IOSR Journal of Computer Science (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727
PP 35-39
www.iosrjournals.org

Efficient Implementation of Aes By Modifying S-Box

Vijay L Hallappanavar[1], Basavaraj P Halagali[2], Veena V Desai[3]
[1]*KLES's College of Engineering & Technology, Chikodi, Karnataka*
[2]*V S M Institute of Technology, Nipani, Karnataka*
[3]*Gogte Institute of Technology, Belgaum, Karnataka*

ABSTRACT:*Security of data and messagesare an imperative issue because of fast evolution of digital data exchanges over unsecured network. Data security is achieved by methods of cryptography, which deals with encryption of data. Standard symmetric encryption algorithms provide better security for the multimedia data. In the existing AES algorithm we are modifying S-Box based on the some of the reference papers which we have gone through. After implementing modification the output is compared to original output, in terms of timing analysis, Hamming distance, Balanced output and Aavalanche effect and so on. Hence the efficiency of modified AES is verified after the comparison.*

Keywords **:***Avalanche effect,Hamming distance, Polynomial for S-box, Symmetric encryption,Swapping words in S-box*

I. Introduction

AES belongs to secret key cryptographic algorithm. With secret key cryptography, a single key is used for both encryption and decryption. The sender uses the key (or some set of rules) to encrypt the plaintext and sends the cipher text to the receiver. The receiver applies the same key (or rule set) to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called symmetric encryption. With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver.

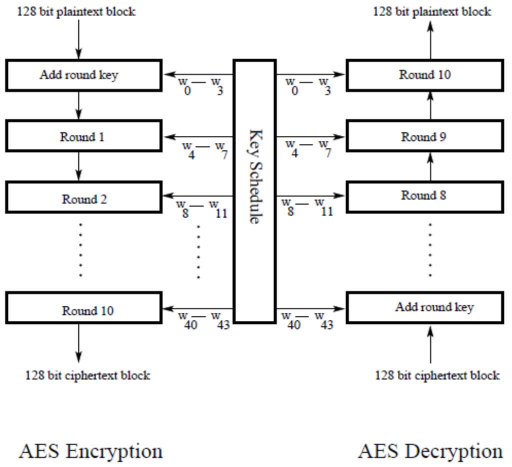
AES is a block cipher with a block length of 128 bits. AES allows for three different key lengths: 128, 192, or 256 bits. In this paper we have selected the key length of 128 bits. Encryption consists of 10 rounds of processing for 128-bit key, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Except for the last round in each case, all other rounds are identical. Each round of processing includes one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. The order in which these four steps are executed is different for encryption and decryption. To appreciate the processing steps used in a single round, it is best to think of a 128-bit block as consisting of a 4×4 matrix of bytes, arranged as follows:

.

$$\begin{bmatrix} Byte0 & Byte4 & Byte8 & Byte12 \\ Byte1 & Byte5 & Byte9 & Byte13 \\ Byte2 & Byte6 & Byte10 & Byte14 \\ Byte3 & Byte7 & Byte11 & Byte15 \end{bmatrix}$$

Therefore, the first four bytes of a 128-bit input block occupy the first column in the 4 × 4 matrix of bytes. The next four bytes occupy the second column, and so on. The 4 × 4 matrix of bytes is referred to as the state array.AES also has the notion of a word.A word consists of four bytes that is 32 bits. Therefore, each column of the state array is a word, as is each row. Each round of processing works on the input state array and produces an output state array. The output state array produced by the last round is rearranged into a 128-bit output block.

The Advanced Encryption Standard comprises three block ciphers, AES-128. AES has a fixed block size of 128 bits and a key size of 128,192 or 256 bits. Block size has a maximum of 256 bits, but the key size has no theoretical maximum. Encryption consists of 10 rounds of processing for 128 bit keys, 12 rounds of 192 bit keys and 14 rounds of 256 bit keys.



