ISSN : 2231-0711

Advanced Encryption Standard

Nikita Kangude, Priyesh Wani, Sanil Raut

Electronics and Telecommunications, Information Technology, Computer Science
University Of Pune

***Abstract*— Maintaining privacy in our personal communications is something everyone desires. Encryption is a means to achieve that privacy. It was invented for this very purpose. As more and more information is stored on computers or communicated via computers, the need to ensure that this information is invulnerable to snooping and/or tampering becomes more relevant. There are many standards available to encrypt data, including Data Encryption Standard (DES), Triple Data Encryption Standard (3 DES) and Advanced Encryption Standard (AES) out of which AES is recognized as industrial standard. AES provides strong security, simple design and better performance. It is used by many companies for creating commercially available encryption products. This paper is an overview of AES algorithm and its necessity in today's world.**

***Keywords*— AES, Encryption, DES, FIPS, Rijndael.**

I. INTRODUCTION

Cryptography is the practice and study of hiding information. Information is written in form of a code so as to keep it private, ensure security and safe transmission. There are various methods to achieve this. With the increase in the use of internet, messages are largely being transmitted in the form of electronic data. To protect the sensitive information in this form different algorithms and standards are created. Encryption is conversion of data to an unintelligible form called cipher text. Decryption is converting it back to plain text.

Federal Information Processing Standard 197 (FIPS 197) or AES is issued by the National Institute of Standards and Technology (NIST) after approval by the Secretary of Commerce in 2001. It falls under Computer Security Standard, Cryptography. This standard, specifying Rijndael algorithm, is used to protect electronic data.

Advanced Encryption Standard, AES or FIPS 197, is a symmetric-key encryption standard adopted by U.S. government. AES is symmetric since the same key is used for encryption and decryption. The key is used to convert a block of 128 bit data to an encrypted block of the same size. AES can be configured to use different size key-lengths depending on the data. The standard is available for 3 lengths of 128 bits, 192 bits and 256 bits. The resulting algorithms are called as AES-128, AES-192 and AES-256 respectively. As the length of the cipher increases the level of security increases with the time necessary for an attacker to stage a brute force attack, that is an exhaustive search of all possible key combinations to find the right one.

II. HISTORY

During 1970s, block ciphers such as RAS, DES and TDES were used. FIPS-46 or Data Encryption Standard (DES) provided mathematical algorithm for encrypting and decrypting binary coded information. The operations were carried out using a binary number called key. DES key consists of 64 bits out of which 56 bits are randomly generated and 8 are used for error detection. Triple Data Encryption Standard (TDES or 3DES) applies DES cipher algorithm three times to each data block.

Hardware now used is faster, having more memory with increasing use of internet and computer networks. DES now proves inadequate over several occasions, specially while ransmitting sensitive information over public networks. Hardware can be built to crack the DES key within a few hours. While DES is considerably fast in hardware applications, it is very slow in software, and hence is nadequate for the new age. While TDES id three times more secure than DES, it shows one third the speed of DES and still not suitable for software applications. The key of TDES hough more complex than that of DES can still be found out by complex computing software within a small period of time. Thus DES and TDES had to be replaced.

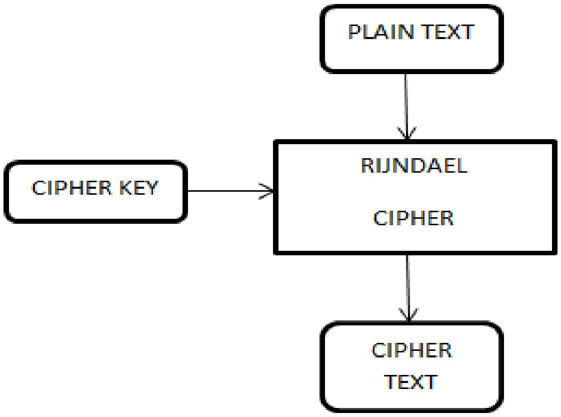
In 1997 the US National Institute of Standards and Technology (NIST) decided to draw on the world's finest cryptographic minds and asked them to submit new algorithms. The entire process took five years. Fifteen algorithms competed for three years and algorithms such as Rijndael, Twofish, Serpent were thoroughly scrutinized. The Rijndael cipher, developed by Joan Daemen and Vincent Rijmen was declared winner. AES was taken from these algorithms.

