Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 3, Issue 5, May-2016

IMPLEMENTATION OF ADVANCED ENCRYPTION STANDARD IN SIMPLE, COMPOSITE AND EXTENDED MODE

[1]Ms. Suman S. B. ,[2]Mrs. Channakka Lakkannavar
[1] P.G. Student
[2] Assistant Professor
ECE Department
SDM College of Engineering and Technology, Dharwad

Abstract--Cryptography is a technique related to aspects of information security such as data confidentiality, data integrity and entity authentication. In data and telecommunication systems, Security is the most important part for an effective communication, where to increase the security as well as complexity, more randomization in secret keys is necessary to enhance the cryptography algorithms. In AES, Even though the round keys have high security, Power analysis attack and Saturation attack are effective to the key expansion algorithm of AES due to reducible key rounds and it leads to security problems. AES in simple, composite and extended mode, the main objective of the AES in extended field is to reduce the memory required to perform sub bytes by replacing the two 256 byte lookup table (LUT's) that is one for encryption and the other for decryption. The cost of this replacement leads to an increase in combinational logic which is used to perform affine transformation. The field over GF(28) called an extended field while the more compact representation GF(24)2 is called a composite field. It reduces the size of LUT's to 8 bytes. The designs presented in this paper is simulated using Xilinx 14.2 software.

Key Terms—Advanced encryption standard, key expansion, simple, composite and extended etc.

I. INTRODUCTION

The word cryptography comes from the Greek words κρυπτο which means hidden or secret and γραφη means writing. Cryptography is the art of secret writing. Generally, many people think of cryptography as the art of mangling information into apparent unintelligibility in a manner by allowing a secret method of unmangling. The basic service provided by cryptography is the ability to send information between the users in a way which prevents others from reading it. Cryptography is based on representing the information as numbers and mathematically manipulating those numbers. This kind of cryptography can provide other services, such as

• Integrity checking—reassuring the recipient of a message that the message has not been changed since it was generated by a legitimate source.

•Authentication- verifying someone's identity. A message in its original form is known as **plain-text** or **clear-text**. The mangled information is known as **cipher-text**. The process for producing Cipher-text from plaintext is known as **encryption**. The reverse of encryption is called **decryption**.

Advanced encryption standard (AES) was published by NIST(national institute of standard and technology) in 2001. AES is symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192 and 256 bits which replaced DES (data encryption standard) as an approved standard for a wide range of application which can encipher and decipher the data. The AES algorithm consists of three main parts:

1. Cipher (encryption).
2. Inverse cipher (decryption).
3. Key expansion.

Encryption converts the data into indiscernible form is called as cipher-text. Decryption converts the cipher text into its indigenous form is called as decryption. Key expansion is used for generating the keys for 10 rounds which is produced by the original input key every round keys are different from one another in order to improve the security of the algorithm it is essential to improve the performance of the key expansion from the external attacks.

In some applications, such as smart cards, relatively small amounts of random access memory (RAM) and or read only memory (ROM) are available for such purpose as code storage (generally in form); Representation of data object such as S-boxes and sub key storage (RAM).

II. ADVANCED ENCRYPTION STANDARD ALGORITHM

2.1 AES Specification:

The Rijndael proposal for AES defined a cipher in which the block length and the key length can be specified into 128, 192, or 256 bits. The AES specification uses the three key size alternatives but limits the length of the block to 128 bits.
Rijindael is designed to have the following characteristics:

- Resistance against all known attacks.
- Speed and code compactness on a wide range of platforms.
- Design simplicity.

The length of the key is represented by Nb=4.the input 128 bits is arranged in 4*4 matrix into 16 bytes which reflects the number of 32 bit-bit words in the state.

Key size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plain text block(words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of rounds	10	12	14
Round key size(words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded key size(words/bytes)	44/176	52/208	60/240

Table 1: AES parameters

2.2 Modular inversion in extended field:

There are two fundamental steps performed in the subbytes transformation. The first step is the most complex and the most difficult to implement in hardware. The second step is the affine transformation. The inverse subyte transform is simply the two step reversed. Therefore during decryption the inverse subbytes step executes the inverse affine transform and then determines the modular inverse. It is important because it allows entire subbyte operation to use one 256 byte look up table containing all modular inverses in a Rijndael field. The affine transform used during subbyte is shown below, Where b is the affine transform. The coefficient of the polynomial is transformed and multiplied with a binary matrix, the result of which is XORed with a constant bit pattern (011000110). If each coefficient in b is treated as separate entity rather than as elements of a column vector, it is possible to map a unique equation for each coefficient of b. It can be achieved by a ordinary matrix multiplication 8*8 binary matrix and the vector of a coefficients.