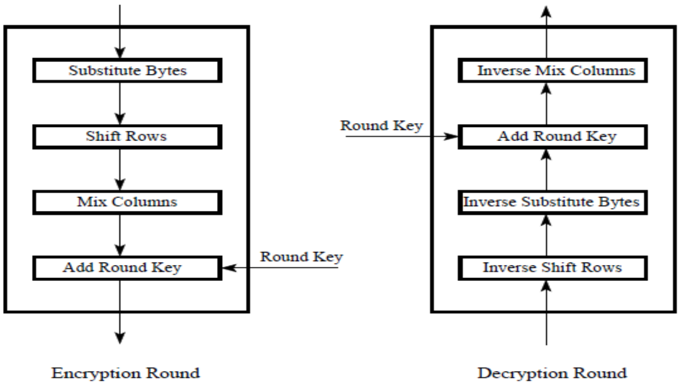


Fig 1: Block diagram of AES Fig 2: Steps in each round of AES

The cipher uses number of encryption rounds which converts plain text to cipher text. The output of each round is the input to the next round. The output of the final round is the encrypted plaintext known as cipher text. The input given by the user is entered in a matrix known as State Matrix. Following are the four steps used for encryption:

1.1 SubBytes Step

SubBytes, also known as Byte substitution are the first iterative step of the algorithm in each round. In the Sub Byte step, each byte in the matrix is recognized using an 8-bit substitution box. This substitution box is called Rijndael S-box. This operation provides the non linearity in cipher. The S-box used is derived from multiplicative inverse over GF(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points.

1.2 Shift Rows Step

The shift row step is performed on the rows of the state matrix. It cyclically shifts the bytes in each row by a certain offset. The first remains unchanged. Each byte of the second row is shifted one position to the left. Similarly, the third and fourth rows are shifted by two positions and three positions respectively. The shifting pattern for block of size 128 bits and 192 bits in the same.

1.3 Mix column Step

In the Mix column step, the four bytes of each column of the state matrix are combined using an invertible linear transformation. A randomly generated polynomial is arranged in 4 x 4 matrix. The same polynomial is used during decryption. Each column of the state matrix is XOR-ed with the corresponding column of the polynomial matrix. The result is updated in the same column. The output matrix is the input to Add Round Key.

1.4 Add Round Key

A round key is generated by performing various operations on the cipher key. This round key is XOR-ed with each byte of state matrix. For every round a new round key is generated using some operations on the cipher key.

For decryption, each round consists of following four steps: 1) Inverse shift rows, 2) Inverse substitute bytes, 3) Add round key, 4) Inverse mix column. The third step consists of XORing the output of previous two steps with four words from key schedule. Note the differences between the order in which substitution and shifting are carried out in a decryption round vis-à-vis the order in which similar operations are carried out in an encryption round.The last round for encryption does not involve the "Mix column" step. The last round for decryption does not involve "Inverse Mix column" step.

I. MODIFICATION IN AES

The advance encryption standard (AES) algorithm considered to be the leader of this innovation. A large variety of approaches for modifying AES have been appeared so as to satisfy the varying criteria of different applications. Effort spent on evaluating AES security strength has been very limited, it is significant that the majority of researches at AES dealt with performance evaluation rather than with security evaluation. The design and implementation of S-box are representing a key step in the AES algorithm. So many efforts were emulated to redesign, reconstruct or renew the implementation of the S-boxes.

2.1New S-box Construction

Four irreducible polynomials shown in figure 3 can be used to create new substitution tables (S boxes). Fig. 4 and 5 describe the S-box and its inverse for the polynomial ($X^8+X^6+X^5+X+1$).