AES is founded on solid and well-published mathematical ground, and appears to resist all known attacks well. It has been published for a long time and has been the subject of intense scrutiny by researchers all over the world, hus indicating that in fact no back-door or known weakness exists. Enormous amounts of economic value and information s successfully protected by AES.

It is endorsed by National Security Agency (NSA) and is the first open cipher which is easy to approach publicly.

III. WORKING

Whenever the data is to be encrypted with the help of AES, the data is passed through the Rijndael cipher, in which the cipher key acts on the data to create Cipher text. Cipher text is encrypted form of the data.



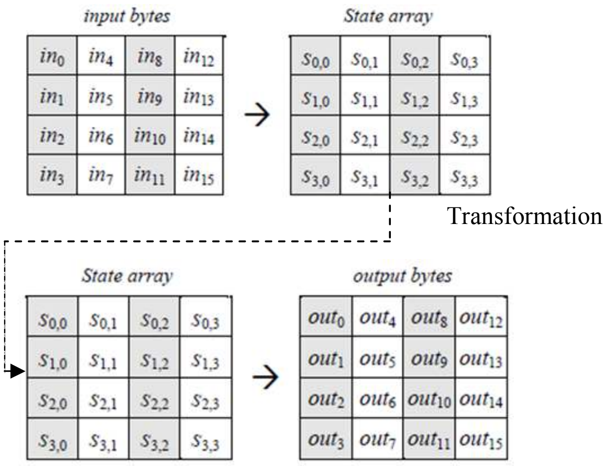
The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits .The block length is limited to 128 bits whereas the key length can be of any size mentioned. In this paper the key length is assumed to be 128 bits, which is most commonly implemented. Some precise steps are followed in the Rijndael Cipher to convert data into encrypted form.

Formation of data block-

Digital data is in the binary form, that is a bit, 0 or 1. 8 bits together form a byte, which is the basic unit for processing data in AES. The input, output and key are represented as arrays of bytes that are formed by dividing these sequences into groups of eight contiguous bits to form arrays of bytes.

If the 128 bit input sequence in represented as {*input0, input1, input2, input3,.....input127}*. Then a byte is formed as
In0= {*input*0, *input*1, …, *input*7};
In1= {*input*8, *input*9, …, *input*15};
In15= {*input*120, *input*121, …, *input*127}.
In general *Inn* = {*input*8*n*, *input*8*n*+1, …, *input*8*n*+7}.

The input block is copied into the state array which is modified at each stage of encryption. After the final stage, state array is copied to the output matrix. A state is a 2 dimensional array of 4 rows and N_b columns [length of block divided by 32(8 bytes X 4 rows) = N_b]. Ordering of bytes within the matrix is by column so for example the first 4 bytes of 128 bit plain text input to the encryption cipher occupy the first column of input matrix. Thus the matrix is as shown below.


input bytes

in_0	in_4	in_8	in_{12}
in_1	in_5	in_9	in_{13}
in_2	in_6	in_{10}	in_{14}
in_3	in_7	in_{11}	in_{15}

