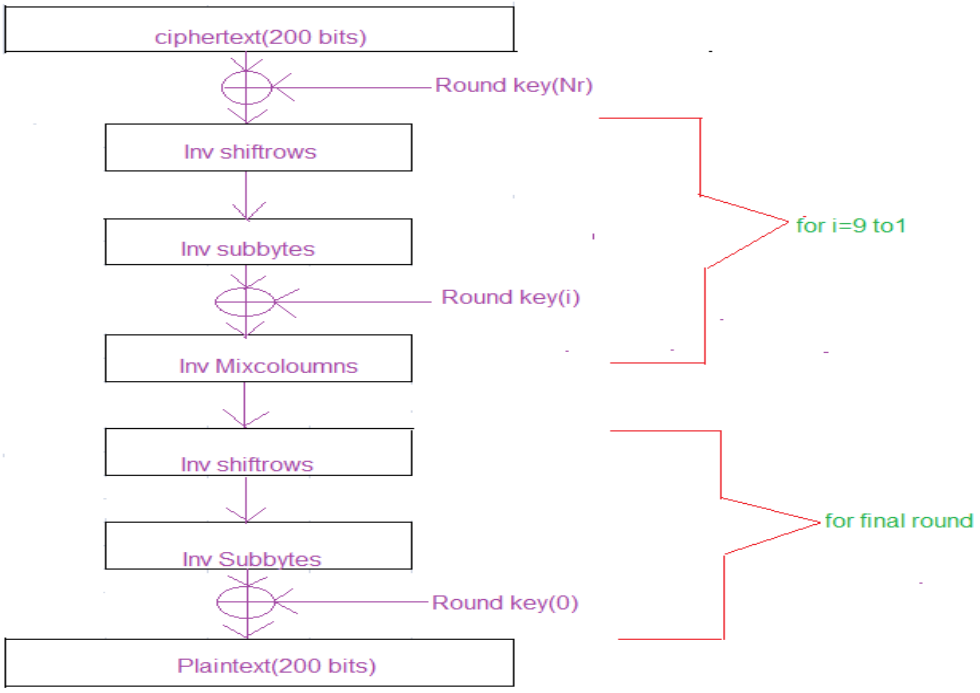

Fig: 5 Decryption Structure of Proposed AES algorithm.

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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
	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 2: Inverse S-Box

3.2 InvShiftRows Transformation—InvShiftRows is the inverse transformation of ShiftRows. In this transformation, the bytes in the first row of the State do not change; the second, third, and fourth and fifth rows are shifted cyclically by one byte, two bytes, three bytes and four bytes to the right respectively [1].

3.3 InvMixColumns Transformation— InvMixColumns is the inverse transformation of MixColumns. This is a complex procedure as it involves severely the byte multiplication under GF (2^8). The whole state is to be multiplied with pre-defined matrix called inverse polynomial matrix as illustrated in Figure 4.

Key Expansion:-Key expansion in AES is again a big task to perform, as it has several transformations. The key is expanded in the same manner as in conventional AES. The Pseudo code for key expansion is written below:

```
Key Expansion (byte Key [5*Nk] word W [Nb*(Nr+1)])
{
for (i = 0; i<Nk; i++)
W[i] = (Key [5*i], Key [5*i+1], Key [5*i+2], Key [5*i+3]), Key [5*i+4]);
for (i = Nk; i<Nb * (Nr + 1); i++)
{
 temp = W [i - 1];
if (i % Nk == 0)
 temp = SubByte(RotByte(temp)) ^ Rcon[i / Nk];
W[i] = W [i - Nk] ^ temp;
}
```

III. Experiment and Result

1. Encryption and decryption time: - The encryption and decryption time is one of the very important parameter while observing performance of any kind cipher. Fig 6.1 and Fig 6.2 below shows how much time the various AES standards will take in encrypting and decrypting the biggest size of data respectively.