Affine transform used during encryption

$$\begin{pmatrix} a7 \\ a6 \\ a5 \\ a4 \\ a3 \\ a2 \\ a1 \\ a0 \end{pmatrix} = \begin{pmatrix} 0&1&0&1&0&0&1&0 \\ 0&0&1&0&1&0&0&1 \\ 1&0&0&1&0&1&0&0 \\ 0&1&0&0&1&0&1&0 \\ 0&0&1&0&0&1&0&1 \\ 1&0&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&1 \\ 1&0&1&0&0&1&0&0 \end{pmatrix} \times \begin{pmatrix} b7 \\ b6 \\ b5 \\ b4 \\ b3 \\ b2 \\ b1 \\ b0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Inverse affine transform used during decryption

$$\begin{pmatrix} b7 \\ b6 \\ b5 \\ b4 \\ b3 \\ b2 \\ b1 \\ b0 \end{pmatrix} = \begin{pmatrix} 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \\ 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \end{pmatrix} \times \begin{pmatrix} a7 \\ a6 \\ a5 \\ a4 \\ a3 \\ a2 \\ a1 \\ a0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

The objective of this design is to reduce the memory required to perform subbytes by replacing the two 256 byte LUT's with one byte LUT that can be used for encryption and decryption. The cost of this replacement is an increase in combinational logic used to perform the affine transformations.

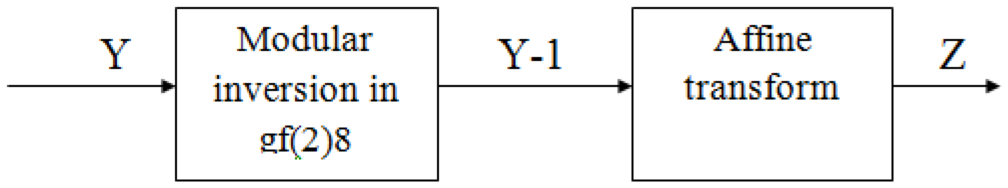


Fig.1. Subbyte design flow for modular inversion in extended field

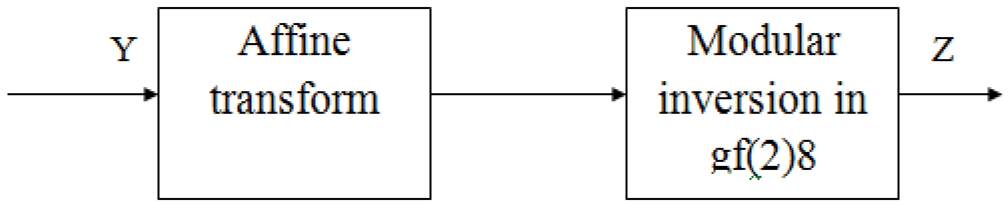


Fig.2. Inverse subbyte design flow for modular inversion in extended field

2.3 Modular inversion in composite field:

The field over GF(28) called an extended field while the more compact representation GF(24)2 is called a composite field. The field GF(24) is referred as a sub field. In order to perform the modular inversion in extended field requires a 256 byte LUT. An equivalent representation of the extended Rijndael field into a more compressed subfield over GF(24)2 reduces the size of the LUT to just 8 bytes.