State array

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

Transformation

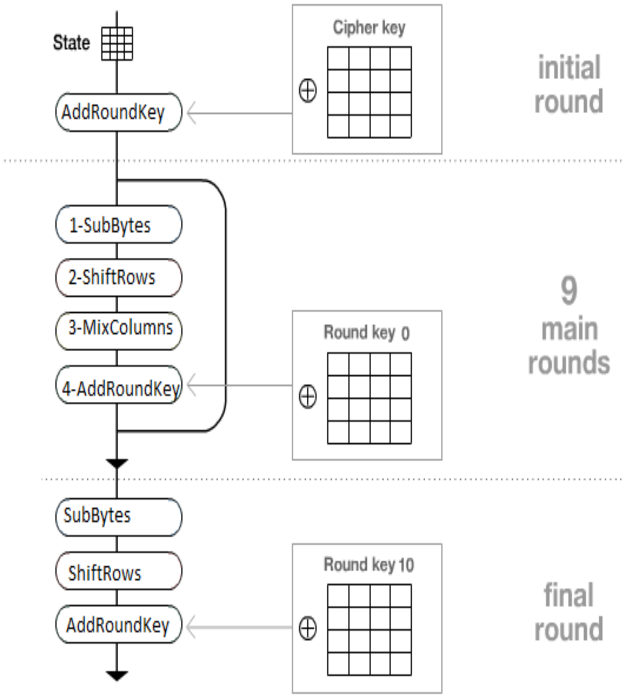
State array

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

output bytes

out_0	out_4	out_8	out_{12}
out_1	out_5	out_9	out_{13}
out_2	out_6	out_{10}	out_{14}
out_3	out_7	out_{11}	out_{15}

Four different transformations are performed on the data block in the process.



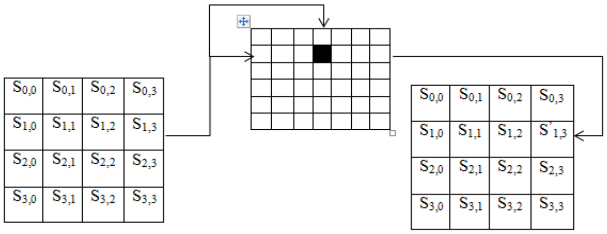
The four transformations mentioned in the above flowchart are:

1. SubBytes (Substitute bytes): Uses an S-box to perform a byte to byte substitution of block.
2. ShiftRows (Shift the rows): A simple permutation.
3. MixColumns
4. AddRoundKey (Add Round Key): A simple bitwise XOR of current block with a portion of expanded key.

There are total eleven rounds in which these transformations are used to alter the data block. The cipher begins with AddRoundKey stage, followed by nine rounds that include all four stages followed by a tenth round of three stages. AddRoundKey is the only stage that makes use of the key, and that is the reason why the cipher ends and begins with this stage. The other three stages provide confusion, diffusion and non-linearity, but by themselves would provide no security because they do not use the key. They cannot be used at the beginning or end as the cipher becomes without the knowledge of the key and so would add no security.

SubBytes:

In this stage the bytes in the data block are substituted with the values in a matrix called S-Box, which is defined by AES as 16 X 16 matrix of byte values. The S-Box contains all possible 256 eight values.



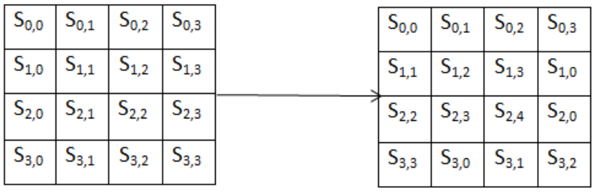
$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S'_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

The upper nibble of byte value of the state is used as a row value of the S-Box and the lower nibble of byte value of the state is used as column value of the S-Box. These rows and column values serve as indices into S-Box to select a unique 8 bit output value.
The S-Box is designed to be resistant to known cryptanalytic attacks.

ShiftRows:

In this stage the rows of the state array are altered by circularly left shifting them. The first row of the State array is not altered. 1-byte circular left shift is performed on the second row. 2-byte circular left shift is performed on the third row and 3-byte circular left shift is performed on the fourth row.



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

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$
$S_{2,2}$	$S_{2,3}$	$S_{2,4}$	$S_{2,0}$
$S_{3,3}$	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$

Shift Row Transformation

In general, ShiftRows transformation is given by:

$$\begin{pmatrix} C_{0,\,j} \\ C_{1,\,j} \\ C_{2,\,j} \\ C_{3,\,j} \end{pmatrix} = \begin{pmatrix} B_{0,\,j} \\ B_{1,\,j\text{-}1} \\ B_{2,\,j\text{-}2} \\ B_{3,\,j\text{-}3} \end{pmatrix}$$

For 0<=j<=3

MixColumns:

Operates on each column individually. Each byte of column is mapped into a new value that is function of all four bytes in that column. The transformation can be defined by following matrix multiplication on state .