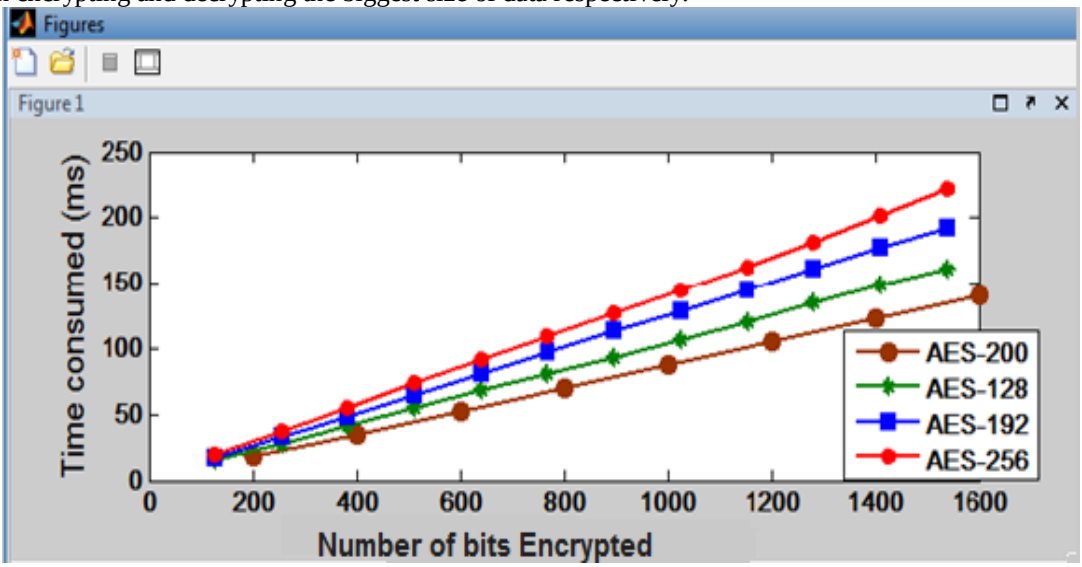

Fig 6.1: Comparison of encryption time of various AES algorithms for large data

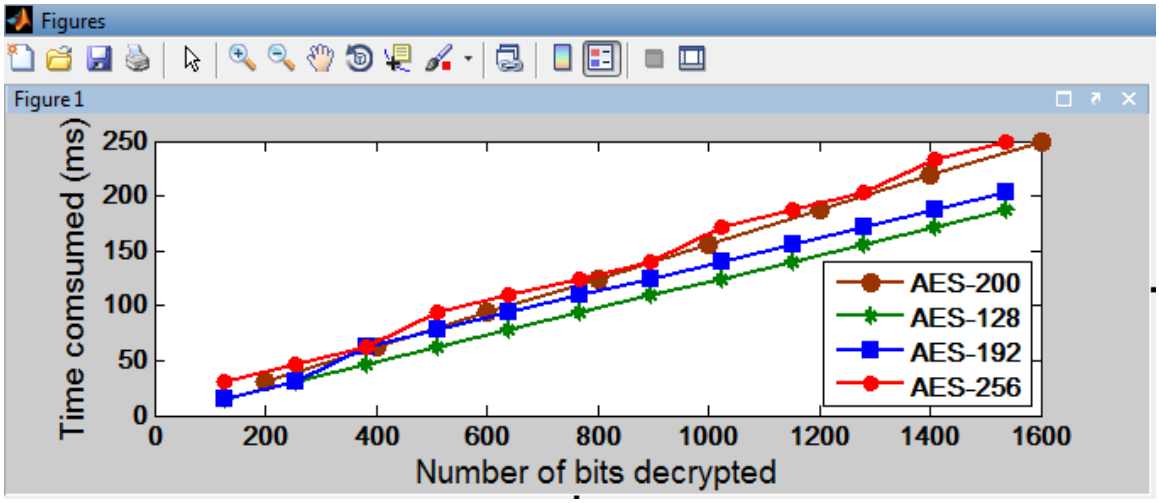

Fig 6.2: Comparison of Decryption time of various AES algorithms for large data

From the above graphs, it can be deduced that for large block of data AES-200 encryption time per bit is reduced up to 20% and decryption time per bit is increased up to 25%.