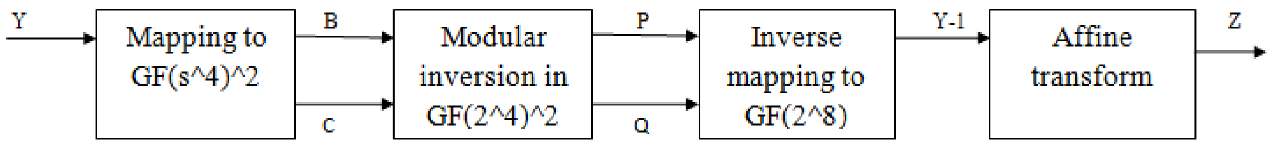


Fig.3. Subbyte design flow for composite field

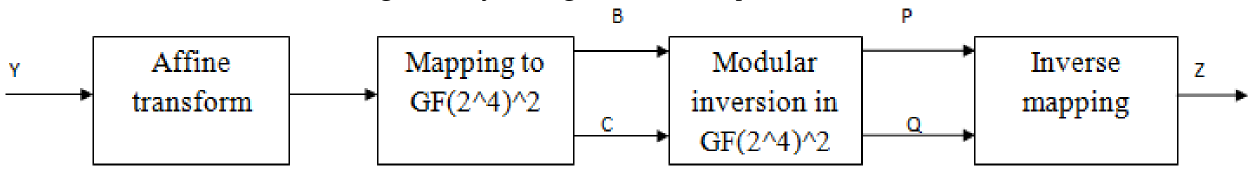


Fig.4. Inverse subbyte design flow for composite field

Paar and Rosner map a finite field over GF(2^8) into an equivalent representation over GF(2^4)2 using a transformation matrix. Rudra used the Paar and Rosner method to develop an algorithm for determining transformation matrices specifically for the Rijndael field. This algorithm composed of three steps that ensure the matrix has the required mathematical properties. The mapping equation with Rudra's transformation matrix is

$$\begin{pmatrix} b3 \\ b2 \\ b1 \\ b0 \\ c3 \\ c2 \\ c1 \\ c0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} y7 \\ y6 \\ y5 \\ y4 \\ y3 \\ y2 \\ y1 \\ y0 \end{pmatrix}$$

Mapping equation

$$\begin{pmatrix} y7 \\ y6 \\ y5 \\ y4 \\ y3 \\ y2 \\ y1 \\ y0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} b3 \\ b2 \\ b1 \\ b0 \\ c3 \\ c2 \\ c1 \\ c0 \end{pmatrix}$$

The inverse mapping equation

As the number of 1's in a matrix increase, so the XOR operations required to perform the matrix multiplication. The transformation matrix in the above contains 29 '1's for a total of 57 '1's in both matrices.

3. ENCRYPTION PROCESS:

Cipher is composed of four different byte oriented transformation.

1. Byte substitution using s-box look up table
2. Row wise permutation of the state array
3. Column wise mixing within each column of the state array.
4. Addition of round key to the state

The structure of AES algorithm for both encryption and decryption is shown in fig.6 The first step in AES encryption is the addition or xoring of original key to the input data, which is called as an initial round, which is followed by nine iteration of a normal round and ends with a modified final round. During each normal round the following operation are performed in the following order that is byte substitution, shift rows, mix columns and addition of the round key. The final round is also a normal round without the column wise mixing process.

3.1 **Byte substitution**: This is byte by byte substitution process which is shown in the fig. 2. The substitution for each input byte is found by using the s-box lookup table. The size of the s- box look up table is 16×16 in order to find the substitute byte for a given input byte, we split the input byte into two four bit patterns, it leads to an integer value between 0 and 15 these can be represented in hex as 0 to F. The first value of the hex is used as a row index and the second value of the hex is used as column index for reaching into the 16×16 s-box look up table.

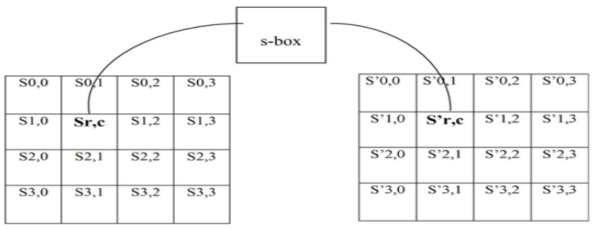


Fig .5 byte substitution

The entries in the lookup table are constructed by a combination of GF (2)^8arithmetic and bit scrambling. The goal of the substitution step is to reduce the correlation between the input bits and the output bits.

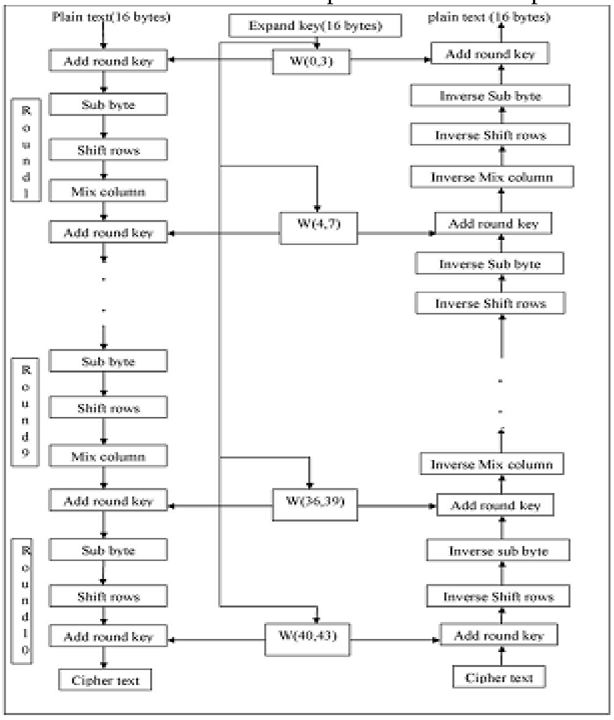


Fig. 6. Structure of AES encryption and decryption

2.4 Row wise permutation: It consist of

i. The first row of the state array is not shifted at all.
ii. The second row is circularly shifted by one byte to the left.
iii. The third row is circularly shifted by two bytes to the left.
iv. The last row is circularly shifted by three bytes to the left.