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{pmatrix} = \begin{pmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{03} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{13} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{23} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{33} \end{pmatrix}$$

The coefficients of matrix in the above equation are based on linear code with maximal distance between code words, which ensures a good mixing among the bytes of each column.
Each element in product matrix is the sum of products of elements of one row and one column. The MixColumn transformation on a single column j(0<=j<=3) of state can be expressed as

$$S'_{0,j} = (2 * S_{0,j}) \oplus (3*S_{1,j}) \oplus S_{2,j} \oplus S_{3,j}$$

$$S'_{1,j} = S_{0,j} \oplus (2*S_{1,j}) \oplus (3*S_{2,j}) \oplus S_{3,j}$$

$$S'_{2,j} = S_{0,j} \oplus S_{1,j} \oplus (2*S_{2,j}) \oplus (3* S_{3,j})$$

$$S'_{3,j} = (3 * S_{0,j}) \oplus S_{1,j} \oplus S_{2,j} \oplus (2* S_{3,j})$$

The MixColumn transformation combined with ShiftRow transformation ensures that after a few rounds all output bits depend on the input bits.

AddRoundKey:

The 128 bits of State are bitwise XORed with 128 bits of round key. This transformation is very simple. It affects every bit of the state. It is the only transformation that uses the key. AddRoundKey transformation along with the other transformations ensures security.

$$\begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} \oplus \begin{bmatrix} K_{0,0} & K_{0,1} & K_{0,2} & K_{0,3} \\ K_{1,0} & K_{1,1} & K_{1,2} & K_{1,3} \\ K_{2,0} & K_{2,1} & K_{2,2} & K_{2,3} \\ K_{3,0} & K_{3,1} & K_{3,2} & K_{3,3} \end{bmatrix}$$

The general method for AddRoundKey transformation is given by:

$$\begin{pmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{pmatrix} = \begin{pmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{pmatrix} \oplus \begin{pmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{pmatrix}$$

For 0<=j<=3

KEY GENERATION:

The whole idea of encryption is that no one except the needed parties can ever deduce the key since the key is the only secret. The code written while developing any software implementing AES doesn't generate a random key. The generated key is a function of the password, and will be exactly the same every time a given password is used. Generally password is accepted from user and this password acts as a seed key for an algorithm that deterministically produces a string of bytes which can be used as a cryptographic key. Random number generator can be used to make encryption more efficient. Using this key, sub keys are generated. Sub key generation algorithm is used which is discussed below in brief with the help of an example.

Suppose that K[r] represents array of sub keys. Key scheduling is to generate K[r].

Key Schedule:
The key schedule is used to provide the sub-keys K[r].
Beginning with the key K, we extend it (called Key Expansion), and get the linear array-W.
W is an array with 4-byte words.

Key generation follows following rule to generate the sub key [N_b *(Nr+1)],
Where N_b- number of columns of the matrix B[i,j]
N_r - is the number of rounds.

For N_b =4, N_r =10, so we have [4*(10+1)] = 44 words of 32b.
Every block of 4 words have 128b and it is the sub-key [r].
The first sub-key is K[0] = (W[0],W[1],W[2],W[3]), which is the copy of the key K.
The next sub-keys are derived from the K[0].

Rule to Derive the Sub-Keys:

W[4] = W[0] xor temp$_1$
W[5] = W[1] xor W[4]
W[6] = W[2] xor W[5]
W[7] = W[3] xor W[6]
W[8] = W[4] xor temp$_2$
...
W[42] =W[38] xor W[41]
W[43] =W[39] xor W[42]

In general,

W [i] = W[i-N_k] xor W[i-1],

Where, i mod N_k < > 0
Where, i mod N_k = 0

We have,
W[i] = W[i- N_k] xor temp$_k$,

Where,
temp$_k$ = Sub-Byte [Shift(1) (W[i-1])] xor rcon[k].

Sub-Byte is applied to W[i-1] with a byte shifted, and rcon[k], is defined as,
rcon[k] = (RC[k], 00, 00, 00), with RC[1] =1,
and RC[k]=X*RC[k-1] =X^(k-1),
are elements in GF (2^8).