2. Throughput:- The throughput may be defined as number of bits that can be encrypted or decrypted during one unit of time. As it was mentioned earlier that all AES variant has equal block size of 128 bits and the proposed algorithm has block size of 200 bits. Thus, in form of equation the throughput may be defined as:

$$THR_{CA}=128/T_{ENC}$$

$$THR_{PA}=200/T_{PENC}$$

Where, THR_{CA} is representation of throughput for conventional algorithms, THR_{PA} is representation of throughput for proposed algorithm, T_{ENC}denotes the time taken to encrypt the 128 bit block message, T_{PENC} represents time taken to encrypt the 200 bit block message of conventional algorithm. In Fig 7.1 below, throughput for encryption side is drawn while the throughput at the decryption side is plotted in Fig 7.2.

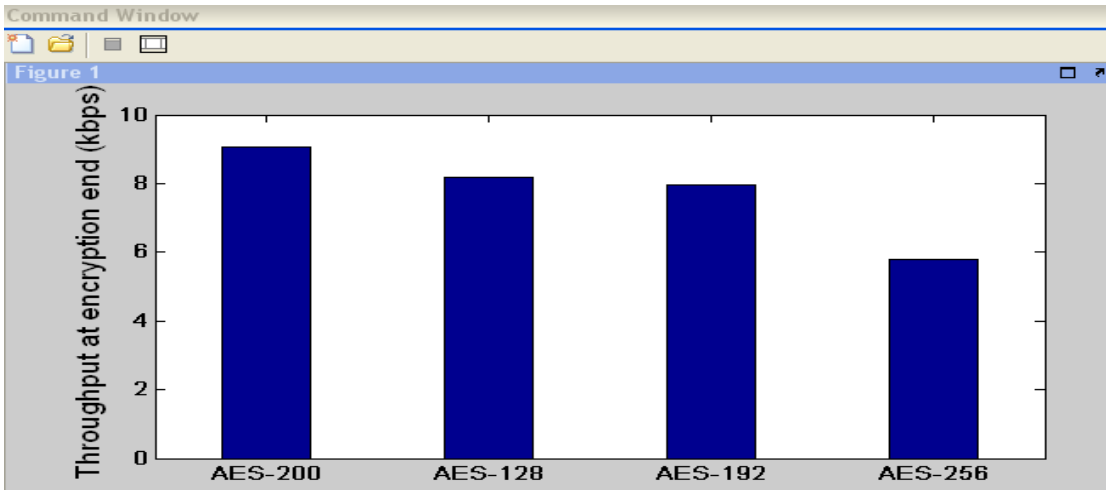

Fig 7.1: Comparison of throughput at encryption side of various AES standards.

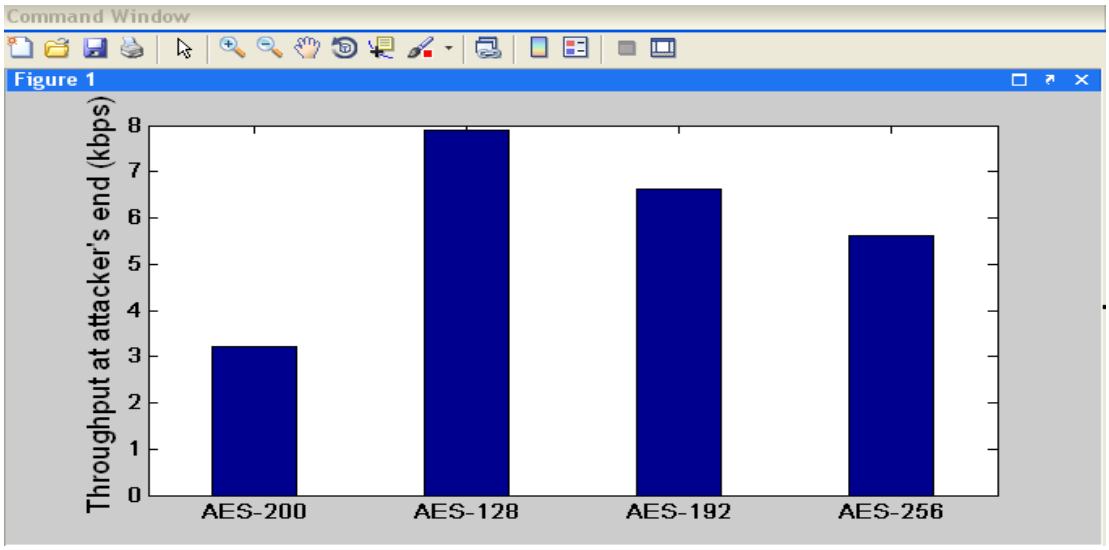

Fig 7.2: Comparison of throughput at decryption side of various AES standards.