The row wise permutation can be represented by

$$\begin{pmatrix} S0,0 & S0,1 & S0,2 & S0,3 \\ S1,0 & S1,1 & S1,2 & S1,3 \\ S2,0 & S2,1 & S2,2 & S2,3 \\ S3,0 & S3,1 & S3,2 & S3,3 \end{pmatrix} ===== \begin{pmatrix} S'0,0 & S'0,1 & S'0,2 & S'0,3 \\ S'1,1 & S'1,2 & S'1,3 & S'1,0 \\ S'2,2 & S'2,3 & S'2,0 & S'2,1 \\ S'3,3 & S'3,0 & S'3,1 & S'3,2 \end{pmatrix}$$

	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	37	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	86	6E	54	A0	52	3B	D6	E3	29	E3	2F	84
56	53	D1	00	Ed	FD	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	Aa	Fb	8C	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8f	CA	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	Ec	4F	97	44	70	C4	A7	7E	30	64	5D	19	73
9	60	81	4f	Dc	E7	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3a	0a	1D	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	Bd	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	1c	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	48	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

Table 2: S-box lookup table

3.3 **Column wise mixing:** This replaces each byte of a column byte a function of all the bytes in the same column.

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 02 & 01 & 01 \end{pmatrix} \times \begin{pmatrix} S0,0 & S0,1 & S0,2 & S0,3 \\ S1,0 & S1,1 & S1,2 & S1,3 \\ S2,0 & S2,1 & S2,2 & S2,3 \\ S3,0 & S3,1 & S3,2 & S3,3 \end{pmatrix} = \begin{pmatrix} S'0,0 & S'0,1 & S'0,2 & S'0,3 \\ S'1,0 & S'1,1 & S'1,2 & S'1,3 \\ S'2,0 & S'2,1 & S'2,2 & S'2,3 \\ S'3,0 & S'3,1 & S'3,2 & S'3,3 \end{pmatrix}$$

The operation for the bytes in the first row can be written as

$S'0,c = (\{02\} \cdot S0,c) + (\{03\} \cdot S1,c) + S2,c + S3,c$..(1)

The operation for the bytes in the second row can be written as
$S'1,c = S0,c + (\{02\} \cdot S1,c) + (\{03\} \cdot S2,c) + S3,c$..(2)

The operation for the bytes in the third row can be written as
$S'2,c = S0,c + S1,c + (\{02\} \cdot S2,c) + (\{03\} \cdot S3,c)$..(3)

The operation for the bytes in the fourth row can be written as
$S'3,c = (\{03\} \cdot S0,c) + S1,c + S2,c + (\{02\} \cdot S3,c)$..(4)

2.5 Addition of round key: The key words are generated by key expansion process. Round key is added to every state by a XOR operation in which each round key consist of Nb words. These Nb words are now added to the columns of the state, [Wi] are the key scheduled words and round is the value in the range 0 round Nr. In the cipher, the first round key addition (XOR) occurs when the round=0, foregoing to the first application of round function add round key transformation to the Nr rounds of the cipher occurs when 1< round < Nr. simple XOR operation can be shown in fig.7.

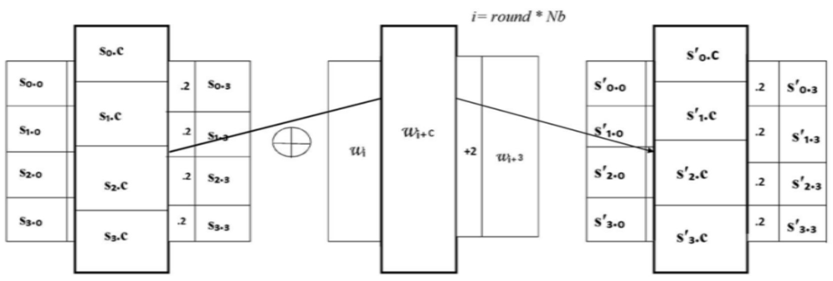


Fig. 7. Addition of round key XOR's each column of the state with a word

4. TRADITIONAL AES KEY EXPANSION

5.