For Example,
Input-
0011 0010 0100 0011 1111 0110 1010 1000
1000 1000 0101 1010 0011 0000 1000 1101
0011 0001 0011 0001 1001 1000 1010 0010
1110 0000 0011 0111 0000 0111 0011 0100
Input bits converted to HEX are [32 43 f6 a8 88 5a 30 8d 31 98 a2 e0 37 07 34]

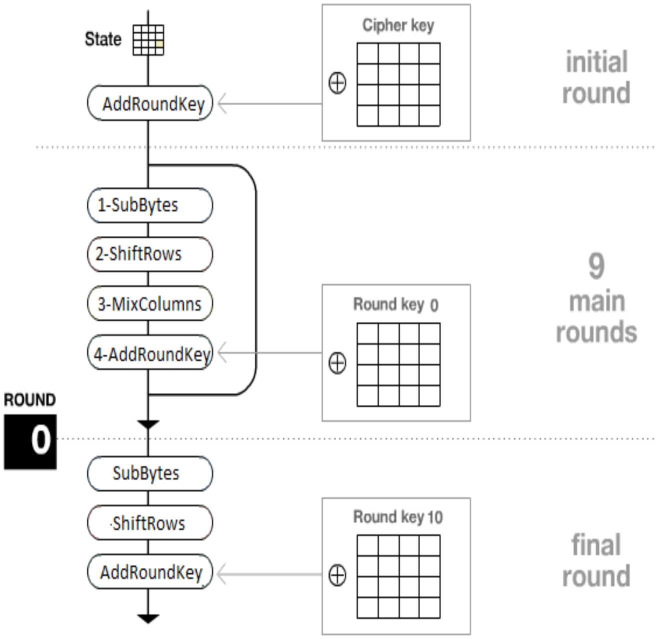
32	88	31	e0
43	5a	31	37
f6	30	98	07
a8	8d	a2	34

Cipher Key

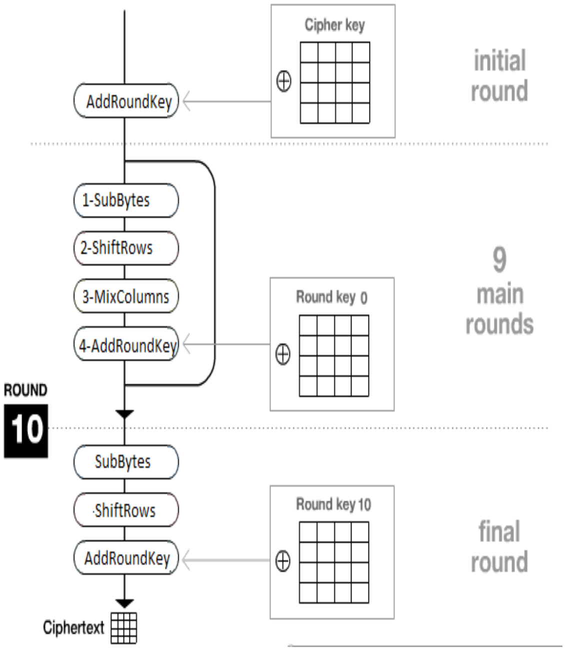
2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

Encryption process:

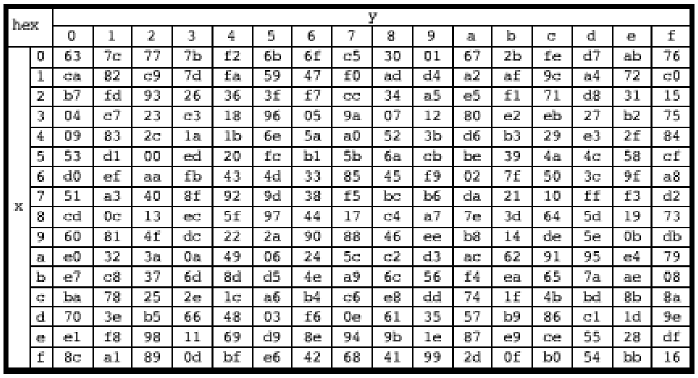
At the start



At the end



S-Box for the Sub-bytes



hex		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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
	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