No.	Polynomial
1.	$X^8 + X^6 + X^5 + X+1$
2.	$X^8 + X^5+X^4+X^3+X^2+X+1$
3.	$X^8+ X^6+X^5+X^2+X+1$
4.	$X^8+ X^6+X^5+X+1$

Fig 3 : Different polynomial used to create new S-boxes

63	7C	E1	60	2F	62	E2	68	48	6C	E3	34	AE	29	E6	95
FB	CE	E9	8F	2E	0D	C5	53	85	4D	46	66	A1	A7	15	EB
22	1B	B8	16	2B	A5	18	D6	C8	9D	54	79	3D	9F	76	65
1D	17	74	0E	F1	31	EC	51	0F	8C	01	AB	58	43	2A	B0
C3	DB	52	6A	83	32	D9	8A	47	EA	00	30	D3	D2	B4	B7
B6	81	11	4F	F8	B1	6E	02	41	7B	10	96	E4	8D	6D	5B
5C	F5	59	4B	E5	B9	D5	40	27	06	4A	A9	A4	C4	77	21
55	9E	99	56	5F	05	0A	37	F3	2C	7E	C0	C7	9C	87	14
33	4E	3F	CC	F6	B3	E7	36	13	98	CB	AF	3E	FD	97	86
71	08	AA	C1	DF	70	CA	1A	3B	A6	BB	DC	88	49	09	BE
89	93	12	EE	57	84	75	9A	A3	8B	07	DD	E8	BC	DE	23
7F	5D	6F	61	D7	67	94	F7	A0	38	19	BF	69	25	72	F0
FC	BA	28	F4	73	9B	7A	3A	20	CD	03	A2	35	D4	FF	5A
4C	D0	D1	92	FA	24	0B	0C	80	04	BD	ED	64	AD	42	FE
78	82	90	AC	1E	A8	F9	C6	7D	1F	50	6B	DA	CF	44	EF
26	39	C9	C2	E0	8E	B2	5E	3C	B5	91	45	1C	F2	D8	2D

4A	3A	57	CA	D9	75	69	AA	91	9E	76	D6	D7	15	33	38
5A	52	A2	88	7F	1E	23	31	26	BA	97	21	FC	30	E4	E9
C8	6F	20	AF	D5	BD	F0	68	C2	0D	3E	24	79	FF	14	04
4B	35	45	80	0B	CC	87	77	B9	F1	C7	98	F8	2C	8C	82
67	58	DE	3D	EE	FB	1A	48	08	9D	6A	63	D0	19	81	53
EA	37	42	17	2A	70	73	A4	3C	62	CF	5F	60	B1	F7	74
03	B3	05	00	DC	2F	1B	B5	07	BC	43	EB	09	5E	56	B2
95	90	BE	C4	32	A6	2E	6E	E0	2B	C6	59	01	E8	7A	B0
D8	51	E1	44	A5	18	8F	7E	9C	A0	47	A9	39	5D	F5	13
E2	FA	D3	A1	B6	0F	5B	8E	89	72	A7	C5	7D	29	71	2D
B8	1C	CB	A8	6C	25	99	1D	E5	6B	92	3B	E3	DD	0C	8B
3F	55	F6	85	4E	F9	50	4F	22	65	C1	9A	AD	DA	9F	BB
7B	93	F3	40	6D	16	E7	7C	28	F2	96	8A	83	C9	11	ED
D1	D2	4D	4C	CD	66	27	B4	FE	46	EC	41	9B	AB	AE	94
F4	02	06	0A	5C	64	0E	86	AC	12	49	1F	36	DB	A3	EF
BF	34	FD	78	C3	61	84	B7	54	E6	D4	10	C0	8D	DF	CE

Fig 4: S-box generated from the polynomial ($X^8+X^6+X^5+X+1$) Figure 5 : Inverse S-box generated from the polynomial

($X^8+X^6+X^5+X+1$)

2.1 Modified AES S-box mapping Calculation

In AES we are constructing S-box and inverse S-box using new polynomial instead of standard polynomial and for modification in AES we are modifying the S-box and Inv S-box by swapping each word of S-box and Inv_s-box generated by new polynomial.