The key expansion algorithm takes input of 128-bits as 16-bytes which is represented by k0,k1………….k15. The first four key words W0,W1,W2,W3 are obtained from those columns which is shown in the Fig.4. This original key words are sufficient to generate add round keys of each ten rounds of the cipher.

In the traditional key expansion algorithm, suppose the original key words are(W0,W1,W2,W3), it generates the sub key words (W4, W5,...........,W43), for 10 rounds.

Original key: W0, W1, W2, W3
1st round key: W4, W5, W6, W7
2nd round key: W8, W9, W10, W11
.....
10th round: W40, W41, W42, W43

The key expansion process of the first round is

- W4= W0+ subword (rotword(W3))+ Rcon(1)
- W5= W1+ W4
- W6= W2+ W5
- W7= W3+ W6

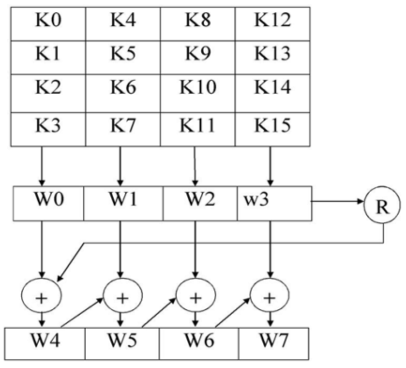


Fig.9. Key expansion process

Except the first round all the rounds will be same up to tenth round as follows.

- W8= W4+ subword (rotword(W7))+ Rcon(2)
- W9= W5+ W8
- W10= W6+ W9
- W11= W7+ W10

Rotword performs a circular left shift. It means that input word [B0, B1, B2, B3] is transformed in to [B1, B2, B3, B0]. Subword performs a byte substitution on each byte of its input word, using the S-box. Fig.8. shows the process of function g.

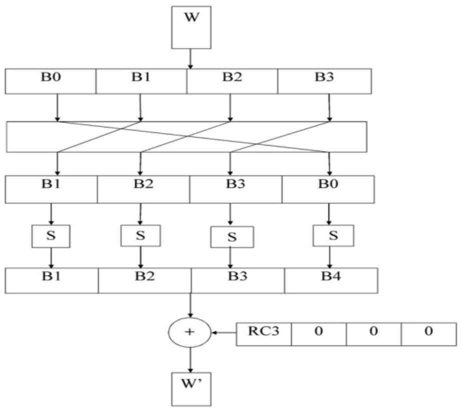


Fig.8. Function g block.

First four words of the expanded key are defined from the original keys. From the second round each W[i] depends on preceding word W [i-1] from the previous round

5. DECRYPTION PROCESS

Decryption process is the reverse operation of the AES encryption. It starts with an initial round and followed by nine iteration of an inverse round and ends with an addition of round key. The decryption round consists of the following operation: addition of round key, inverse column wise mixing, inverse row wise permutation and inverse substitution bytes.

5.1 Inverse row permutation: In decryption process the shifts in the corresponding operation shifts in opposite manner. The first row is left unchanged, the second row is shifted to the right by one byte, the third row is shifted to the right by two bytes and the last row of the state is shifted to the right by three bytes.

$$\begin{pmatrix} S0,0 & S0,1 & S0,2 & S0,3 \\ S1,0 & S1,1 & S1,2 & S1,3 \\ S2,0 & S2,1 & S2,2 & S2,3 \\ S3,0 & S3,1 & S3,2 & S3,3 \end{pmatrix} === \begin{pmatrix} S0,0 & S0,1 & S0,2 & S0,3 \\ S1,3 & S1,0 & S1,1 & S1,2 \\ S2,2 & S2,3 & S2,0 & S2,1 \\ S3,1 & S3,2 & S3,3 & S3,0 \end{pmatrix}$$

5.2 Inverse column wise mixing: it is the inverse process of the column wise mixing that operates on the state column by column as a four term polynomial. These 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^3 + {0d} x^2 + {09} x + {0e}, this can be written in matrix multiplication as

$$\begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \times \begin{pmatrix} S0,0 & S0,1 & S0,2 & S0,3 \\ S1,0 & S1,1 & S1,2 & S1,3 \\ S2,0 & S2,1 & S2,2 & S2,3 \\ S3,0 & S3,1 & S3,2 & S3,3 \end{pmatrix} = \begin{pmatrix} S'0,0 & S'0,1 & S'0,2 & S'0,3 \\ S'1,0 & S'1,1 & S'1,2 & S'1,3 \\ S'2,0 & S'2,1 & S'2,2 & S'2,3 \\ S'3,0 & S'3,1 & S'3,2 & S'3,3 \end{pmatrix}$$