Replacing the elements according to the rows and columns of the S-Box we get,

d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

Shift Rows process:

Row 1 is shifted 0 times to the left
Row 2 is shifted 1time to the left
Row 3 is shifted 2 times to the left
Row 4 is shifted 3 times to the left

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

Mix Columns Process:

Multiply each column with the given matrix and replace this column with the answer obtained.

$$\begin{bmatrix}02&03&01&01\\01&02&03&01\\01&01&02&03\\03&01&01&02\end{bmatrix} \bullet \begin{bmatrix}d4\\bf\\5d\\30\end{bmatrix} = \begin{bmatrix}04\\66\\81\\e5\end{bmatrix}$$

Thus,

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

Add Round key process:

Round key

a0	88	23	2a
fa	54	a3	6c
fe	2c	39	76
17	b1	39	05

(round key column) XOR (State column)= (answer column)

$$\begin{bmatrix}04\\66\\81\\e5\end{bmatrix} \oplus \begin{bmatrix}a0\\fa\\fe\\17\end{bmatrix} = \begin{bmatrix}a4\\9c\\7f\\f2\end{bmatrix}$$

ROUND 1 Answer

a4	68	6b	02
9c	9f	5b	6a
7f	35	ea	50
f2	2b	43	49

The Cipher text is obtained after 10 such rounds

Cipher text

39	02	dc	19
25	dc	11	6a
84	09	85	0b
1d	fb	97	32

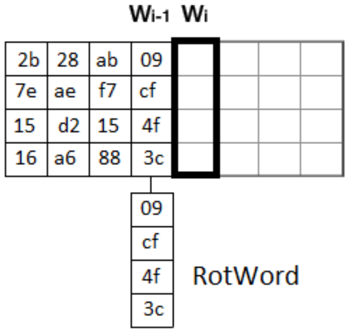
PROCESS TO OBTAIN ROUND KEY:

The expanded key can be seen as an array of 32 bit words (columns), numbered from 0 to 43.
First four columns are filled with the given Cipher Key, which is as follows,

2b	28	ab	09
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

Words in positions that are multiples of 4 (W_4, W_8,...W_{40}) are calculated by,

a) Applying the RotWord and SubBytes transformation to the previous word W_{i-1}.



2b	28	ab	09				
7e	ae	f7	cf				
15	d2	15	4f				
16	a6	88	3c				

Shift the column upwards by one block

$$\begin{array}{|c|}\hline 09 \\ \hline cf \\ \hline 4f \\ \hline 3c \\ \hline \end{array} \quad \text{to} \quad \begin{array}{|c|}\hline cf \\ \hline 4f \\ \hline 3c \\ \hline 09 \\ \hline \end{array}$$

Replace the blocks from the values of S-Box as in SubBytes Transformation

Thus the transformed column is,

$$\begin{array}{|c|}\hline 8a \\ \hline 84 \\ \hline eb \\ \hline 01 \\ \hline \end{array}$$

b) Adding (XOR) this result to the word 4 positions earlier W_{i-4} , plus a round constant Rcon.

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

RCON

Since this function is for Round Key 1, first column of RCON is used.

$$\begin{array}{|c|}\hline 2b \\ \hline 7e \\ \hline 15 \\ \hline 16 \\ \hline \end{array} \oplus \begin{array}{|c|}\hline 8a \\ \hline 84 \\ \hline eb \\ \hline 01 \\ \hline \end{array} \oplus \underbrace{\begin{array}{|c|}\hline 01 \\ \hline 00 \\ \hline 00 \\ \hline 00 \\ \hline \end{array}}_{\textbf{Rcon(4)}} = \begin{array}{|c|}\hline a0 \\ \hline fa \\ \hline fe \\ \hline 17 \\ \hline \end{array}$$

Thus, the answer after first round is,

2b	28	ab	09	a0
7e	ae	f7	cf	fa
15	d2	15	4f	fe
16	a6	88	3c	17

The remaining 32 bit words W_i, are calculated by adding (XOR) the previous word W_{i-1} , with the W_{i-4}.
Thus for 2nd column of Round Key 1