2.1.1 Modification In S_Box Using New Polynomial

36	C7	1E	06	F2	26	2E	86	84	C6	3E	43	EA	92	6E	59
BF	EC	9E	F8	E2	D0	5C	35	58	D4	64	66	1A	7A	51	BE
22	B1	8B	61	B2	5A	81	6D	8C	D9	45	97	D3	F9	67	56
D1	71	47	E0	1F	13	CE	15	F0	C8	10	BA	85	34	A2	0B
3C	BD	25	A6	38	23	9D	A8	74	AE	00	03	3D	2D	4B	7B
6B	18	11	F4	8F	1B	E6	20	14	B7	01	69	4E	D8	D6	B5
C5	5F	95	B4	5E	9B	5D	04	72	60	A4	9A	4A	4C	77	12
55	E9	99	65	F5	50	A0	73	3F	C2	E7	0C	7C	C9	78	41
33	E4	F3	CC	6F	3B	7E	63	31	89	BC	FA	E3	DF	79	68
17	80	AA	1C	FD	07	AC	A1	B3	6A	BB	CD	88	94	90	EB
98	39	21	EE	75	48	57	A9	3A	B8	70	DD	8E	CB	ED	32
F7	D5	F6	16	7D	76	49	7F	0A	83	91	FB	96	52	27	0F
CF	AB	82	4F	37	B9	A7	A3	02	DC	30	2A	53	4D	FF	A5
C4	0D	1D	29	AF	42	B0	C0	08	40	DB	DE	46	DA	24	EF
87	28	09	CA	E1	8A	9F	6C	D7	F1	05	B6	AD	FC	44	FE
62	93	9C	2C	0E	E8	2B	E5	C3	5B	19	54	C1	2F	8D	D2

Fig 6 : Modified S-box generated by swapping each word

4A	5A	C8	4B	67	EA	03	95	D8	E2	B8	3F	7B	D1	F4	BF
3A	52	6F	35	58	37	B3	90	51	FA	1C	55	93	D2	02	34
57	A2	20	45	DE	42	05	BE	E1	D3	CB	F6	F3	4D	06	FD
CA	88	AF	80	3D	17	00	C4	44	A1	A8	85	40	4C	0A	78
D9	7F	D5	0B	EE	2A	DC	32	A5	B6	6C	4E	6D	CD	5C	C3
75	1E	BD	CC	FB	70	2F	A6	18	0F	25	F9	16	66	64	61
69	23	F0	87	1A	73	1B	2E	8F	5B	99	50	E7	27	0E	84
AA	31	68	77	48	A4	B5	6E	7E	8E	1D	4F	7C	B4	86	B7
91	26	C2	B9	08	3C	07	E0	9C	89	E5	22	28	FE	AC	54
9E	BA	0D	F1	9D	62	BC	2B	A0	72	6B	65	F2	46	12	E6
76	97	3E	C7	6A	CF	43	C6	47	A7	92	C1	96	EC	49	D4
D6	21	24	98	63	5F	EB	59	A9	C5	3B	9A	8A	41	1F	10
D7	FC	79	F8	D0	60	09	01	39	7D	E3	AD	83	9B	36	C0
15	30	FF	2C	19	B1	5E	E8	5D	29	DD	DA	C9	AB	DB	8D
33	E4	14	8C	81	F7	56	7A	F5	71	0C	9F	11	AE	A3	DF
38	E9	04	82	53	74	B2	B0	13	2D	8B	BB	ED	94	EF	CE

Figure 7: Modified inv_s_box generated by swapping each word

In standard AES we are using $X^8+X^4+X^3+X+1$ for generating S-box. In the modification we used$X^8+X^6+X^5+X+1$ polynomial equation and each word will be swapped by own word.

III. RESULT ANALYSIS

3.1Comparisonofboththealgorithms

3.1.1HammingDistance: The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. In another way, it measures the minimum number of *substitutions* required to change one string into the other, or the minimum number of *errors* that could have transformed one string into the other.