5.3 Inverse substitution bytes: The inverse substitution byte for each input byte is by using the inverse s-box lookup table. The size of the lookup table is 16×16. In order to find the inverse substitution byte for a given input byte, we split the input byte into two 4-bit patterns, each yielding an integer value between 0 and 15 which can be represented by hex values 0 through F. First value of the hex is used as a row index and the other as a column index for reaching into the 16×16 lookup table. The entries in the lookup table are constructed by a multiplicative inverse of GF (2^8).

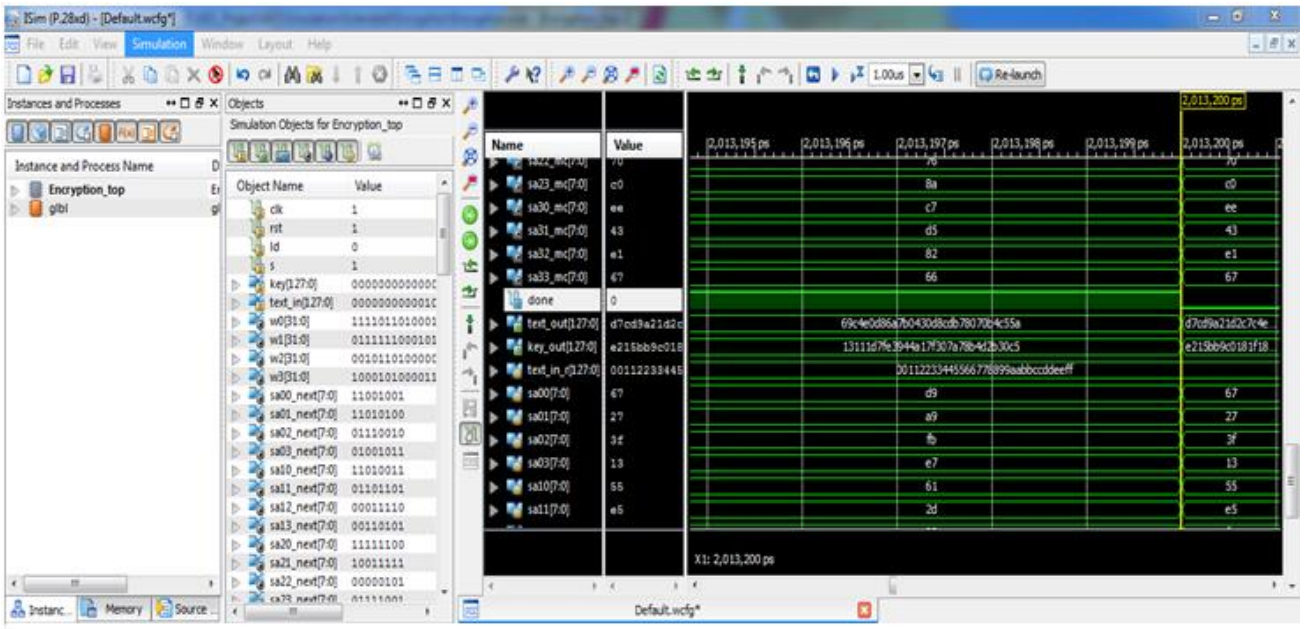
6. SIMULATION RESULT OF AES ENCRYPTION AND DECRYPTION USING PROPOSED KEY EXPANSION ALGORITHM IN SIMPLE, EXTENDED AND COMPOSITE MODE

In the test case, the AES encryption and decryption for a 16-byte data is simulated using the Xilinx 14.2 software tool. The encryption process of the AES algorithm generates a cipher text for a given data.

6.1. Encryption

In the encryption process the AES algorithm generates a cipher text for a given 16-byte data and cipher key. The simulation result of encryption in extended mode is shown in below fig.9.