	W_{i-4}			W_{i-1}	W_i		
2b	28	ab	09	a0			
7e	ae	f7	cf	fa			
15	d2	15	4f	fe			
16	a6	88	3c	17			

$$\begin{array}{|c|}\hline 28 \\ \hline ae \\ \hline d2 \\ \hline a6 \\ \hline \end{array} \oplus \begin{array}{|c|}\hline a0 \\ \hline fa \\ \hline fe \\ \hline 17 \\ \hline \end{array}$$

$$\begin{array}{|c|}\hline 28 \\ \hline ae \\ \hline d2 \\ \hline a6 \\ \hline \end{array} \oplus \begin{array}{|c|}\hline a0 \\ \hline fa \\ \hline fe \\ \hline 17 \\ \hline \end{array} = \begin{array}{|c|}\hline 88 \\ \hline 54 \\ \hline 2c \\ \hline b1 \\ \hline \end{array}$$

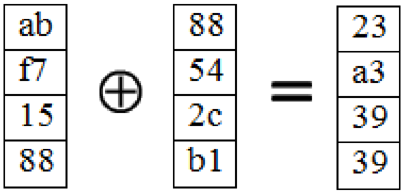
Thus,

2b	28	ab	09	a0	88
7e	ae	f7	cf	fa	54
15	d2	15	4f	fe	2c
16	a6	88	3c	17	b1

For the 3rd column,

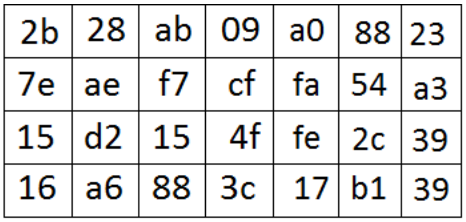
		W_{i-4}			W_{i-1}	W_i	
2b	28	ab	09	a0	88		
7e	ae	f7	cf	fa	54		
15	d2	15	4f	fe	2c		
16	a6	88	3c	17	b1		

$$\begin{array}{|c|}\hline ab \\ \hline f7 \\ \hline 15 \\ \hline 88 \\ \hline \end{array} \oplus \begin{array}{|c|}\hline 88 \\ \hline 54 \\ \hline 2c \\ \hline b1 \\ \hline \end{array}$$



$$\begin{array}{|c|}\hline ab\\\hline f7\\\hline 15\\\hline 88\\\hline\end{array} \oplus \begin{array}{|c|}\hline 88\\\hline 54\\\hline 2c\\\hline b1\\\hline\end{array} = \begin{array}{|c|}\hline 23\\\hline a3\\\hline 39\\\hline 39\\\hline\end{array}$$

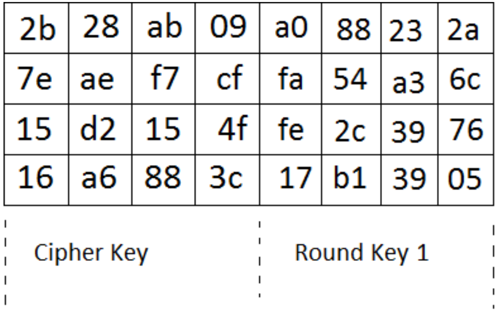
Thus,



2b	28	ab	09	a0	88	23
7e	ae	f7	cf	fa	54	a3
15	d2	15	4f	fe	2c	39
16	a6	88	3c	17	b1	39

The fourth column for the Round Key is formed in the same manner.

The Round key used for the next AddRoundKey transformation is,

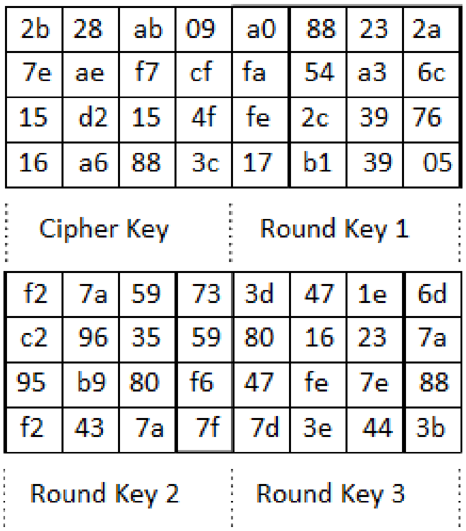


2b	28	ab	09	a0	88	23	2a
7e	ae	f7	cf	fa	54	a3	6c
15	d2	15	4f	fe	2c	39	76
16	a6	88	3c	17	b1	39	05