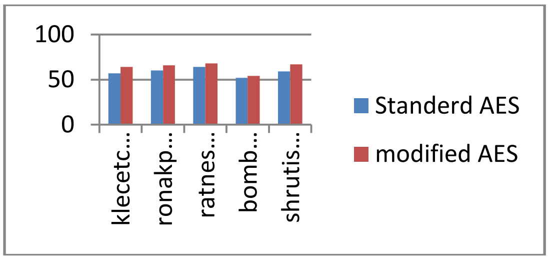


Fig 8 : Bar graph for hamming distance output table

3.1.2 Balanced Output **:** Balanced output is the output which has the number of 1's & 0's should be nearly same.

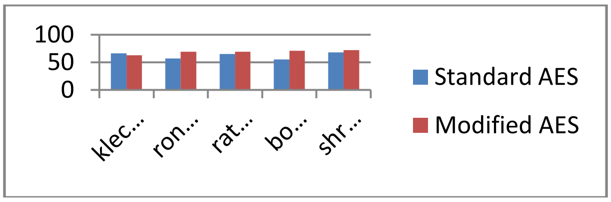


Fig 9: Bar graph for Balanced output table

3.1.3Avalancheeffect:

The avalanche effect is evident if, when an input is changed slightly the output changes significantly. In the case of quality block ciphers, such a small change in either the key or the plain text should cause a drastic change in the cipher text.

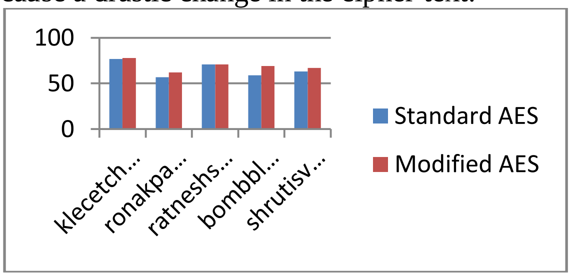


Fig 10 : Bar graph for Avalanche effect output table

IV. CONCLUSION

Usually lightweight encryption algorithms are very attractive for multimedia applications. Luckily we have achieved through our research a fast light weight encryption algorithm to secure our multimedia data from unauthorized access. For the security of multimedia data, we have proposed a Modified AES with change in S-Box with gives better results than the standard AES. Theoretical analysis and experimental results of the achievement makes it very suitable for high rate and less overhead on the data. For all these compensation it is suitable for any large scale text and image transfer. From the results we can conclude that the output from modified AES is better than standard AES in terms Hamming Distance, Balanced Output and Avalanche Effect.

REFERENCES

[1] Lai, Xuejia, and Massey, James L., A Proposal for a New Block Encryption Standard, Advances in Cryptology - EUROCRYPT '90, Lecture Notes in Computer Science, Springer-Verlag, 1991: 389-404.
[2] Mediacrypt AG, The IDEA Block Cipher, submission to the NESSIE Project, http://cryptonessie.org
[3] Meier, W., On the Security of the IDEA block cipher, Advances in Cryptology
[4] Menezes, A., van Oorschot, P., and Vanstone, S. 1996. Handbook of Applied Cryptography. CRC Press. This book may downloaded from http://www.cacr.math.uwaterloo.ca/hac/
[5] Musa, M., Shaefer, E., and Wedig S. 2003. A Simplified AES Algorithm and its Linear and Differential Cryptanalysis. Cryptologia. 17 (2): 148 - 177.
[6] Phan, R. 2002. Mini Advanced Encryption Standard (Mini-AES): A Testbed for Cryptanalysis Students. Cryptologia. 26 (4): 283 - 306.
[7] Schaefer, E. 1996. A Simplified Data Encryption Standard Algorithm. Cryptologia. 20 (1): 77 - 84.
[8] Schneier, B. 1996. Applied Cryptography, Second Edition. Wiley.