Cipher key: 000102030405060708090a0b0c0d0e0f
Text_in: 00112233445566778899aabbccddeeff
Cipher text: 689c4e0d86a7b0430d8cdb78070b4c55a



Object Name	Value
clk	1
rst	1
ld	0
s	1
key[127:0]	000000000000
text_in[127:0]	00000000000010
w0[31:0]	1111011010001
w1[31:0]	0111111000101
w2[31:0]	0010110100000
w3[31:0]	1000101000011
sa00_next[7:0]	11001001
sa01_next[7:0]	11010100
sa02_next[7:0]	01110010
sa03_next[7:0]	01001011
sa10_next[7:0]	11010011
sa11_next[7:0]	01101101
sa12_next[7:0]	00011110
sa13_next[7:0]	00110101
sa20_next[7:0]	11111100
sa21_next[7:0]	10011111
sa22_next[7:0]	00000101
sa23_next[7:0]	01111001

Name	Value	2,013,197 ps	2,013,200 ps
sa22_mc[7:0]	70	76	70
sa23_mc[7:0]	c0	8a	c0
sa30_mc[7:0]	ee	c7	ee
sa31_mc[7:0]	43	d5	43
sa32_mc[7:0]	e1	82	e1
sa33_mc[7:0]	67	66	67
done	0		
text_out[127:0]	d7cd9a21d2c	69c4e0d86a7b0430d8cdb78070b4c55a	d7cd9a21d2c7c4e...
key_out[127:0]	e215bb9c018	13111d7fe3944a17f307a78b4d2b30c5	e215bb9c0181f18...
text_in_r[127:0]	00112233445	00112233445566778899aabbccddeeff	
sa00[7:0]	67	d9	67
sa01[7:0]	27	a9	27
sa02[7:0]	3f	fb	3f
sa03[7:0]	13	e7	13
sa10[7:0]	55	61	55
sa11[7:0]	e5	2d	e5

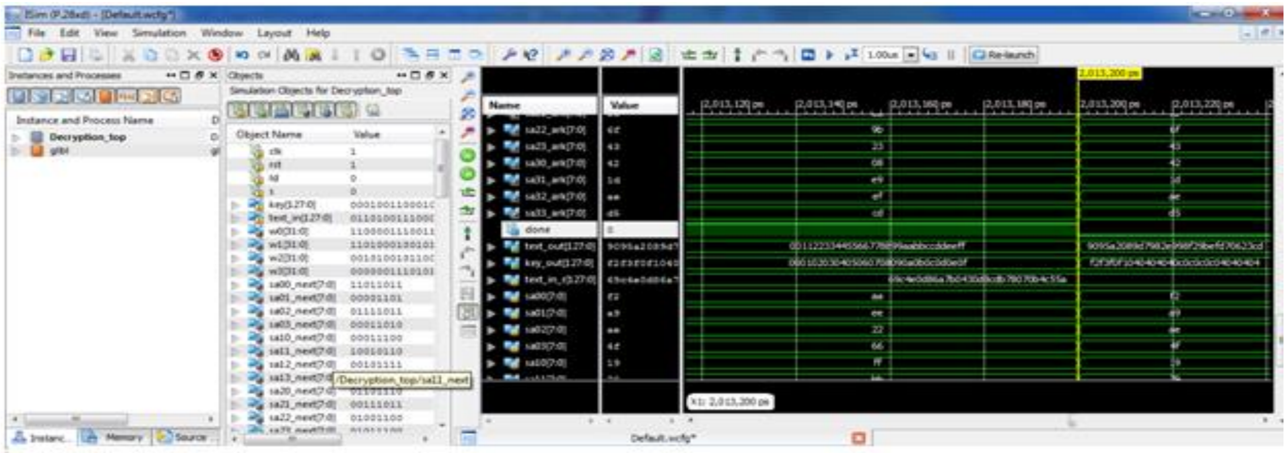
2,013,195 ps 2,013,196 ps 2,013,197 ps 2,013,198 ps 2,013,199 ps 2,013,200 ps
X1: 2,013,200 ps

Fig.9. simulation result of encryption in extended mode

6.2. Decryption

I the decryption process the AES algorithm generates a plain text for given 16-byte data and cipher key. the simulation results of decryption in simple, extended and composite mode is shown in Fig.10.