Cipher Key

Round Key 1

After 10 such functions, we get Round Keys for each transformation in this manner,



2b	28	ab	09	a0	88	23	2a
7e	ae	f7	cf	fa	54	a3	6c
15	d2	15	4f	fe	2c	39	76
16	a6	88	3c	17	b1	39	05

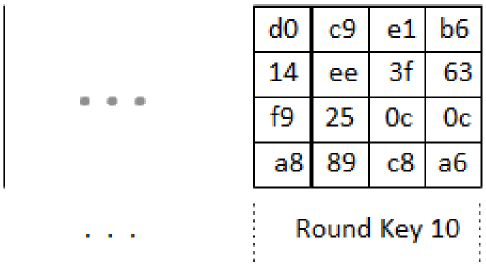
Cipher Key

Round Key 1

f2	7a	59	73	3d	47	1e	6d
c2	96	35	59	80	16	23	7a
95	b9	80	f6	47	fe	7e	88
f2	43	7a	7f	7d	3e	44	3b

Round Key 2

Round Key 3



. . .

d0	c9	e1	b6
14	ee	3f	63
f9	25	0c	0c
a8	89	c8	a6

. . .

Round Key 10

Thus the Encrypted text is

0011 1001 0010 0101 1000 0100 0001 1101
0000 0010 1101 1100 0000 1001 1111 1011
1101 1100 0001 0001 1000 0101 1001 0111
0001 1001 0110 1010 0000 1011 0011 0010

IV. Conclusions

AES is seen in all areas of life. It is used in C++, Java, .NET, PHP, also for other encrypting algorithms. There are a number of softwares available for encrypting and decrypting in AES. These make it easy for any individual to send sensitive information. One such software is AES Crypt. AES Crypt is a file encryption software product available on several operating systems that uses the industry standard AES to easily and securely encrypt files. AES Crypt is also the perfect solution for those who wish to backup information and store that data at a bank, in a cloud-based storage service, and any place where sensitive files might be accessible by someone else.

There is currently no evidence that AES has any weaknesses making any attack other than exhaustive search, i.e. brute force, possible. Even AES-128 offers a sufficiently large number of possible keys, making an exhaustive search impractical for many decades, provided no technological breakthrough causes the computational power available to increase dramatically and that theoretical research does not find a short cut to bypass the need for exhaustive search. There are many pitfalls to avoid when encryption is implemented, and keys are generated. It is necessary to ensure each and every implementations security, but hard since it requires careful examination by experts. An important aspect of an evaluation of any specific implementation is to determine that such an examination has been made, or can be conducted. Correctly implemented AES-128 is likely to protect against a million dollar budget for at least 50 – 60 years and against individual budgets for at least another 10 years.

ACKNOWLEDGMENT

Success is the manifestation of diligence, perseverance, inspiration, motivation and innovation. Every nice work begins with a systematic approach reaching successful completion. I am deeply grateful to Prof. Ashwinee Jagtap for guiding us and immensely honoured by her connection.

REFERENCES

[1] *Cryptography and Network Security* by William Stallings, 4th edition, Pearson Education.

[2] *Journal of Research of the National Institute of Standard and Technology* by Westlund, Harold B. (2002).

[3] J. Daemen and V. Rijmen, *AES Proposal: Rijndael, AES Algorithm Submission*, September 3, 1999.

[4] *Improved Cryptanalysis of Rijndael* by John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner and Doug Whiting, Fast Software Encryption, 2000.

[5] *AES Announced* by Bruce Schneier, 2000.

[6] National Institute of Standards and Technology (NIST). *Advanced Encryption Standard (AES).* Federal Information Processing Standards (FIPS) Publication 197, Nov. 2001.