From the above plots, it is observed that the throughput at encryption end of AES-200 is 15% more than AES-128, 20% more than AES-192 and 30% more than AES-256. The decryption process of AES-200 is slower than conventional AES. It can be seen from the graph that the proposed algorithm is 50% slower from AES-128, 40% from AES-192, and 25% from AES-256.

IV. Conclusion

This paper presented a new AES model having bigger block size which is 200 bits rather than conventional 128 bits AES. Also, the block is made by 5 rows and 5 columns unlike the AES's 4 rows and 4 columns. As the size of the matrix has increased, all the transformations of the AES don't need to change except the mix column transformation. During mix column transformation, the diffusion takes place in form of matrix multiplication under finite field. Having a bigger block, hence, requires a new matrix of size 5* 5, to enable matrix multiplication [3] .Here in this paper firstly we have compared the encryption and decryption time for various AES standards and then compared the same with our proposed algorithm time. We have concluded that for large block of data AES-200 encryption time per bit is reduced up to 20% and decryption time per bit is increased up to 25% than conventional AES. Secondly we compared the throughput of various AES standards and concluded that that the throughput at encryption end of AES-200 is 15% more than AES-128,

20% more than AES-192 and 30% more than AES-256. The decryption process of AES-200 is slower than conventional AES. The Security of the proposed model is examined by performing the test: Strict Avalanche Criterion and Bit Independence Criterion. SAC tells about the probability of the bit change while the BIC states the correlation that output bit possess. Both of the criteria are analyzed and the proposed algorithm falls within the desired level of security. Hence, it can be said that the proposed model is secure and can be considered for communication where high data rate is required.

References

Journals

[1] Xinmiao Zhang and Keshab K. Parhi,"Implementation approaches for the advanced encryption standard algorithm", IEEE Transactions 1531-636X/12©2002IEEE.

[2] Chih-Pin Su, Tsung-Fu Lin, Chih-Tsun Huang, and Cheng-Wen Wu, National Tsing Hua University,"A high throughput low cost AES processor" IEEE Communications Magazine 0163-6804/03 © 2003 IEEE.

[3] Navraj Khatri, Rajeev Dhanda , Jagtar Singh ,"Comparison of power consumption and strict avalanche criteria at encryption/Decryption side of Different AES standards''International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 4, August 2012.

[4] "Advanced Encryption Standard (AES)", Federal Information Processing Standards Publication 197, November 26, 2001.

[5] Chong Hee Kim,"Improved Differential Fault Analysis on AES Key Schedule" IEEE Transaction on Information Forensics and Security, Vol. 7, No. 1, Feb 2012.

[6] Irbid, Jordan, "A new approach for complex encrypting and decrypting data" International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.2, March 2013.

[7] Mohan H.S and A Raji Reddy,"Performance analysis of AES and MARS encryption algorithm" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011

[8] Amish Kumar , Mrs. Namita Tiwari,"Efficient implementation and avalanche effect of AES" International Journal of Security, Privacy and Trust Management (IJSPTM), Vol. 1, No 3/4, August 2012.

[9] Diaa Salama Abdul. Elminaam, Hatem M. Abdul Kader and Mohie M. Hadhoud," Performance Evaluation of Symmetric Encryption Algorithms on Power Consumption for Wireless Devices" International Journal of Computer Theory and Engineering, Vol. 1, No. 4, October, 2009.

Books

[10] Stallings W., Cryptography and Network Security, Third Edition, Pearson Education, 2003

[11] Atul Kahate, Cryptography.