Text_in: 689c4e0d86a7b0430d8cdb78070b4c55a
Cipher key: 000102030405060708090a0b0c0d0e0f
Cipher text: 00112233445566778899aabbccddeeff



Object Name	Value
clk	1
rst	1
ld	0
s	0
key[127:0]	0001001100010
text_in[127:0]	0110100111000
w0[31:0]	1100001110011
w1[31:0]	1101000100101
w2[31:0]	0010100101100
w3[31:0]	0000001110101
sa00_next[7:0]	11011011
sa01_next[7:0]	00001101
sa02_next[7:0]	01111011
sa03_next[7:0]	00011010
sa10_next[7:0]	00011100
sa11_next[7:0]	10010110
sa12_next[7:0]	00101111
sa13_next[7:0]	[illegible]
sa20_next[7:0]	01101110
sa21_next[7:0]	00111011
sa22_next[7:0]	01001100
sa23_next[7:0]	[illegible]

Name	Value	2,013,160 ps	2,013,220 ps
sa22_ark[7:0]	6f	9b	6f
sa23_ark[7:0]	43	25	43
sa30_ark[7:0]	42	08	42
sa31_ark[7:0]	16	e9	16
sa32_ark[7:0]	ee	ef	ee
sa33_ark[7:0]	d5	cd	d5
done	0		
text_out[127:0]	9095a2089d7	00112233445566778899aabbccddeeff	9095a2089d7982e998f29befd70623cd
key_out[127:0]	f2f3f0f1040	000102030405060708090a0b0c0d0e0f	f2f3f0f10404040404cccccccc04040404
text_in_r[127:0]	69c4e0d86a7	69c4e0d86a7b0430d8cdb78070b4c55a	
sa00[7:0]	f2	ee	f2
sa01[7:0]	a9	ee	a9
sa02[7:0]	ee	22	ee
sa03[7:0]	4f	66	4f
sa10[7:0]	19	ff	19

2,013,120 ps 2,013,140 ps 2,013,160 ps 2,013,180 ps 2,013,200 ps 2,013,220 ps
X1: 2,013,200 ps

Fig.10 simulation result of decryption in extended mode

7. CONCLUSION

The goal of this proposed algorithm is to increase the level of security for Advanced encryption standard and provide a faster processing time. the difficulty of attacking the key is increased which powers the ability to resist major attacks due to the high randomization of key expansion in the new algorithm.AES in simple mode uses a more number of LUT's which requires more space this can be overcome with extended and composite mode in which the number of LUT's are reduced. The algorithm was synthesized using XILINX 14.3 and implemented on FPGA. The simulation result for AES-128 is obtained by simulating the proposed key expansion algorithm using Verilog Hardware Description Language.

ACKNOWLEDGEMENT

I am highly obliged to Department of Electronics and Communication Engineering, SDM College of Engineering and Technology. And I am highly grateful to my guide Mrs. Channakka Lakkannavar for her valuable instructions, guidance, corrections and presentation in my project work.

REFERENCES

[1] AI-Wen Luo, Qing-Ming Yi, Min Shi. "Design and Implementation of Area-optimized AES on FPGA" IEEE Inter.conf.chal sci com engin.,978-1-61284- 109-0/2011.

[2] H. Mestiri, N. Benhadjyoussef, M. Machhout and R. Tourki, "A Comparative Study of Power Consumption Models for CPA Attack,"

[3] International Journal of Computer Network and Information Security, Vol. 5, No. 3, pp. 25-31, 2013.

[4] A. Moh'd, Y. Jararweh and L. Tawalbeh, "AES-512: 512 bit Advanced Encryption Standard algorithm design and evaluation," 7th International Conference on Information Assurance and Security (IAS 2011), pp. 292- 297, 2011.

[5] M. Mozaffari-Kermani, and A. Reyhani-Masoleh, "Concurrent structure in dependent fault detection schemes for the advanced encryption standard," IEEE Transactions on Computers, Vol. 59, pp. 608-622, 2010.

[6] J.Yang, J.Ding, N.Li and Y.X.Guo,"FPGA-based design and implementation of reduced AES algorithm" IEEE Inter.Conf. Chal Envir Sci Com Engine(CESCE).,Vol.02, Issue.5-6, Jun 2010, pp.67-70.

[7] A.M. Deshpande, M.S. Deshpande and D.N. Kayatanavar, "FPGA Implementation of AES Encryption and Decryption"IEEE Inter.Conf.Cont,Auto,Com,and Ener., vol.01,issue 04, Jun.2009, pp.1-6.

BIOGRAPHIES

[1]. Ms. Suman S.B.
Completed B.E. in E&CE from SKSVMACET, Lakshmeshwar in the year 2014. Currently Pursuing 4th Sem M.Tech in Digital Electronics in SDMCET, Dharwad. My area of interest are Digital design, network security, Image Processing and Embedded Systems.

[2]. .Mrs. Channakka Lakkannavar
She is currently working as Assistant Professor in the department of E&CE, SDMCET, Dharwad. Her area of interest are digital design, network security and embedded systems.